Web Accessibility in E-learning

– Identifying and Solving Accessibility Issues for WCAG 2.0 Conformance in an E-learning Application

Sebastian Lundqvist
Johan Ström

Tutor: Anders Fröberg
Examiner: Erik Berglund
Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Sebastian Lundqvist, Johan Ström
Abstract

With the rise of e-learning and recent legislations of the European Union enforcing conformance to Web Content Accessibility Guidelines 2.0 for web applications of public sector bodies, the issue of identifying and solving web accessibility issues in e-learning applications is more relevant than ever. This thesis is based on a case study of a publisher of course literature whose intention is to improve the accessibility of their e-learning applications. The thesis contains a theoretical foundation on disabilities, e-learning and web accessibility including assistive technologies, the web accessibility guidelines WCAG 2.0 and the web accessibility evaluation method WCAG-EM. This theory is used for developing solutions to accessibility issues found in the e-learning applications and comparing accessibility with e-learning theory. The results are presented as concrete examples to be useful for developers of e-learning applications. It was found that there exist a few instances where accessibility and e-learning contradict each other, but the solutions to accessibility problems are most often not affected by the learning premise of the applications.

Keywords: Web Accessibility, Web Accessibility Evaluation, E-learning, Web Content Accessibility Guidelines, WCAG 2.0, WCAG-EM
Acknowledgement

This thesis would not have been possible without the assistance of a few helpful people to whom we would like to extend our sincere gratitude.

First and foremost, we would like to thank the publishing company for giving us the chance to work with them, and for the resources they have provided us. In addition, we would like to thank our supervisors at the company who have always been very helpful and friendly.

Furthermore, we would like to thank our examiner Erik Berglund for providing us with valuable feedback. We are also grateful for the feedback received from our fellow student Gustav Olsson, who helped us see the thesis from a different perspective.

Last but not least, we would like to extend our gratitude to Hampus Sethfors at Axess Lab for providing us with valuable insights on how to best solve certain accessibility issues, as well as answering our many questions about web accessibility in general.

Linköping in June 2018
Sebastian Lundqvist and Johan Ström
Table of Contents

1 Introduction ......................................................................................................................... 1

1.1 Motivation ....................................................................................................................... 1

1.2 Aim .................................................................................................................................. 2

1.3 Research Question .......................................................................................................... 2

2 Background .......................................................................................................................... 3

2.1 The Employer ................................................................................................................. 3

2.2 The E-learning Application ............................................................................................ 3

3 Theory ................................................................................................................................ 4

3.1 Introduction to Disabilities .............................................................................................. 4

3.1.1 Visual Impairments ...................................................................................................... 4

3.1.2 Cognitive and Learning Disabilities ........................................................................... 5

3.1.3 Motor Disabilities ........................................................................................................ 5

3.2 Web Content Accessibility Guidelines ............................................................................ 6

3.2.1 The Past ...................................................................................................................... 6

3.2.2 The Present ............................................................................................................... 6

3.2.3 The Future ................................................................................................................ 8

3.3 Web Accessibility and Assistive Technologies ................................................................. 8

3.3.1 Keyboard Navigation ................................................................................................. 8

3.3.2 Contrast ..................................................................................................................... 9

3.3.3 Text Presentation ....................................................................................................... 10

3.3.4 Screen Reader Compatibility and Page Layout ......................................................... 11

3.3.5 Text Content ............................................................................................................. 13

3.4 Introduction to E-learning ............................................................................................... 14

3.4.1 Cognitive Theories and How People Learn .............................................................. 14

3.4.2 Benefits of E-learning ............................................................................................... 15

3.4.3 Hazards of E-learning ............................................................................................... 15

3.4.4 Guidelines for E-learning Applications ................................................................... 16

3.4.5 Asynchronous and Synchronous E-learning ............................................................ 19

3.4.6 Blended Learning ..................................................................................................... 19
3.4.7  E-learning for People with Disabilities .............................................. 20

3.5  Web Accessibility Evaluation ........................................................................ 21
  3.5.1 Website Accessibility Conformance Evaluation Methodology .......... 21
  3.5.2 Automated Testing Tools ......................................................................... 24
  3.5.3 Sampling ................................................................................................. 26

4  Method .............................................................................................................. 28
  4.1  Pre-study ....................................................................................................... 28
  4.2  Implementation ............................................................................................. 29
    4.2.1  Prototyping ............................................................................................. 29
    4.2.2  Feature Development ............................................................................. 30
  4.3  Evaluation ..................................................................................................... 30

5  Result ................................................................................................................. 33
  5.1  Accessibility Evaluation of Baseline Applications ................................... 33
  5.2  Adherence to E-learning Principles ........................................................... 34
  5.3  Solving WCAG Level A and AA Accessibility Issues ............................... 35
    5.3.1  Non-text Content (1.1.1) ................................................................. 35
    5.3.2  Audio-only and Video-only (Prerecorded) (1.2.1) ......................... 36
    5.3.3  Captions (Prerecorded) (1.2.2) ......................................................... 37
    5.3.4  Info and Relationships (1.3.1) ........................................................... 37
    5.3.5  Use of Color (1.4.1) ........................................................................... 39
    5.3.6  Contrast (Minimum) (1.4.3) .............................................................. 40
    5.3.7  Resize Text (1.4.4) .............................................................................. 44
    5.3.8  Keyboard (2.1.1) ................................................................................. 45
    5.3.9  Bypass Blocks (2.4.1) ......................................................................... 46
    5.3.10 Focus Visible (2.4.7) .......................................................................... 46
    5.3.11 On Input (3.2.2) .................................................................................. 47
    5.3.12 Consistent Navigation (3.2.3) ............................................................ 48
    5.3.13 Labels or Instructions (3.3.2) .............................................................. 49
    5.3.14 Parsing (4.1.1) ................................................................................... 50
    5.3.15 Name, Role, Value (4.1.2) ................................................................. 50
  5.4  Auxiliary Accessibility Improvements ......................................................... 51
    5.4.1  Visual Presentation ................................................................................ 51
Table of Figures

Table 1: Passing WCAG success criteria in baseline applications ........................................... 33
Table 2: Passing WCAG success criteria in updated applications ........................................ 54

Figure 1: The WCAG 2.0 hierarchy ........................................................................................ 6
Figure 2: The five steps of WCAG-EM, adapted from WAI [53] ............................................ 22
Figure 3: Balsamiq wireframe tool ......................................................................................... 29
Figure 4: Audio-only and video-only (prerecorded) example .................................................. 37
Figure 5: Example of a word selection page ......................................................................... 38
Figure 6: Use of color example problem ............................................................................. 39
Figure 7: Use of color example prototype ............................................................................ 39
Figure 8: Use of color example implementation ................................................................. 40
Figure 9: Accessibility toolbar prototype ............................................................................ 41
Figure 10: Before and after altering a color with the contrast conformance tool .............. 41
Figure 11: Contrast selection ............................................................................................... 43
Figure 12: Text size example ............................................................................................... 45
Figure 13: Two options in a multiple-choice quiz, with option A selected ......................... 47
Figure 14: Text input quiz before and after ........................................................................ 48
Figure 15: Application hierarchy ......................................................................................... 49
Figure 16: Backwards navigation button from example assignment .................................. 49
Figure 17: Disabled button example ................................................................................... 51
Figure 18: Large line height (left) and large letter spacing (right) ...................................... 52
The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.

*Tim Berners-Lee, W3C Director and Inventor of the World Wide Web*
1 Introduction

In the physical world, accessibility is often integrated in society and regulated by law in developed countries. Public transportation, workplaces, hospitals and other infrastructure are made accessible to people with different types of disabilities. The internet, being largely unregulated, has not yet been made accessible to the same extent, despite the tremendous opportunities the technology presents to the disabled community [1].

In recent years, however, the accessibility of the web has become an increasingly important topic. The Web Accessibility Initiative\(^1\) (WAI) is a working group originating from the World Wide Web Consortium\(^2\) (W3C) with the aim of increasing the overall accessibility of the web. The group was formed in 1997 and the first Web Content Accessibility Guidelines\(^3\) (WCAG) were released in 1999. In 2008, WCAG 2.0 was released, raising the bar further as well as including cognitive and learning disabilities to a higher degree.

Another apparent trend in recent years is the advancement of e-learning [2]. E-learning not only has the advantage of being cheap and effective, but also has the potential to help disabled students overcome barriers that previously prevented them access to education. Unfortunately, accessibility problems are present in many e-learning applications today [1] [3]. However, as e-learning moves into the realm of public education, it becomes subject to regulations not normally associated with the internet. This includes accessibility legislations that promote equal access to education for everyone. The European standardization bodies ETSI, CEN and CENELEC have developed a common standard for accessibility requirements in government procurement of IT systems [4]. The web accessibility requirements in this standard are retrieved from WCAG 2.0. This standard has since been adopted as an EU Directive, forcing public sector companies to make their websites accessible.

1.1 Motivation

Adhering to the WCAG and in other ways ensuring that the content of the web is accessible to any user should be a primary concern of every organization. According to a report commissioned by Microsoft, as many as 57% of computer users have at least some type of mild difficulty or impairment that might impact computer use and are likely to benefit from the use of accessible technology [5]. For companies and organizations creating e-learning content for public education, it becomes especially important since they become subject to accessibility laws and regulations. In the light

\(^1\) https://www.w3.org/WAI/
\(^2\) https://www.w3.org/
\(^3\) https://www.w3.org/WAI/intro/wcag
of this, it is justified to investigate how compatible the WCAG is with the peculiarities of the e-learning domain.

Previously, Guenaga et al. [6] studied how various guidelines, including WCAG, align with the actual needs of disabled e-learning users. However, at the time of their research, WCAG 1.0 was the most recent version and several major changes have since been introduced. Other papers investigate the current situation for people with disabilities in the online learning environment [1] [7] [8]. In this thesis we intend to expand on this work.

1.2 Aim

This thesis aims to investigate and evaluate the accessibility of an e-learning application used in a blended learning environment for primary and secondary school students in terms of WCAG 2.0 conformance, while considering both accessibility and e-learning perspectives to produce effective solutions for the identified accessibility issues. The thesis further aims to provide developers with example solutions for these accessibility issues.

1.3 Research Question

The research question for this thesis is:

How can accessibility issues be identified and solved in an existing e-learning application to improve accessibility in terms of WCAG 2.0 conformance?

To answer this question, we need to define what e-learning is and find best practices and guidelines for e-learning to find any conflicts between these guidelines and the accessibility guidelines of WCAG. We must also relate the web applications to e-learning theories to find out if there are any e-learning properties or e-learning guideline violations that would affect design decisions in our implementation. Furthermore, we need to find accessibility evaluation methods capable of detecting as many of the present accessibility issues as possible.
2 Background

This chapter introduces the company that provided the assignment and briefly describes the e-learning application studied in the thesis.

2.1 The Employer

The assignment for this thesis was designed and implemented in collaboration with a publishing house that wishes to remain anonymous and will henceforth be referred to as the company. The company is based in Sweden and mainly works with textbooks for primary and secondary school students. In recent years, the company has invested time and resources trying to digitize part of their printed content. This effort has partly resulted in a set of web applications intended for teachers and students as a complement to the printed textbooks. Recent accessibility legislations have prompted the company to realize the need to revisit and improve these web applications.

2.2 The E-learning Application

The web applications are built using the model-view-controller framework BackboneJS4, in which the JavaScript library jQuery5 is used to dynamically update content. The web applications are purely front-end, with the content being fetched from XML files generated by the company’s production tools.

A set of templates are used to structure the web applications, each template representing a different area of the web applications, such as the navbar, footer or a certain type of quiz within the application. In this way, the functionality of all web applications is changed by modifying only a few templates. This library of web applications will henceforth be referred to as the web applications, or simply the applications, and the basis of our changes to these web applications will be referred to as the baseline applications.

4 http://backbonejs.org/
5 https://jquery.com/
3 Theory

This chapter forms the theoretical background for this thesis. It contains topics of disabilities, e-learning and web accessibility, including the Web Content Accessibility Guidelines, assistive technologies and web accessibility evaluation.

3.1 Introduction to Disabilities

Disabilities exist in many forms and degrees of severity. Whether they are present from birth, developed later in life or caused by an accident, they inhibit a person’s abilities to carry out daily tasks. Modern definitions of disability stray from the notion that something is wrong with the person, emphasizing that a disability is an incompatibility between the individual and whichever medium they interact with. This opens broader perspectives on disabilities. This thesis focuses on visual impairments, cognitive disabilities and motor disabilities.

In 2012, the US Census Bureau reported that 19% of Americans in 2010 suffered from some form of disability [9]. Similarly, the Public Health Agency of Sweden [10] and Funka [11], Swedish-founded international company of accessibility experts—responsible for the authorized translation of WCAG 2.0 to Swedish and involved in developing the accessibility requirements in public procurement in the European Union—reported that between 1.3 million and 1.8 million people in Sweden suffer from disability. While hearing impairments and issues with mobility are most common among the elderly, there is a significant number of young people diagnosed with mainly visual impairments and cognitive disabilities. The Swedish National Agency for Education [12] reports that 6% of the youngest children are diagnosed with some disability. This figure increases to 17% of children aged 16 [12].

3.1.1 Visual Impairments

There are several types of visual impairments that each affect people to different degrees. For some, the impairment may have almost no impact on everyday life, while others have such a severe disability that the visual sense is completely lost. For people with severe visual impairment, technologies like the internet can facilitate interaction with the outside world. For someone who is only mildly affected, the impairment might merely be a slight annoyance when browsing the web. In either case, the accessibility for these users can be significantly improved by specifically designing web applications according to their needs.

Reports show that as much as 3.8% of the Swedish population has visual impairments so severe that they have difficulty reading the newspaper even with glasses or corrective lenses [13]. The most common types of visual impairments in Sweden are short- and far-sightedness as well as astigmatism according to a report from the Swedish National Board of Health and Welfare [14]. Other types of visual impairments include color blindness and age-related macular degeneration (AMD).
3.1.2 Cognitive and Learning Disabilities

Cognitive disabilities are disabilities related to different cognitive actions, such as learning, perception, concentration and memory. In a study on university students with disabilities by Fichten et al. [1], the most common type of disability with a large margin was learning disabilities, with 41% of the respondents reporting having one. According to a statistics compilation carried out on behalf of the Swedish Post and Telecom Authority, the single most common diagnosis in these areas is dyslexia, with between 5-8% of the Swedish population suffering from the disability [13]. According to Dyslexia International, the prevalence is at least 10% for any given population [15]. The number of diagnoses between people speaking different languages may vary, however, since the manifestation is more severe in languages with deep orthographies, such as English or French, compared to languages such as Italian [16]. Dyslexia is just one diagnosis related to difficulties in reading or writing. It is estimated that 20% of the Swedish population suffers from reading/writing disability [13].

The diagnosis ADHD/ADD may be the second most common cognitive disability among children, affecting 3-6% of Swedish children aged 5 to 18 [13]. The corresponding figure for adults is 2-3% [13]. Around 80% of the latter have at least one other psychiatric diagnosis, often depression, anxiety, eating disorders or personality disorders.

3.1.3 Motor Disabilities

Motion and motor disabilities are disabilities related to either compound movements such as walking or reduced mobility in body parts such as the arms and hands respectively. While 560 000 people in Sweden are estimated to suffer from a motion disability, half of those are more than 80 years old [11]. A much larger number, 1 330 000 people, is reported to have reduced mobility in arms and hands, making interactions with computers difficult for them [13]. This reduced mobility is not limited to people with old age, as temporary disabilities resulting from accidents—in sports, for example—affect people of all ages. Permanent spinal cord injuries resulting in quadriplegia also make it difficult to use a mouse or keyboard, and individuals with quadriplegia rely on assistive technologies in order to access the web [17]. The most common cause of spinal cord injuries is motor vehicle accidents [17]. According to Friedman & Bryen, many individuals with cognitive disabilities have problems with limited fine motor control, hand/eye coordination and finger dexterity [18].
3.2 Web Content Accessibility Guidelines

This section presents the basic concepts of the current web accessibility guidelines as well as a brief history and predicted future of WCAG.

3.2.1 The Past

Ever since the early years of the internet, people have been aware of the issue of disability access. Many authors and organizations developed guidelines for web accessibility during the second half of the 1990’s and, in 1998, the University of Wisconsin-Madison published the 8th version of the *Unified Web Site Accessibility Guidelines*, which served as the starting point for the first web accessibility recommendations from the World Wide Web Consortium. In 1999, W3C published their first web accessibility guidelines, WCAG 1.0.

WCAG 1.0 consists of 14 guidelines, each containing a number of checkpoints describing how to apply the guideline. Every checkpoint has a priority level. Priority 1 checkpoints *must* be satisfied and conforming to these is described as level A conformance. Priority 2 checkpoints *should* be satisfied, resulting in level AA conformance, and priority 3 checkpoints *may* be satisfied, resulting in level AAA conformance. [19]

3.2.2 The Present

In 2001, the first concept proposal of WCAG 2.0 was published. In the years following the proposal, accessibility experts and members of the disability community provided feedback and several iterations of proposals were written. In 2008, WCAG 2.0 was published as a W3C Recommendation.

The WCAG 2.0 is divided into four accessibility principles, each principle having a number of guidelines to follow in order to ensure accessibility for as many as possible. The guidelines are further divided into testable success criteria, the passing or failing of which determines the conformance level of the application. The A, AA and AAA conformance levels are maintained from WCAG 1.0. W3C also provides a wide variety of techniques sufficient for meeting the success criteria. [20]

The WCAG 2.0 hierarchy is illustrated in Figure 1.

**Figure 1**: The WCAG 2.0 hierarchy
For a web page to conform to WCAG 2.0, there are five conformance requirements that need to be satisfied.

1. **Conformance Level**
   
   The conformance levels are, as previously stated, A for the minimum level of conformance, and AA and AAA for higher levels of conformance. For a certain level of conformance, *all* success criteria of that level must be satisfied.

2. **Full Pages**
   
   Conformance is assessed for entire web pages, meaning parts of a web page cannot be excluded for conformance.

3. **Complete Processes**
   
   When a sequence of web pages is presented for accomplishing an activity, all web pages have to conform to the specified level or better.

4. **Only Accessibility-Supported Ways of Using Technologies**
   
   Technologies relied upon to satisfy the success criteria are used in *accessibility-supported* ways, and if any information or functionality is provided in a way that is not accessibility supported, it is also provided in a way that is accessibility supported.

5. **Non-Interference**
   
   The ability of users to access a page is not blocked if technologies are used in ways that are not accessibility supported, or if they are used in a non-conforming way. The web page should also continue to meet the conformance requirements should any technology that is not *relied upon* be turned on, off or not be supported by a user agent.

If these requirements are satisfied, a conformance claim can be made, but is not required. [20]

In 2014, the European standardization organizations ETSI, CEN and CENELEC published a common European standard for accessibility, EN 301 549[6], on behalf of the European Commission. The section of the standard pertaining to web content is based on WCAG 2.0. Essentially, web pages and mobile applications of public sector bodies are to conform to the AA level of the WCAG 2.0 conformance requirements.

In October 2016, these requirements for web content were approved as EU Directive 2016/2102[7]. New websites must comply with the directive by 23 September 2019, old websites from 23 September 2020, and mobile applications from 23 June 2021.


3.2.3 The Future

While the WAI has done a good job of bringing light to the problem of universal web accessibility, criticism has been expressed towards the guidelines for not being inclusive enough or not always being suitable for all purposes.

The introduction of WCAG 2.0 states that the guidelines will not be able to make websites accessible for people with all combinations of disabilities, particularly in cognitive and learning areas. This is also where much of the criticism towards the guidelines is directed. Partly because of this, but perhaps mostly because of rapid advancements in mobile devices, WCAG 2.1 is, at the time of writing, in development. WCAG 2.1 claims to provide additional guidance for these types of disabilities but cannot provide universal coverage. WCAG 2.1 adds 17 new success criteria, many of which are aimed at tablets and mobile devices.

WAI recommends that websites should have WCAG 2.1 as their new conformance target for improved accessibility and to anticipate future changes in policies. They are, however, also working on a WCAG 3.0 in parallel, a project expected to take a long time and include more extensive changes.

3.3 Web Accessibility and Assistive Technologies

From accessibility research and WCAG, five major groups of accessibility problems and solutions have been identified. These are regarding keyboard navigation, contrast, text presentation, page layout and screen reader compatibility, and text content. The issues and some technical solutions found in theory are presented in this section.

3.3.1 Keyboard Navigation

Many groups of disabled users, such as those with low vision or motor disabilities, are unable to use a mouse for navigating the web [21]. Temporary disabilities such as having a broken wrist may also prevent someone from using a mouse, forcing them to rely on keyboard navigation to carry out daily tasks [22]. Furthermore, many people with various cognitive disabilities may also have problems with mouse navigation [18]. Efficient keyboard access is therefore highly important to make websites accessible for these people. However, most websites are optimized for mouse navigation [21]. Even some of the most popular websites, such as Facebook and Twitter, have had issues with keyboard navigation, rendering people with motor disabilities unable to perform simple navigation tasks, such as skipping navigation, while showing low levels of user satisfaction [23].

The optimization for mouse navigation usually affects keyboard navigation negatively because of mouse users’ preferred structure of web pages. Several empirical studies have shown that users find resources faster in structures that are broad rather than deep [21]—in other words where many links are displayed on one page, with few clicks to get to the destination. These studies have, however, focused on users able to use a mouse for navigation. Keyboard navigation in pages with this layout is usually tedious,
since many links on a page means that the keyboard user must press the TAB key many times to find the right link [21].

There are, however, several methods one can employ to reduce the number of TAB presses a keyboard navigator has to make. One is to add links for page navigation to the top of the page [21]. If content on the page can be organized and divided into logical sections, links to these sections can be added to the top of the page and, upon clicking one of them, the focus is set on the first link in the group. Adding such links, however, might confuse sighted users, and therefore a method needs to exist to show or hide the links depending on the user’s preferences [21]. This can be achieved by adding a script to identify if the user is navigating with the mouse or the keyboard, using the JavaScript events onMouseMove and onKeyDown [21]. For example, if the user has pressed the TAB key a number of times and the mouse has not been moved, the links will be displayed. Another use of such scripting is displaying a link to skip the navigation and jump to the main content when the TAB key is pressed. This method is used in many popular websites of different types (for example, Facebook8, Reddit9 and KhanAcademy10).

Another way to improve keyboard navigation using the TAB key is to modify the TAB-chain with the use of the tabindex attribute. By setting the tabindex attribute of an element to a positive value, that element will receive focus earlier than elements with a tabindex of 0, which are placed into the regular tab index of the page [24]. With the use of this attribute, complex user interface components such as tree controls or tab panels can be navigable with the keyboard [24]. The attribute can also be used to shorten the TAB-chain by excluding multiple links that point to the same target. This is done by setting the tabindex attribute to -1 for such duplicate links [21]. This method should also be used to exclude inactive elements such as graphics from the TAB-chain [21].

### 3.3.2 Contrast

While guidelines for contrast and how text should be presented may be most intuitively associated with visual impairment, other groups also have added needs for perceiving written text. Black text on white background is optimal for visually impaired users since it provides the highest contrast, but this combination is not recommended for dyslexics, as they are sensitive to such high contrast, which can cause words to appear to blur together [25]. Lobier et al. [26] found that dyslexics have a parallel visual processing deficit, resulting in this poor visual attention span. Thus, having easily readable text on websites is especially important for dyslexic people.

WAI themselves have recommendations for different color combinations for people with different needs. They also state that some people need high contrast, that colors

---

8 https://www.facebook.com/
9 https://www.reddit.com/
10 https://www.khanacademy.org/
with high luminance are not readable to dyslexics and that web pages have to work when users attempt to change these colors [27]. While WCAG 2.0 includes a success criterion for minimum contrast ratio, WAI’s recommendation to ensure the possibility of lowering the contrast ratio for those who need to has, as of WCAG 2.0, not been translated into any success criteria.

In addition to the extended needs of visually impaired and dyslexics, children may have additional difficulties perceiving content with bad contrast values. According to a study by Benedek et al. [28], the contrast sensitivity of children increases until at least age 14. A less recent study by Beazley et al. [29] concluded that contrast sensitivity reaches its peak in the age group 18-29.

In a study by Rello, Kanvinde & Baeza-Yates [25], the authors developed a set of layout guidelines for improving the accessibility of web text for dyslexics. These guidelines are based on interviews and eye-tracking tests of twenty-one dyslexics, and are about colors, fonts and spacings. The accessibility practices used are, according to the authors, beneficial not only to dyslexics but to all internet users, including those with other disabilities [25]. For background/text color combinations, off-white/off-black has traditionally been recommended for dyslexic users but, in this study, none of the participants preferred this color combination [25]. Most participants instead chose the black on yellow color combination, but the eye-tracking part of their study showed that this color combination required the longest time for the eyes to focus on. The authors’ explanation is that black on yellow has a very high contrast ratio, which has been said to be bad for dyslexics, but appears to be the most readable at first sight. The color combination that was the fastest to read was black on cream [25].

Evett & Brown [30] have compared guidelines for dyslexics and the visually impaired and found a high degree of overlap. The authors produced a set of text specifications for producing text for both dyslexics and the visually impaired and claim that they should improve readability for all [30]. The resulting recommendations are consistent with the recommendations of Rello, Kanvinde & Baeza-Yates [25]. For dyslexic users, they recommend dark blue on light blue or black on yellow but, for the visually impaired, they recommend black on white. Therefore, they argue that a choice between foreground and background colors should be provided [30].

This guideline for web page customization is frequently cited. In Friedman & Bryen’s [18] compiled guidelines for users with cognitive disabilities, web page customization capabilities should not only include changing the contrast, but also the font size, sound and placement of navigation. According to Evett & Brown [30], the suggested customization capabilities are choosing font style and size as well as background color. Font size and the contrast between elements on a web page are somewhat related, since a larger font size does not require as high of a contrast ratio to be read as smaller font sizes [31].

### 3.3.3 Text Presentation

Regarding the text properties, a study by Rello et al. [32] concluded that an increased font size up to 18 points has a positive effect on readability for dyslexics. A study by
Zorzi et al. [33] showed improved reading ability in dyslexic children when extra-large letter spacing was used. Letter spacing is not included in any WCAG 2.0 guideline but is being added in WCAG 2.1. Rello, Kanvinde & Baeza-Yates’ study [25] showed a preference for even larger font sizes, in contrast to Rello et al.’s study [32]. A majority of the participants of this study preferred the largest font size used in the study, 26 points [25].

Regarding the character, line and paragraph spacing, most participants preferred standard character spacing or slightly more separated characters. The narrower the spaces between both characters and lines, the longer it took to read the passage, according to the study [25]. The recommendations produced by the authors were a line spacing of 1.4 and a paragraph spacing of 2. [25] The optimal column was decided to be 77 characters. Evett & Brown [30] here recommend 60-70 letters per line, extra lines between paragraphs and 1.5-2 line spacing.

Rello & Baeza-Yates [34] found that good fonts for dyslexics are sans-serif and monospaced fonts, with examples being Helvetica, Courier, Arial and Verdana. This is consistent with recommendations from both Friedman & Bryen’s [18] and Evett & Brown’s [30] studies.

In light of the findings above, the accessibility for users with disabilities such as visual or cognitive impairments can be vastly improved by purposefully designing websites according to their needs. One such common design is to include some type of magnifying solution that can be controlled by the users. Many modern browsers today are shipped with such a magnifier. Some websites also offer more extensive solutions with selective magnification. These solutions allow users to enlarge or shrink content based on the users’ needs, as opposed to magnifying the entire page at once. This can better serve the needs of different users, as some may only need to enlarge text in certain parts, while others only need to enlarge certain images or tables. According to a study conducted by Fichten et al. [35], more than two-thirds of visually impaired and blind students use some type of magnifying solution.

The drawback with these types of magnifying solutions is that important meaning or context might be lost in the process, making the interface of the application harder to understand and thus impairing usability [36]. As an example, a picture pertaining to a certain paragraph might end up at the bottom of the page after magnification, instead of next to the intended paragraph. Additionally, long text sections might become cumbersome to read if they are not adequately adjusted after resizing, forcing the user to scroll back and forth horizontally as well as vertically. However, such pitfalls can usually be avoided through thoughtful design choices.

### 3.3.4 Screen Reader Compatibility and Page Layout

A logical and consistent layout for web pages is important for anyone, but certain groups of people with disabilities have extended needs on the layout. For dyslexics, clutter or too many items on a page will worsen their difficulty reading the text [37]. Users with ADHD may have problems focusing on the main content if there are too many distracting elements. According to Friedman & Bryen’s compiled guidelines for
users with cognitive disabilities [18], using an uncluttered, simple screen layout is one of the most frequently cited web design guidelines. The navigation and design of every page should follow the same pattern, and navigation buttons should be clear, large and consistent [18]. Other recommendations they have is to use headings, titles and prompts.

These are also highly important for the visually impaired or others who use screen readers, that is, software designed to read the content of a web page aloud. Results from a study conducted by Fichten et al. [35] show that close to 100% of blind students use screen readers, as well as 50% of students with impaired vision. Screen readers are also commonly used by people with dyslexia [13]. One of the most important design recommendations in Friedman & Bryen [18] is to support screen readers and use alternative text tags. Evett & Brown [30] provide a more expanded set of recommendations for screen readers, including to punctuate after bullet points, number menu items, use as few symbols as necessary and provide a logical read order for tables.

It is clear that the page layout and structure affect accessibility both in terms of what the eye can perceive on the screen, and how well a screen reader can convey the information of a web page. No guidelines in WCAG 2.0 exist explicitly for the former, but there are several guidelines for ensuring screen reader compatibility. Guideline 1.3 says to create content that can be presented in different ways without losing information or structure. Even more closely tied to screen readers is guideline 4.1, which says to maximize compatibility with current and future user agents, including assistive technologies.

With the introduction of HTML5, there are new methods with which a developer can structure the content of a web page to make it more navigable for screen readers. The use of so-called landmarks, which are HTML tags such as nav, main and footer, allows the screen reader to identify these parts of the page, making it easier to navigate. [38]
There are many variants of screen readers on the market today, popular examples include JAWS\textsuperscript{11} and NVDA\textsuperscript{12} \cite{39}. For a blind or visually impaired user who uses a screen reader, of course, some of the content will be lost, namely purely graphical content such as layout and images. However, if an alternative text tag is provided for an image, it will be read by the screen reader as a substitute. This solution, however, has some obvious limitations and the quality of the alternative text decides its usefulness. Unfortunately, the alternative text is often misunderstood or completely left out \cite{36}.

The usefulness of screen readers is not limited to users with visual impairments. On the contrary, screen readers can be used by any user who wishes to take a break from reading and instead listen to the content. According to a survey conducted in October 2017 by the non-profit organization WebAIM\textsuperscript{13} on screen reader users, almost 11\% of the 1792 respondents answered that they had no disability. Among the respondents with some kind of disability, a large majority (75\%) were blind and roughly 20\% were otherwise visually impaired. Furthermore, other groups such as cognitive disabilities, deafness or hearing impairments and physical disabilities were reported. However, none of these exceeded more than 5\% of the respondents. \cite{39}

\subsection*{3.3.5 Text Content}

While the guidelines of text formatting and page layout are of great relevance, one must not forget the importance of considering the text content itself. Complicated language is one of the key problems encountered by dyslexics \cite{25}. The guidelines for dyslexics and the visually impaired compiled by Evett & Brown \cite{30} as well as the guidelines for people with cognitive disabilities compiled by Friedman & Bryen \cite{18} do include some such guidelines regarding the text content.

The top recommendation according to Friedman & Bryen \cite{18} is using pictures, graphics, icons and symbols along with the text, with the second most important one being to use clear and simple text. Both of these also appear in the study by Evett & Brown \cite{30} and are unrelated to the architecture of the web page and instead focus on the text content. Due to the nature of these recommendations, they are difficult to translate into web accessibility guidelines. Therefore, they are missing from current WCAG, even though they have been proven to be so important. To ensure accessibility of a website, both the developers and the content providers therefore must look beyond WCAG.

\textsuperscript{11} https://www.freedomscientific.com/Products/Blindness/JAWS

\textsuperscript{12} https://www.nvaccess.org/

\textsuperscript{13} https://webaim.org/
3.4 Introduction to E-learning

E-learning is defined as “instruction delivered on a digital device that is intended to support learning” [40]. E-learning courses include both information and techniques that help people learn the content, and they are delivered via digital devices using words and pictures. The words can be spoken or written, and the pictures can be static, such as illustrations and photos, or animated like videos. This form of technology-based learning has grown rapidly since the start of the millennium, with an 11% share of technology delivered instruction in 2001 to 39% in 2011-2013 [40]. With rapid advancements in high-fidelity technologies for online interaction [41] and the introduction of Web 2.0, more and more options and opportunities for online learning are developed [42], so there is no reason to believe this trend is going to revert.

3.4.1 Cognitive Theories and How People Learn

Many e-learning courses ignore human cognitive processes. They should be based on cognitive theory of how people learn and research studies concerning e-learning features that best promote learning. According to Clark & Mayer [40], the development of e-learning applications should not have too much focus on technology, because this may result in the role of the learner being ignored. Instead, one should have a learner-centered approach to learning with technology, since this approach is more effective for promoting productive learning [40].

Mayer [43] presented three metaphors of learning, which represent three different perspectives of what learning is. These are strengthening correct responses and weakening incorrect responses, adding new information to your memory, and building a mental representation of the presented material. The first approach involves rewarding correct responses and punishing incorrect ones. The second approach conflicts with a lot of learning theory, in that it ignores psychological engagement [40]. The third metaphor represents the view that people are active sense-makers [40], and learning will not be effective if one attempts to just pour information into the brain of the student. Clark & Mayer [40] find the third approach to be the most fruitful, since effective instruction does not only involve presenting information, but also encourages the cognitive processing of the student.

There are three principles of cognitive science that form the basis for this view. The first is that people have separate channels for visual and auditory material; the second that those two channels have a limited capacity; and the third that learning occurs when engaging in appropriate cognitive processing [40].

In e-learning, the use of multimedia allows for utilization of both channels. In the learning process for multimedia content, there are three sub-processes that take place. The first is the selection of words and images, which occurs when the student pays attention to the material presented to them. The second is the organization of those words and images in the working memory. The third is the integration of the new information with existing knowledge in the long-term memory. [40]
To facilitate these processes, one should consider the capacity limits of working memory. *Extraneous processing* should be limited since it does not support learning [40]. Improving the layout of content displayed by reducing the amount of irrelevant text or pictures will reduce this type of processing [40]. Providing step-by-step demonstrations and placing printed words close to their corresponding graphic are other methods for reducing the level of extraneous processing [40]. *Essential processing* refers to the level of cognitive processing required to comprehend the material. If the content is very complex, it could be beneficial to divide the content into smaller parts or teaching facts separately [40]. Using audio instead of text can also help manage this processing [40]. *Generative processing* is something e-learning environments should aim to increase. It refers to gaining deeper understanding of the material, and this is accomplished by supporting engagement with the material to increase the motivation of the student [40]. Such engagement can be gained by using conversational language, asking students to elaborate, or having them play games [40].

### 3.4.2 Benefits of E-learning

If theories of human learning are applied correctly to the e-learning domain, there are several benefits to be had compared to traditional methods of learning. The potential of customizing the learning experience to each student’s individual needs is one [40] [44]. Clark & Mayer [40] mean customization as in tailoring of content, instruction and navigation. It also includes the benefits of students being able to progress at their own pace in asynchronous learning. Similarly, others highlight the benefit of not being restricted to time and space, leaving time for other commitments [45] [41].

Another benefit is the possibility of using graphics and interactive elements to promote psychological engagement, helping acquire new knowledge and skills. The use of multimedia, the combination of text, audio, images and animations, can also help this acquirement. To take it even further, the engagement can be increased by means of gamification, as in adding elements from games to provide motivating and effective learning experiences. [40] Other benefits include the reduced costs compared to traditional classroom training and reduced efforts of teachers through online examination and automatic grading [45].

### 3.4.3 Hazards of E-learning

While the benefits above indicate a promising future for e-learning, there are hazards, often related to these same benefits, that can occur if any of the features forming the basis for the benefits are overdone. Using too many sounds and animations to deliver the content is damaging to learning, because of the inherent limited capacity of the human brain. At the same time, not using enough features proven to promote learning has the same damaging effect. Such design mistakes include having a wall of text to convey the information without using relevant visuals to provide explanations and having a very low level of interactivity in the e-learning application. [40]

A benefit mentioned above is the added control a learner can be given over their learning but leaving too much control in the hands of the learner might be dangerous.
Not providing any structure to the learning environment rarely works [40], so appropriate guidance needs to be given to the students. Problems stemming from the lack of self-regulation capabilities among students are a threat to the efficiency of online components of learning environments. Huang & Zhou [46] found that these problems are significant among Chinese students, and likely exist in other parts of the world.

In a study by Liaw [47], students believed in the potential of e-learning as an assisted learning tool but, at the same time, they expressed concerns about system quality. The system should provide a good level of interactivity. This is in line with Clark & Mayer’s view [40] that learning will not be effective if the student is simply presented with information; one has to be engaged in appropriate cognitive processing.

In addition to this, an empirical study by Concannon et al. [48] concluded that negative experiences with the e-learning aspect of the course in the study were concentrated on technical problems, highlighting the importance of providing sufficient technical support for the students. However, in the same study, none of the participants expressed any difficulties in using the e-learning environment, regardless of previous computer experience, indicating that a well-designed e-learning course can be used effectively without requiring any generic computer training.

3.4.4 Guidelines for E-learning Applications

Clark & Mayer have developed a long list of guidelines for e-learning, based on a set of principles derived from theories about how people learn. These principles are backed up by extensive empirical evidence. Minding these principles and following the guidelines will improve the student’s ability to learn in e-learning courses.

Multimedia Principle

One of the perks of using computers in an educational context is their ability to convey information in a multitude of different formats. Using a combination of words and pictures rather than using words alone is beneficial to learning. This is the multimedia principle, which is based on cognitive theory and recommends the use of both words and graphics in e-learning applications. This allows students to engage in active learning by connecting both words and pictures. It is, however important to select the right type of graphics to accompany words. Decorative graphics, which purely serve to decorate a page, do little to enhance the information to be learned and should be avoided because processing them still occupies the limited capacity of the visual channel. Instead, the graphics presented in an e-learning environment should be organizational, showing relationships among content; relational, summarizing quantitative relationships; transformational, illustrating changes; or interpretive, concretizing intangible phenomena. [40]
Contiguity Principle

Another principle important in the use of several media to present a piece of information is the contiguity principle. This principle entails that, for example, text and its corresponding graphic should be presented close to each other, in a way that makes it easy to connect the two. If a student must search for the part of a graphic corresponding to a piece of text, they waste limited processing capacity. Following the contiguity principle allows the students to instead use this processing capacity for understanding the material. A common example of a problem related to this principle is in scrolling screens where an illustration is displayed above or below its related text, and the user has to scroll past one to see the other. Solutions for this proposed by Clark & Mayer [40] are using text boxes that pop up when users hover over the graphic or placing graphics alongside text instead of below it. Such integrations of words and images allow for making meaningful connections between them. [40]

Modality Principle

While words are easiest to present in written form, using audio for conveying such information should not be overlooked. If a student is given the possibility of listening to text while looking at graphics related to it, they can utilize both visual and auditory channels, thus processing the information more effectively. This is the modality principle. Presenting the text in spoken form minimizes the risk of overloading the student’s visual channel. The modality principle applies where graphics and commentary are presented simultaneously and is especially important when the multimedia lesson is fast-paced and the graphic is complex, but less so when the material is simple. Since, in some cases, it is not practical to implement the modality principle because of added technical demands and increased development costs, Clark & Mayer recommend only applying the multimedia principle in these situations. [40]

Redundancy Principle

Furthermore, words should only be present in spoken text when there is no on-screen text describing the same thing, and vice versa. This is another principle, the redundancy principle, which states that people learn better from concurrent graphics and audio, rather than if an on-screen text component is also added [40]. This is because attention is shifted from the graphics to the on-screen text. There are, however, situations where redundant on-screen text can be beneficial, such as when there are no graphics or the presentation is slow-paced [40]. This principle opposes the common belief that people have different learning styles, that some prefer to obtain information visually and other auditorily. Clark & Mayer [40] claim that this belief is not supported by the available research evidence, and that it makes unwarranted assumptions about how people learn. They instead refer to the dual channels theory, which suggests that presenting on-screen text in addition to the spoken narration and graphics would put unnecessary strain on the visual channel [40].
Coherence Principle

A principle that is straightforward but commonly violated is the coherence principle. In its essence, it means not cluttering the lesson by adding material that does not support the instructional goal. Content descriptions should be kept short and concise, and stories and trivia should be avoided. Even if it might be tempting to add extraneous words for interest, there is evidence against doing it. This also applies to added words for expanding on key ideas and adding technical depth. Extraneous graphics should also be omitted, since they interfere with the process of sense-making. Simpler, less detailed visuals are better for learning than more detailed ones, and static imagery is better than animated. A certain level of added detail designed to evoke emotions, such as adding facial features to objects, has, however, been proven to improve learning. Extraneous audio such as background music can also disrupt and overload the cognitive system and should therefore be avoided. [40]

Personalization and Embodiment Principles

The personalization and embodiment principles promote learning by evoking emotions in the learner. Using informal, conversational and polite language in first or second person for e-learning lessons gives a feeling of social presence and has been proven to lead to cognitive processing and higher information retention. However, it is also important not to overdo it. The student should feel like the computer is a conversational partner, but the conversation should not be so informal that it distracts from the material. Using polite language is effective with novice learners, while experienced learners benefit from a more direct language. The voice quality of narrations also affects students’ information retention, showing an improvement where the narrator is a human voice rather than a machine voice. [40]

The embodiment principle concerns having a character on the screen teaching the student. Such characters are called on-screen agents. To benefit learning, research has shown that on-screen agents do not have to look real, but they should behave like humans. They should speak instead of having their words written on the screen, and they should use conversational language. [40]

Segmenting and Pretraining Principles

The final principles in Clark & Mayer [40] concern making complex material easier to comprehend by managing essential processing. The segmenting principle entails breaking a complex lesson into manageable segments to avoid overloading the cognitive system. An example of this is an animated sequence that demonstrates how to perform a task. To break it into segments, one could divide the procedure into parts and present them independently, with the user pressing a “continue” button to proceed to the next part. To further reduce the load of essential processing, the pretraining principle can be applied, which involves familiarizing the student with the key concepts of the lesson or the functionality of the virtual learning environment before the lesson. This works to unload cognitive processing by redistributing a part to a pretraining portion. [40]
3.4.5 Asynchronous and Synchronous E-learning

Literature on e-learning often distinguishes between asynchronous and synchronous e-learning content, where asynchronous content is consumed by the student at their own pace, at any time and any place, and synchronous e-learning is more like that of traditional classroom learning, where the teaching is performed in real-time by an instructor. Synchronous platforms include virtual classrooms and webinars [40].

Asynchronous and synchronous e-learning both have their different benefits and drawbacks. Asynchronous e-learning allows students to learn when they have the time, allowing them to flexibly combine their education with other commitments. This asynchronous nature of many online courses is a reason why many people choose to take them [42]. Participants are also usually more thoughtful about their contributions, since they are not expected to reply immediately and therefore have lots of time to think through their answers [42].

The downside of asynchronous e-learning, however, is the risk of the student feeling isolated, as a result of infrequent face-to-face communication with teachers and other students [42]. On the other hand, synchronous e-learning may mitigate this feeling of isolation since the communication feels more like talking, even though there is no face-to-face communication [42]. However, the efficiency and quality of contributions are reduced because of the added pressure of expecting an immediate response [42].

3.4.6 Blended Learning

While studies have failed to show a significant difference between the effectiveness of different media in delivery of learning content, different platforms are better or worse than others in more specific aspects and might be better or worse for different types of students. Computers are one of the most flexible options because of their versatility in types of media they can deliver [40]. As concluded in a study by Sun et al. [49], it is important that students can choose the method of learning that suits them best, so this flexibility of computers should be exploited to accommodate for many different needs and learning styles.

The flexibility does not have to end here, however. The use of computers can be combined with other means of instruction. This is called blended learning and has had slightly different definitions but is essentially the combination of traditional face-to-face learning systems and distributed learning systems, with an emphasis on the role of computer-based technologies [41]. The reasons why one would choose to use a blended learning system are many, but Graham [41] found the three most common reasons were improved pedagogy, increased access and flexibility, and increased cost effectiveness.

While the concept of blended learning is promising, there are challenges to consider when designing a blended e-learning system. The first is the amount of live interaction that should be included in the learning process. There are differing perceptions of the role of live interaction. Some propose that it is primarily used for socialization reasons,
while others highlight how important it is for students to feel part of a learning community [41] [42].

The second challenge arises when online components require a large amount of self-discipline from the students. The learning environment should be designed in such a way that it supports increasing self-regulation capabilities among the students [41]. The organization has to support blended learning environments and the students have to possess the technical skills required to use the platform effectively.

The differences in technology available for different socio-economic classes is another challenge to consider [41]. If a student does not have access to a computer, they cannot take part of the digital aspects of a blended learning environment. Material may also have to be adapted to local audiences to make it culturally relevant [41].

Finally, there is a balance to be struck between innovation and production of blended learning systems, which may prove difficult due to the ever-changing nature of technology [41]. As long as these challenges are considered, blended learning approaches can be adopted rather easily since this kind of environment is consistent with the values of higher education institutions [50].

3.4.7 E-learning for People with Disabilities

Accessibility in e-learning is important not only because of the recent additions in the jurisdiction on accessibility in public procurement but also because of the purpose of the application. In an application where the main purpose is educating its users it is vital that everyone regardless of disability has the same conditions for learning the content and that time that could be spent on learning the content is not spent on navigating the application. Cognitive and learning disabilities may have a larger impact in these applications since the need for long-term information retention is of highest importance. W3C themselves have expressed that even the AAA level of conformance in WCAG 2.0 does not ensure accessibility for people with all types of disabilities, with cognitive, language and learning disabilities being disabilities for which the criteria are particularly lacking [51]. For this reason, efforts have been made by some researchers to develop better approaches for evaluating the accessibility of e-learning applications.

Kelly et al. [44] argued that while WAI has made a significant contribution in raising awareness of the issues of web accessibility, there may be incompatibilities with their guidelines and other interests in more specific contexts such as e-learning. According to the authors, there is a need for a wider perspective for accessibility guidelines when it comes to the domain of e-learning since there are certain limitations of the guidelines as well as difficulties with implementing them [44]. Concerns include difficulties in understanding the guidelines, conflicts between accessibility and usability, and the guidelines being too theoretical [44].

However, much has happened since these concerns were raised. The issues these authors reported were regarding WCAG 1.0, while a version 2.0 has now been in effect since 2008, which has addressed many of the concerns. The technical issues with e-learning accessibility presented in their article from 2004 [44] are partly about
technologies that have since been rendered obsolete and about browser inconsistencies that have since been mitigated.

More interesting and time-relevant are the pedagogic issues affecting e-learning accessibility. In certain scenarios prevalent in e-learning, adhering to the WCAG would not be desirable from a pedagogic standpoint [44]. In a case study by Kelly et al. [44], problems arose when students were required to identify images and connect them to a phrase to demonstrate their knowledge of certain facts. To comply with WCAG, these images would have to have ALT attribute describing the image to people using screen readers, highlighting one of WCAG’s limitations.

This brings questioning to the enforcement of the Accessibility requirements suitable for public procurement of ICT products and services in Europe (ETSI EN 301 549) [4], with the web accessibility part being based on the WCAG 2.0 conformance requirements. Kelly et al. [44] argue that a more appropriate solution would be to make reasonable adjustments according to the Special Education Needs and Disability Act 2001 (SENDA). Such reasonable adjustments could be providing an oral examination as an alternative [44], but such a solution would violate EN 301 549.

For these reasons, the Kelly et al. [44] further argue that a broader, holistic approach should be taken to e-learning. They have developed a model for it, where the accessibility guidelines of WCAG form one part, and other parts taken into consideration are usability, local factors, infrastructure, learning outcomes, and learner needs [44]. Seale [52] agrees that the aim of accessible e-learning should be satisfying the needs of the students, rather than conforming to standards.

3.5 Web Accessibility Evaluation

There are several methods of conducting accessibility evaluations of websites, each with their own benefits and drawbacks. The method described in the following section is an established method developed by the W3C. Furthermore, the section also covers automated testing tools as well as the concept of sampling.

3.5.1 Website Accessibility Conformance Evaluation Methodology

The Website Accessibility Conformance Evaluation Methodology (WCAG-EM) is a conformance evaluation methodology developed by the W3C for evaluating conformance to the three different conformance levels of WCAG, namely A, AA and AAA. The first level (A) is the lowest level. To reach AA level conformance, all A level success criteria must be met, as well as all AA level criteria [20] [27]. The second level (AA) is often recommended and used as a minimum target in accessibility legislations [53], such as in ETSI’s recommendation [4]. Websites that do not reach AA level will be hard to use for disabled users, even excluding disabled users from certain parts of the website.

WCAG-EM is designed to be used by a large group of accessibility stakeholders, including third party web consultants who are employed to analyze and evaluate websites, website owners, developers or researchers [53]. The result of the
methodology, that is, the level of conformance achieved is based on a 5-step iterative process depicted in Figure 2.

**Figure 2:** The five steps of WCAG-EM, adapted from WAI [53].

*Define the Evaluation Scope*

In this first step the scope of the evaluation is determined. Firstly, the target conformance level needs to be decided upon. As an example, if the target level conformance is AA then all AAA level success criteria can be ignored. Furthermore, this step also includes choosing a minimum set of technologies that are to be accessibility supported. Accessibility supported means that users’ assistive technologies are supported when using these predefined technologies [20]. This is necessary since the possible combinations of operating systems, web browsers and assistive technologies are endless, and thus it becomes impossible to provide and ensure support for all such combinations. The WCAG 2.0 are designed to be independent from any particular technology stack [20].

*Explore the Target Website*

After defining the scope, the next step of the evaluation is to explore the target website. The exploration is a first step to understand and determine the current accessibility status of the target website. The goal is not an exhaustive inspection, but rather to explore and identify parts of particular interest for further inspection in later stages of the evaluation.

Parts of interest include common pages such as home page, navigation and footer. These pages are always important since, for example, an inaccessible home page might render an entire website inaccessible because it is the entry point to the rest of the
website [54]. The exploration also identifies pages that can be classified as different types. Different types could mean web pages with a certain design or layout, or web pages with varying types of content such as forms, tables or multimedia. Web pages can also be classified according to specific coding styles or templates used when generating the page.

Additionally, essential functionality that can be performed by users of the target website, as well as all identified web technologies that are relied upon, such as HTML or CSS, should be noted during the exploration. Lastly, any additional web pages that are specifically relevant for the accessibility of the website should be noted for further inspection [53].

Select a Representative Sample

When performing an accessibility evaluation, it is often not feasible to perform an exhaustive evaluation since websites often are comprised of a very large number of web pages [36]. Furthermore, modern web technologies also introduce state and dynamically generated content that is rendered differently for different users which further complicates any system evaluation.

As a result, it becomes necessary to choose parts of the website that can adequately represent the entire website for the evaluation. The process of selecting a subset of web pages is called sampling, in the WCAG-EM this process is divided into three steps. First, all web pages identified in the previous exploration step are added to the sample as a baseline called the structured sample. Second, a random selection of web pages is added to the sample. WCAG-EM recommends that the number of randomly selected pages should be 10% of the total number of pages included in the structured sample. The randomly selected web pages are added to the evaluation sample to ensure that the structured sample is sufficient. [53]

The third and final step of the sampling process identifies all web pages that are part of a process from the sample previously selected. For example, a page showing the results of a quiz is part of a process where the user first answered a set of questions. For each such process, the starting point of the process and all subsequent steps up to the sample page are identified and added to the sample. [53]

Audit the Selected Sample

This step aims to audit the sample selected in previous steps. First, all web pages from the sample that are not part of a process, for example the home page, should be evaluated according to the five conformance requirements of WCAG 2.0 described in section 0. Then, all complete processes present in the sample should be evaluated according to the same requirements. Here, attention must be paid to certain web page components, such as forms and input elements. In addition, user interaction feedback associated with the process, such as input confirmation or error messages, should be thoroughly checked.
After the audit of the structured sample is complete, the randomly selected sample is evaluated according to the same process. The results are then compared to the results of the evaluation of the structured sample. Should any discrepancies between the two results exist, the selected structured sample cannot sufficiently represent the website under evaluation. Provided that is the case, the sampling process needs to be redone to include the newly identified web pages. As a result, all subsequent steps of the evaluation must be repeated as well. However, given that no discrepancies are detected, the structured sample is assumed to be complete and the evaluation can continue to the last step. [53]

Report the Findings

The final step of the evaluation is carried out throughout the evaluation process. The outcome of each preceding step is recorded to ensure reliability and replicability of the evaluation. The granularity level of this documentation depends on the context and needs to be agreed upon before the evaluation. Both evaluation commissioners, that is, the person or organization who ordered the evaluation, and evaluators depend on this documentation to support any claims made in the result of the evaluation. [53]

To summarize, the evaluation methodology deals with (1) selecting a representative sample, (2) auditing the selected sample and then (3) ensuring that the sample was sufficient by comparing the results to a random sample. The audit is the most complicated step of the process. The WCAG-EM explicitly states that performing the audit requires deep understanding of the WCAG 2.0 and expertise in the web accessibility field [53]. Although WCAG 2.0 was designed to be testable, previous studies have shown that novice evaluators achieve poor results when performing evaluations of web pages according to the WCAG [55] [56]. However, studies also show that even experts often fail to agree on conformance evaluations, especially for certain WCAG [57]. Furthermore, although WCAG-EM is effective at finding accessibility problems, it is less effective at distinguishing between important and unimportant problems [58]. Nonetheless, WCAG-EM is one of the most widely used evaluation methods and the method is relatively cost effective, especially when combined with automated testing [58].

3.5.2 Automated Testing Tools

The first version of WCAG was commonly criticized for lacking testability [59], in response to this WCAG 2.0 associated success criteria to each guideline that must be met in order to fulfill the guideline as a way of improving the testability. Automated accessibility validation tools have been around ever since the first version of WCAG, but the applicability of such tools has increased following the update of the guidelines [54]. However, these tools initially received a lot of critique for not being a viable evaluation alternative since the nature of WCAG didn’t allow for automated evaluation [60].

An attempt to classify the WCAG according to degree of testability was made by Centeno et al. [60]. They categorized the guidelines into four groups, namely;
Objectively automated, Subjectively automated, Semi-automated and Manual. Centeno et al. then proceeded to theoretically categorize each of the 65 guidelines of WCAG 1.0, resulting in 16 guidelines being classified as Objectively or Subjectively automated, 31 as Semi-automated and 18 as Purely manual [60].

Much of the critique received initially still holds true. Some guidelines in WCAG 2.0 are ambiguous and require subjective interpretation, thus different evaluation tools produce different results for the same website. Furthermore, as with WCAG 1.0, some guidelines can only be partly tested using automated tools while others require complete manual testing [60]. The result being that the coverage, that is, the number of testable success criteria, is less than 50% for such fully automated tools [59].

Despite the drawbacks and limitations with automated accessibility evaluation it remains the most common evaluation method [54]. This is partly explained by the fact that automated testing is affordable and fast [61]. Furthermore, it eliminates the need for expert evaluators [59].

In order to assess the performance of different automated tools, and to perform comparisons, a set of three metrics has been suggested by Brajnik [62]. The first metric is completeness, defined as the number of accessibility defects discovered by the tool, relative to the actual number of accessibility defects present in the website [62]. Vigo et al. [59] adopts the same definition of completeness, with the addition that the ground truth is defined as the number of accessibility defects discovered by accessibility experts through manual inspection. In other words, completeness describes how well a tool reduces false negatives while maximizing true positives. A false negative is in this case an actual accessibility defect not discovered by the tool [62].

The second metric is correctness, meaning the tool’s ability to minimize false positives, while at the same time maximizing the number of true positives [62]. False positives generate noise in the evaluation reports, which must be manually inspected before they can be discarded. In some cases, such inspections might not be feasible, and therefore the errors are incorrectly treated as true positives [59]. It should be noted, however, that false positives cannot be completely avoided while simultaneously achieving acceptable completeness [62].

The third metric defined by Brajnik [62] is specificity, defined as the number of different issues that can be detected by a tool. However, the author also mentions that this metric is not necessarily related to tool effectiveness beyond the context set out in the article [62].

Relying solely on automated tests for accessibility evaluation might lead to unwanted consequences [59]. Several common problem areas exist for automated evaluation tools. As mentioned before, some of these are due to the nature of WCAG [60]. One such example is the use of alternative texts, required for all non-text content. An automated tool can only detect the existence of alternative text. However, only through manual inspection can the quality of the text be ascertained [36]. Nonetheless,
automated tools can be effective when evaluation accessibility of the web, especially if used in conjunction with manual inspection methods such as WCAG-EM [44].

Another potential problem area for automated evaluation is the detection of keyboard traps. To verify that no such traps exists requires extensive simulation or real interaction [59]. Additionally, the increasingly dynamic content of many web applications today makes automated testing even more complicated [36] [54].

There are many different automated tools available on the market today, both commercial and free alternatives. Many are available as online applications or as browser extensions, making evaluation of the target website easily achievable in the browser itself. A recent meta study on web accessibility evaluation methods showed the usage rate of automated evaluation tools in research papers on evaluation methodologies. This revealed that the five most frequently used automated evaluation tools were AChecker, followed by TAW, EvalAccess, WAVE and Total Validator. [54]

In the previously mentioned study by Vigo et al. [59], the authors attempt to compare different automated tools using a benchmarking model. The tools chosen for the comparison were AChecker, SortSite, TotalValidator, TAW, Deque and AMP. Two of these, namely SortSite and AMP are desktop applications, while the rest are online applications. The benchmark model included the above-mentioned metrics completeness, correctness and specificity [59].

Deque (henceforth referred to by its product name Axe-Core), received an overall high score in the study conducted by Vigo et al. [59]. The tool performed consistently and relatively well for all areas of the study without performing poorly in any particular area [59]. Axe-Core is used as a browser extension and performs evaluations directly on the displayed website. In instances where Axe-Core detects a possible issue but is not certain that it actually is an issue, the evaluator gets a prompt stating which problem was found, where, and because of what reason. The evaluator can then manually inspect the instance to confirm whether it was an issue or not. Like many other evaluation tools have difficulties with dynamic web applications, so does Axe-Core. However, in instances where other evaluation tools might fail the criterion or pass by the issue completely, Axe-Core has a number of fail-safes to prevent this from happening and instead allows the evaluator to determine the outcome.

3.5.3 Sampling

In the following section the concept of sampling is expanded beyond the approach suggested in the WCAG-EM mentioned in section 3.5.1.

When performing an accessibility evaluation of a website it is often not feasible to evaluate the entire website, regardless of method [36]. Instead, sampling is performed to select certain parts to evaluate. The sample selection method can be completely random, however knowledge about the website can increase the completeness (also called accuracy [63]) of the evaluation, meaning that a larger number of actual accessibility issues is detected. Furthermore, certain pages are especially important for
accessibility. Research shows that the homepage is the most important page being the entry point to the website [54].

Aspects that need to be carefully considered during the sampling process are sample size and evaluation metric, which have been shown to be tightly coupled with the choice of sampling method [63]. For websites that are composed of less than 50 web pages an exhaustive evaluation can be performed [64]. However, such small websites are uncommon, and modern dynamic web applications and websites often have a much larger number of web pages and states, making sampling crucial.

As previously stated, the sample size affects the completeness of the evaluation. This is especially true for conformance evaluation metrics such as WCAG-EM [63]. The benefits of an effective sampling method is apparent for small sample sizes, as a selective sample selection strategy, focused on identifying diverse types of web pages, outperforms an ad-hoc or a page-by-page strategy [36] [63].

When selecting the size of the sample the first thing to consider is the size of the website under evaluation. A larger website usually requires a larger evaluation sample [53]. However, the consistency (or diversity) of the website is sometimes an even more important factor to consider when selecting the size of the sample. The number of accessibility problems often correlate more directly to the consistency of the website rather than the total number of web pages, meaning that a website with many different types of content will often include more accessibility issues [36]. Lastly, the precision of the evaluation, or the required level of confidence in the evaluation will eventually affect the required sample size [36].
4 Method

This chapter describes how the theoretical foundation for this thesis was built, how features aimed at solving the problems found in the e-learning application were implemented, and how the results were evaluated.

4.1 Pre-study

This study was preceded with a literature review for gathering relevant research and defining the context for this thesis. The fourth edition of the book *E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning* by Ruth C. Clark & Richard E. Mayer was found to be the most comprehensive, updated and well-cited book on the topic of e-learning, and therefore it was chosen to serve as a foundation for the e-learning theory. Theory on accessibility was collected from several sources, but the book *Web Accessibility: A Foundation for Research* provided a comprehensive view on the topic and served as a starting point. These books were supplemented by research articles delving deeper into specific concepts or providing more empirical data. The articles were found through the Linköping University e-Library together with the Google Scholar search engine, using keywords such as *e-learning and disability* and *e-learning and accessibility*. While there are plenty of publications on e-learning, accessibility and disability, there is a limited number of publications covering the intersection between the three topics, for example [1], [44], [52], [65] with the most notable contributions coming from author Jane K. Seale.

Even fewer publications are closely related to specific accessibility guidelines, and none were found to include a technical approach on how to handle the accessibility problems arising for disabled people in e-learning applications. This led to a further search for technical solutions for accessibility issues. The relevance of the articles was evaluated according to their relation to the keywords used, the publication date as well as the number of citations. Most articles were taken from journals and conference proceedings, with one journal, *Computers & Education*, supplying many of the articles. The statistics used for finding disabilities and issues found in a school environment were taken from Skolverket (the Swedish National Agency for Education) and a compiled statistics document from Post- och telestyrelsen (the Swedish Post and Telecom Authority), among other sources.

The research was concluded by comparing and evaluating different accessibility evaluation methods in order to find which combinations would be achievable to use and give the most accurate and at the same time efficient evaluation given the authors’ previous level of experience with the topics. Preceding the implementation phase, an evaluation of the accessibility of the baseline application was performed in order to identify problems and areas in which the accessibility of the application needed to be improved. Details of the evaluation process are found in section 4.3.
4.2 Implementation

Once the theoretical foundation had been established, and the accessibility of the baseline applications had been evaluated, features aimed to solve the identified accessibility problems were implemented. This section describes the approach, starting with the prototyping and ending with the implementation details of the feature development.

4.2.1 Prototyping

Proposed solutions for the identified accessibility problems were first illustrated in a functional wireframe prototype, created with the wireframe tool Balsamiq\textsuperscript{14}. A wireframe is a guide suggesting the structure and relationship between pages in an interface [66]. Each failing success criterion was connected to one, or a set of, wireframes aimed at showing the proposed solution to that particular problem.

The prototypes were presented to the company and accessibility expert and co-founder of Axess Lab Hampus Sethfors in order to increase the credibility of the solutions, as well as getting suggestions for modifications and an expert opinion. The prototypes were altered according to the input and suggestions received at the presentation and served as guidelines for the feature implementation. An example of what the prototype creation process and the Balsamiq wireframe tool looks like is illustrated in Figure 3.

\textbf{Figure 3}: Balsamiq wireframe tool

\textsuperscript{14} https://balsamiq.com/
4.2.2 Feature Development

All accessibility features developed in this project were implemented in a copy of the original application’s source code, obtained at the beginning of the project. A description of the baseline applications can be found in the Background chapter. In total, four consecutive weeks were dedicated to feature development. The repository management tool GitLab\textsuperscript{15} was used both for version control, issue-tracking and organizing the project workflow.

After the first evaluation, each success criterion categorized as failed was added as an issue in GitLab. These issues were then organized using the built-in feature Issue Boards that can be used to categorize each issue according to its respective workflow status. For a project of this size, three boards were deemed sufficient, representing the different statuses To Do, Doing and Done. Furthermore, bugs hampering accessibility discovered in the original application were added to the To Do board upon discovery, as well as bugs introduced as new features were implemented.

All features were developed and integrated directly into the copied version of the source code. However, the accessibility toolbar was initially developed and tested in a separate application used solely for that purpose. Near the end of the development phase, the toolbar was integrated with the rest of the implemented solutions in the copied version of the source code, leaving enough time for testing and finding solutions for new bugs that had been introduced.

During the development phase, accessibility solutions affecting screen reader compatibility were tested using the Swedish version of JAWS (version 2018.1804.26)\textsuperscript{16}. The free version of the software was used. Although there are no functional limitations to the free version, there is a 40-minute time limit that requires a restart of the software once 40 minutes has elapsed. Testing with JAWS was done to ensure that the implemented solutions had the desired effect on user experience for screen reader users, rather than merely meeting the minimal requirements of WCAG.

4.3 Evaluation

An evaluation suite of the applications’ accessibility was performed twice. Once on the baseline applications in the beginning of the project in order to identify the accessibility problems and once after the implementation of the accessibility solutions to confirm that the problems were solved. The evaluation suite was based on the WCAG-EM and was complemented by the automated accessibility testing tool Axe-Core.

First, the scope of the website, conformance target and accessibility support baseline were defined according to the WCAG-EM. The conformance target was set to AA

\textsuperscript{15} https://gitlab.com/

\textsuperscript{16} http://www.freedomscientific.se/Download2.htm#JAWS
according to the company’s wishes, and the scope included all web applications under the domain for the company’s web application library.

Second, the library of web applications was explored, one application at a time. Whenever a common web page, essential functionality or a new type of web page was identified, it was added to the corresponding list. As more of the web applications were explored, the frequency of new web page types decreased as expected, since they are built using the same templates. When every web application had been explored, the web technologies relied upon were listed.

Third, a structured sample reflecting all identified common web pages, essential functionality, web page types and relied upon web technologies was chosen. Since the evaluators had access to the source code, it was known that every web application was built using a rather small set of templates. To get good coverage on the evaluation, every template would have to be used at least once. The coverage of the selected sample is because of this knowledge of the source code estimated to be good, even if the sample turned out to be rather small. When the structured sample had been chosen, the source code of the generated static HTML pages was compared to that of the templates, ensuring every template was represented.

The structured sample ultimately included 18 web pages. Then, a random sample of 2 web pages, or ~10% of the structured sample, was selected according to WCAG-EM by randomly generating which link on the page to click and the number of clicks from the start page.

According to the WCAG-EM all web pages that are part of a process should be included in the sample as well. However, since all pages belonging to a series presenting a complete process were already included in the selected sample, these 18 + 2 web pages formed the entirety of the sample. Descriptions of the pages in the sample are found in Appendix B.

In the fourth step of the evaluation the selected structured sample was audited according to the five WCAG 2.0 conformance requirements. Axe-Core was used to quickly find some of the accessibility issues in the sample. It was found that this automated tool was particularly good at spotting flaws in the HTML, such as missing ALT attributes and duplicate IDs, as well as insufficient color contrast between elements on the pages. Axe-Core also reports best-practices, in addition to failures to meet WCAG success criteria. These suggestions were noted for future implementation, but not mentioned in the conformance review.

Since automated tools cannot find all accessibility issues, each of the sample pages were manually inspected, while covering each of the success criteria not covered by the automated tool. Manual inspection involved a wide variety of techniques simulating user interaction with the web pages, such as navigating using the keyboard or interacting with elements such as buttons and forms since this cannot be achieved by automated tools. Failures to meet a certain success criterion were reported for each failing page if there was a discernible difference between the elements causing the
failure. In other words, issues with a certain element appearing in multiple pages were only reported once.

The fourth step was concluded by auditing the random sample according to the same requirements used for the structured sample. Both results were then compared to assure that no discrepancies between the results existed.

In the fifth and final step of the evaluation, the outcomes of each of the steps in the evaluation process were documented using W3C’s own WCAG-EM report tool.
5 Result

This section starts with the results from the accessibility evaluation of the baseline applications, continuing with a more informal evaluation of the e-learning qualities of the applications. The section further lists the accessibility issues found related to specific WCAG success criteria and the measures taken for solving them. WAI’s descriptions of the individual success criteria are found in Appendix A. The section also lists accessibility improvements made unrelated to specific WCAG success criteria. Finally, the results of the evaluation of the updated applications are presented.

5.1 Accessibility Evaluation of Baseline Applications

Using the evaluation methodology outlined in the method chapter, an accessibility evaluation was conducted on the baseline applications prior to any accessibility solutions being implemented in order to establish the initial accessibility status of the applications. The results produced from the evaluation set the foundation for the solutions described in the prototype, which were subsequently implemented in the applications.

As stated previously, the conformance target was set to level AA in the first step of the evaluation, meaning that all AAA level success criteria were disregarded in the evaluation and therefore not included in the result. By utilizing a combination of automated tools and manual inspection to perform the evaluation on the selected sample, the status of each A and AA level criterion was categorized as either passed, failed, not present or cannot tell. The distribution of success criteria that were categorized as either passed or not present is shown in Table 1. Additionally, success criteria in the table are divided by conformance level and the accessibility principle they belong to.

Table 1: Passing WCAG success criteria in baseline applications

<table>
<thead>
<tr>
<th>Principle</th>
<th>Level A</th>
<th>Level AA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Perceivable</td>
<td>3 / 9</td>
<td>2 / 5</td>
</tr>
<tr>
<td>2. Operable</td>
<td>7 / 9</td>
<td>2 / 3</td>
</tr>
<tr>
<td>3. Understandable</td>
<td>3 / 5</td>
<td>3 / 5</td>
</tr>
<tr>
<td>4. Robust</td>
<td>0 / 2</td>
<td>0 / 0</td>
</tr>
<tr>
<td>Total</td>
<td>13 / 25</td>
<td>7 / 13</td>
</tr>
</tbody>
</table>

The table indicates that 20 of the 38 principles under consideration were categorized as either passed or not present in the original application. For the purpose of determining the overall conformance level of a system, a success criterion categorized as not present is equivalent to passed. As an example, if a system does not include any
live stream video content, success criterion 1.2.4, which states that all live audio and video content should have captions, can be considered as passed.

Furthermore, the evaluation categorized 16 success criteria as failed and another 2 were categorized as cannot tell. Both criteria categorized as cannot tell were related to audio content in the applications and the content. This content could not be fully determined before the first evaluation. Furthermore, potential alternatives to the audio content needed to be examined further before labelling the success criteria as either passed or failed. A comprehensive table of the result from the evaluation of the baseline applications showing the result for each success criterion individually can be found in Appendix D.

5.2 Adherence to E-learning Principles

The web applications are efficient in terms of the three types of processing outlined in Clark & Mayer [40] since they use very basic assignments and layouts, limiting extraneous processing. Content is not complex, and it is divided according to book chapters, with multiple short assignments per chapter, also limiting the amount of essential processing required to perceive the material. Generative processing that should be increased [40], is also promoted by using the web applications as a complement to the course literature. The students are engaged by interacting with the material, which may improve their learning.

The way in which the web applications are included as a complement to course literature and classroom teaching allows for effective utilization of several benefits of e-learning, while circumventing many of its hazards. The applications are never used to convey large amounts of information, as this is better done in the classroom. The occasional chapter summaries present are presented as bullet points. They have a high level of interactivity without distracting elements, stimulating learning by engaging the student in appropriate cognitive processing. The e-learning issue of lack of self-regulation among students is mitigated because the web applications serve as a complement to course literature and traditional classroom teaching. It also solves the issue of live interaction in e-learning and blended learning environments.

Having access to a computer is essential for being able to take part of digital aspects of a blended learning environment, and differences in available technology for students was proposed as another challenge by Graham [41]. This ties in with web accessibility, since it means that the web should be for everyone. However, this issue might be non-existent for Swedish students who are the intended market for these web applications. According to statistics from the Internet Foundation in Sweden [67], 99% of Swedish children aged 12-15 use the internet at home, while 98% use it in their mobile phone. In total, it is reported that 100% of children aged 12-15 use the internet.

For Swedish children between the ages of 4 and 11, the percentage of children who have used an e-learning application is between 69% and 80% for each age group, and 91% of 11-year-olds have used the internet for school work. For Swedish students, it is
therefore safe to say that a vast majority have both the technical skills and the technology at home to use these web applications as intended.

The web applications adhere reasonably well to the principles outlined in Clark & Mayer. Beside the educational videos present in some of the web applications, there is a lack of multimedia implementations for conveying information in a multitude of different formats, to improve learning with the multimedia principle. However, this might be a result of the web applications being more focused on tests and practice, rather than teaching, which is done in the classroom.

The spoken chapter narration is an example of effective application of the modality principle, allowing the student to look at the pictures in the course literature, while having the text read to them. This allows them to utilize both the visual and auditory channels. It is also a great example of blended learning, where e-learning is combined with more traditional teaching methods. However, this example also ties in with the redundancy principle if the student were to read along the spoken narration, which is not recommended by Clark & Mayer [40] as it puts unnecessary strain on the visual channel. The simple layout and straightforward content of the web applications also satisfies the coherence principle. The content of the web applications is divided according to corresponding book chapters, with short assignments for each chapter, avoiding overloading the cognitive system and satisfying the segmenting principle.

In conclusion, the baseline web applications are better aligned with e-learning theory than with accessibility theory, which is to be expected since the content creators are experts in the learning domain, but inexperienced in accessibility. Therefore, there is a reason to focus efforts on improving the accessibility, rather than the already effective e-learning implementations.

5.3 Solving WCAG Level A and AA Accessibility Issues

The following section describes the accessibility problems discovered in the first evaluation which caused a total of 18 success criteria to be marked as failed or cannot tell in the baseline applications. Additionally, for each failed criterion and other identified issues, a solution is suggested together with an explanation of the implementation details of that solution. The different types of pages appearing in this section are briefly described in Appendix B.

5.3.1 Non-text Content (1.1.1)

Problems

Two different issues related to this success criterion were found in the baseline applications. Axe-Core found missing ALT attributes on several images. Firstly, images that purely served a design purpose, such as different icons, were missing ALT attributes. Every image, regardless of purpose, needs to have an alternative text for the web page to conform to this criterion. In this example, most of the icons, that serve only a design purpose, should be ignored by screen readers.
Secondly, in “What’s in the picture” quizzes, the images representing the questions were also missing alternative text. This specific issue can be considered exclusive to the domain of e-learning, and there exists no trivial solution for making this type of quiz accessible to those who cannot perceive the image visually. If one were to write an alternative text describing the image, the purpose of the test would be invalidated. This criterion therefore has an exception for this type of tests, stating that the text alternative “should at least provide descriptive identification of the non-text content”.

**Solutions**

If an image is missing an alternative text, screen readers have difficulties interpreting it. For the image to be ignored by the screen reader, omitting the alternative text is not a viable solution. Instead, images that should be ignored by screen readers need an empty ALT attribute:

```
<img alt=''
```

The images in the “What’s in the picture” quiz could be solved from a WCAG perspective, by adding the alternative text:

```
<img alt='for the assignment'
```

Resulting in screen readers reading “image for the assignment”. From a user perspective, however, no solution exists. A user who cannot visually perceive the image, can never perform this type of assignment, and the creator of the material must be responsible for providing other types of assignment for these individuals.

**5.3.2 Audio-only and Video-only (Prerecorded) (1.2.1)**

**Problems**

Some of the web applications contain pages where the student can have chapters from the book read to them by a narrator. According to this criterion, all prerecorded audio should have an alternative that presents equivalent information, unless the audio is a media alternative for text and clearly labeled as such.

**Solutions**

Since the pages in which audio files could be played use the audio as an audio alternative to chapters in the course literature, the solution for satisfying this success criterion is as simple as adding a paragraph in the markup clarifying that the audio narration can also be read in the course book.
Figure 4: Audio-only and video-only (prerecorded) example

While these pages were used for the purpose of narrating chapters in the current state of the web applications, it would be possible to use the template for other purposes, such as listening comprehensions. Then, this simple solution would no longer be valid, and methods of transcribing the audio files would have to be investigated.

5.3.3 Captions (Prerecorded) (1.2.2)

This success criterion is very similar to 1.2.1 for audio content, but instead of any alternative for the time-based media, captions have to be provided for all prerecorded audio in synchronized media. This criterion however also excludes content that is clearly labelled as an alternative for text, and thus, the solution for the web applications is the same as for 1.2.1.

5.3.4 Info and Relationships (1.3.1)

Problems

The intent of this success criterion is to preserve the visual structural cues available to sighted users even as the presentation format changes. In other words, no information about relationships between objects on a web page for example should be lost to blind or visually impaired users using a screen reader. This also includes information that is conveyed through the layout of the web page.

The word selection page, illustrated in Figure 5, exemplifies a violation of this success criterion. Firstly, although not explicitly visible to sighted users, the layout of the page
is produced using HTML table syntax to achieve the row and column style layout for each word, the corresponding circular toggle-button and the translation of the word. The desired layout structure is successfully achieved using this technique. However, for users using a screen reader, the table syntax results in redundant information being conveyed to the user, such as the current row and column.

![Example Assignment](image)

**Figure 5:** Example of a word selection page

Secondly, it is obvious for sighted users to determine which word the toggle-button belongs to through the page structure. However, this information is lost for screen reader users because each button itself contains no programmatically determinable information about its functionality or the word it belongs to.

**Solutions**

The first problem, that is, page layout being achieved using table syntax, was solved by rewriting the page layout using div elements. This was done while still maintaining the same layout as before and the div elements were aligned using CSS properties such as width and display.

The second problem was solved by adding a unique identifier to each span element containing a word to be translated. Then, the identifier was linked to the corresponding toggle-button using the aria-labelledby attribute on the button element. As a consequence, the screen reader will announce the word to be translated as the corresponding button receives focus.
5.3.5 Use of Color (1.4.1)

Problems
A problem related to this success criterion was found by manual inspection in the multiple-choice quizzes, where upon selection of an option, the button briefly turns red or green depending on if the answer was correct or incorrect, before automatically redirecting the user to the next question. Users who have difficulties perceiving these colors or distinguishing between them, such as people suffering from red/green color blindness, will not understand if their answer was correct or not.

Figure 6: Use of color example problem

Prototype Solutions
The solution proposed in the prototype involved loading an icon into the button that was selected, representing a correct or incorrect answer. Displaying an icon along with the color change to convey the information satisfies this success criterion.

Figure 7: Use of color example prototype

Implemented Solutions
The functionality of the implemented solution is similar to that of the prototype, but instead of dynamically loading a X or a checkmark into the button itself, it was decided to display it to the side of the button to make the design choices consistent with other quiz implementations in the web applications. The correct/false icons are always present but hidden from view and from screen readers until the user has selected their answer. This implementation also simplified screen reader adaptations further explained in section 5.4.2.
Several issues regarding the minimum contrast ratio of the text and background elements of the web applications were found with Axe-Core, although many of them were easily identified without the use of any tools. The background colors of the applications consist of one primary and one secondary color for each web application. The color scheme of the web application is derived from the color scheme of the corresponding printed book. In most of the web applications however, the secondary color in particular did not have a conformant contrast ratio with the white text and icons. Since each web application has its own color scheme, there are over 300 colors that must be checked for conformance and possibly altered to pass the success criterion.

**Prototype Solutions**

The prototype solution featured two separate adjustments. The first being altering each of the non-conforming colors by making them lighter or darker while maintaining the same hue. For this, a tool would be created to calculate the correct colors.

The second adjustment involved utilizing the functionality of the same tool to allow the user to increase or decrease the contrast to their preference by lightening or darkening the background colors of the application. This adjustment was first intended to be implemented as buttons in the accessibility toolbar, increasing or decreasing the contrast in increments. The toolbar prototype is found in Figure 9.
**Implemented Solutions**

In order to easily alter a non-conforming color combination into a conforming one, and for the company to use in future development of color schemes, a tool capable of lightening or darkening colors, calculating the contrast ratio between the colors, and making a non-conforming background color conformant with the inputted text color was developed. Figure 10 demonstrates how the tool can be used to make a non-conforming background color conformant with the selected text color with the press of a button.

The tool takes input colors in hexadecimal, R,G,B format, or by HTML color names. Conversion between hexadecimal and R,G,B is done with string parsing and bitwise manipulations. Translating color names to hexadecimal is as simple as creating a map with keys representing the color names, for example “lightgreen”, and its corresponding value representing the hexadecimal color value, “#90ee90”. Functions for converting colors of the different formats are shown in Appendix C.

The colors can be made lighter or darker using the sliders. A button also exists to make the background contrast conformant with the text color. Both features use the same function for altering the lightness of the color, which is found in Appendix C.
The contrast ratio between two colors is calculated by first calculating the relative luminance of the two colors using the formula:

\[ L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \]

where R, G and B are defined as:

for each \( C \) in \( R, G, B \):

\[ if \ C_{sRGB} \leq 0.03928 : C = \frac{C_{sRGB}}{12.92} \]

\[ else : C = ( ( C_{sRGB} + 0.055 ) / 1.055 )^{2.4} \]

and \( C_{sRGB} \) is defined as:

\[ C_{sRGB} = C_{8bit} / 255 \]

The contrast ratio is subsequently calculated using the formula:

\[ (L_1 + 0.05) / (L_2 + 0.05) \]

where \( L_1 \) is the relative luminance of the lighter of the foreground or background colors, and \( L_2 \) is the relative luminance of the darker of the foreground or background colors. [68] JavaScript functions for above calculations can be found in Appendix C.

The contrast tool was used to quickly check and adjust the primary and secondary colors of the web applications to conforming ratios, and the company was advised to use it in future development of color schemes. Because the primary and secondary colors themselves did not always have a high contrast ratio between them, white borders around buttons and icons were added for improved visibility.

Considering that dyslexics and the visually impaired have different needs for contrast [30] and the prevalence of such users, coupled with the fact that children’s contrast sensitivity is not fully developed until adolescence [28] [29], and the reasoning that children using an e-learning application should not be expected to be able to change their browser’s settings for changing the contrast, it was deemed useful to extend the contrast functionality beyond the requirements of WCAG. The ability to change the contrast was therefore implemented as a native feature of the web applications.

The prototype solution of being able to change the contrast in increments could be helpful for those who need a lower, or a very high contrast to perceive the content. However, after consultation with Hampus Sethfors, it was soon realized that most users would benefit more from a set of options for text and background color, hand-picked based on research on text perception of individuals with different types of visual and cognitive impairments. The options chosen were a default mode that maintains the original, but still conformant colors, a black on white mode, a light on dark mode, or “night mode”, a dark blue on light blue mode, and a black on yellow mode. The black on white mode is useful for users with very low vision. The light on dark mode, while useful for anyone not wanting to put a strain on their eyes, is particularly important for users with light sensitivity [30]. The dark blue on light blue, and black on yellow modes are useful for dyslexic users [30]. The different modes are illustrated in Figure 11.
Figure 11: Contrast selection
The different modes were implemented by dynamically adding classes to elements subject to contrast changes, overriding their default color values. Since the baseline applications relied on many edge cases and dynamically loaded content, the integration of the contrast option functionality was non-trivial. Elements such as icons for correct/incorrect answers with green/red background for example, were judged to better be excluded from the color changes. Such elements were given the class “exclude”, and the selector of the elements subject to contrast change would exclude elements with this class.

It was not sufficient to change only the color of the text and background however. The color of the icons would also have to be changed and given a border of the same text color. New sets of SVG icons with black and dark blue colors were created by editing the SVG’s stroke and fill properties. The new icons were put in separate directories, but maintained the same names, allowing for switching between them using the function found in Appendix C.

The functions for changing contrast were called on the contrast buttons of the accessibility toolbar, but also when new content is loaded into the page, to ensure that also dynamically loaded content receives the correct colors.

5.3.7 Resize Text (1.4.4)

*Problems*

Manual inspection detected that while content scales as expected with the CTRL++ and CTRL+- keys, text did not always scale properly with the browser settings. For this reason, and because of the notion that the target audience of the web applications, who are children, should not have to bother with such browser settings, the problem would be solved with functionality built into the accessibility toolbar. This functionality would scale only the text, as opposed to all content as with the CTRL++ and CTRL+- keys.

*Solutions*

The prototype solution and the implemented solution for this success criterion were very similar. Three buttons exist in the accessibility toolbar for increasing, decreasing and resetting the text size. The minimum and maximum values for the text size were set to 80% and 400%. It was decided to allow the user to go beyond the 200% success criterion of WCAG to allow for users with more severe visual impairments to perceive the content. A 400% text size increase allows for great magnification while not breaking much of the content or functionality. It was ensured that there was no loss of content up to a 200% text size increase, to conform to the success criterion. In some instances, some content would appear off-screen when the text size increase came closer to the 400% setting due to non-responsive practices used in the baseline applications.

Each text element along with its default text size was stored in an array to allow for proper text scaling for each element. The array is updated when new text content is loaded to ensure that dynamically loaded content also receives the right text size setting. The text size setting is stored as a multiplier to be multiplied with each element
in the array upon changing the text size in the accessibility toolbar. An example of the text size functionality is found in Figure 12.

![Image](image.png)

**Figure 12**: Text size example

### 5.3.8 Keyboard (2.1.1)

#### Problems

Manual inspection detected an issue where in many places in the web applications, buttons or other interactive elements could not receive keyboard focus. One example was in the multiple-choice quizzes, where the buttons for selecting an answer could only be clicked and navigated with a mouse, rendering everyone relying on alternative navigation devices unable to complete the quiz.

#### Solutions

Upon further investigation of the code, it was found that the cause of these navigation issues was due to incorrect use of HTML elements. Buttons did not always use the button element, but instead a div or even a span element. The solution in both the prototype and the implementation would be to change the elements to their proper types. In the few instances where changing the type of element would cause irreversible damage to the styling and functionality of the element, the element was given a role="button" and a tab index to allow for keyboard focus without compromising functionality relying on the element being of a certain type.
5.3.9 Bypass Blocks (2.4.1)

Problems

Axe-Core detected an issue where there existed no method for skipping repeated content of the pages of the web applications. While there was no navigation bar to speak of in the baseline applications, rendering a “skip navigation” button useless, the content should still be divided into sections to improve navigability for screen readers.

Solutions

One of WAI’s methods for how to meet this criterion is by using ARIA landmarks, which can tell screen readers the role of the section, allowing such users to skip to that section. An ARIA landmark for the main content looks like the following.

```html
<div role="main"></div>
```

Since the introduction of HTML5, however, there are a number of landmark elements in the form of HTML tags, intended to replace div tags with assigned ARIA landmarks [38]. Such a HTML5 landmark for the main content instead looks like the following.

```html
<main></main>
```

These landmarks provide an arguably clearer structure than their ARIA counterparts and were therefore used to divide the sections of the pages of the web applications in the prototype as well as the final implementation. The landmarks used were nav, main and footer.

Since the accessibility adjustments introduced an accessibility toolbar with a significant number of buttons, it was decided that a “skip navigation” button was to be added to the beginning of the expanded accessibility toolbar, to relieve keyboard navigators from having to tab through all the buttons of the toolbar to get to the main content. This was implemented with an onclick event on the button, giving the element with id #main, the main landmark, focus. To give a normal div or a landmark focus, it needs to have a tabindex attribute. Since it is not intended to receive focus when simply tabbing through the application, the tab index was set to -1.

5.3.10 Focus Visible (2.4.7)

Problems

Manual inspection detected an issue in several places, where even if an interactive element could receive keyboard focus, it was lacking an indicator for such a focus. An example was in the selection of words to practice in a vocabulary quiz. The buttons for selecting all words, deselecting all words, starting the quiz, as well as the radio buttons for selecting or deselecting a specific word could all receive keyboard focus, but there was no way of telling that they had focus. This rendered selecting words with the keyboard very difficult and caused a failure of this success criterion.
Solutions

The implementation for a solution for this success criterion is straight-forward and the same as that of the prototype. Interactive elements in the baseline applications had a rather vague onHover effect where the text within the button received an underline property. To make the onHover and onFocus effects more visible, a box shadow effect was added in addition to the underline to all buttons, and the effects were made consistent through all interactive elements within the web applications. The effect is illustrated in Figure 13.

![Figure 13: Two options in a multiple-choice quiz, with option A selected](image)

5.3.11 On Input (3.2.2)

Problems

Issues were detected by manual inspection in multiple choice quizzes and text input quizzes where the user was redirected to the next question without much warning or feedback on the answer. If context is changed upon changing a setting of an interface component, the page violates this success criterion. Since in the multiple-choice quiz, there were buttons redirecting the user to the next question, it was not deemed to be a violation of WCAG, since pressing a button does not change a setting of an interface component [69], while the text input field of the text input quizzes does.

However, there were still limited opportunities to reflect on the answer in the multiple-choice quiz. In the text input quiz, upon submitting an incorrect answer, a “Next” button appeared along with the correct answer, allowing for such reflection. If the user were to press the TAB key from this point, expecting to navigate to the “Next” button, they would automatically be redirected to the next question instead of changing the focus to the “Next” button, causing an unexpected context change, violating this success criterion. Upon submission of a correct answer, the user would be automatically redirected. The design of the input field was also confusing, acting as both text input and a submit button (see Figure 14), which makes it unclear to the user which actions will submit the answer.

Solutions

The prototype solution for the multiple-choice quiz was to add a “Next” button for the user to press after answering, allowing them time to reflect on the answer. This would satisfy the success criterion and would also be in line with the segmenting principle of
e-learning but would be an annoyance to a majority of the users who do have enough time to reflect on the answer in the one second window before the user is redirected. One of the main purposes of these types of quizzes, perhaps the vocabulary type of quizzes in particular, is to be able to practice the words many times in quick succession. Requiring another button press after each correct answer would defeat this purpose. This is an example of where this type of interactivity inherent to e-learning could conflict with accessibility guidelines depending on the implementation details.

Another proposed solution was to add an option in the beginning of the quizzes to enable or disable the automatic redirection. For the final implementation however, it was argued that everyone should have a similar experience when performing this quiz, regardless of disability. Therefore, the “Next” button was kept in the text input quiz when the user submitted an incorrect answer but was left out in other instances. Instead, an effort was made to improve the feedback given to different categories of users, such as those relying on a screen reader. These adjustments are detailed further in section 5.4.2. Furthermore, to satisfy the success criterion, the submit button was separated and distinguished from the input field, only redirecting the user upon pressing the button. The issue where a TAB press resulted in a redirection was solved by altering the event listener for these elements. The design of the input field before and after the adjustment is shown in Figure 14.

![Text input quiz before and after](image)

**Figure 14:** Text input quiz before and after

### 5.3.12 Consistent Navigation (3.2.3)

**Problems**

Another issue found by manual inspection was that some pages had a button directing back to the layer above in the hierarchical structure of the web applications (see Figure 15). For example, in some of the assignment pages, a link was found to the assignment’s chapter. In others, this link did not exist, possibly confusing some users.
Figure 15: Application hierarchy

Solutions
The implementation for this success criterion matched that of the prototype. The effort of conforming to the criterion involved simply making the navigation consistent by adding a backwards link to any views where they were missing, as shown in Figure 16.

Figure 16: Backwards navigation button from example assignment

5.3.13 Labels or Instructions (3.3.2)

Problems
Using automated testing tools, it was discovered that pages that contained form input elements, such as the text input quiz and the accounting quiz pages, lacked programmatically determinable labels. Thus, the purpose of the input fields become difficult to understand for screen reader users as the screen reader only announces a generic default description for a text input field.

Solutions
The two pages used to exemplify the problem in the above description received two slightly different solutions. Firstly, the solution implemented in the text input quiz page is essentially the same as in section 5.3.4 Info and Relationships (1.3.1), although the problem is different. However, by adding the aria-labelledby attribute to the text input field and linking it to the element containing the question to be answered, both the problem of unspecified relationships and the problem of missing labels are solved.

Furthermore, for the accounting quiz page, each text input form received a static label using the attribute aria-label. The label describes the purpose of the field, that is, either debit or credit.
5.3.14  Parsing (4.1.1)

Problems
Axe-Core detected duplicate IDs in some applications. For example, alternatives in multiple-choice quizzes were assigned an ID to be able to compare the answer to the correct one. However, these dynamically loaded alternatives were each given the same ID, violating this success criterion.

Solutions
The solution to this problem varied slightly from case to case, but the general solution involved moving the ID attribute to the parent element of the answers. Furthermore, the correction functions were changed to check for the parent element instead of the individual answers when retrieving the question ID.

5.3.15  Name, Role, Value (4.1.2)

Problems
The last success criterion categorized as failed in the evaluation of the baseline applications was success criterion 4.1.2. Briefly, this criterion states that all interactive user interface components should have programmatically determined names, roles and values. Otherwise, assistive technologies such as screen readers cannot convey this information to the users.

Several violations to this success criterion were found using both automated tools and manual inspection. First, Axe-Core flagged all toggle-buttons in the word selection page (see Figure 5, page 38) for lacking discernable names. This can also be interpreted as a violation to both success criteria 1.3.1 and 3.3.2.

Second, the state of the toggle-buttons, that is, either pressed or not pressed, can only be determined by visual inspection of the page since the state is not programmatically determined. Thus, for blind or visually impaired users using a screen reader, the states of the toggle-buttons are unknown. This also constitutes a violation to the Name, Role, Value success criterion.

Third, through manual inspection it was discovered that on several occasions, buttons were dynamically disabled or enabled depending on user interaction with the page. As an example, if all words in the word selection page are selected, the “Select All” button will become disabled (see Figure 17). This was done by appending a class called “inactive” to the button element, which in turn caused a visual indication that the button was disabled. Furthermore, it also disabled the onClick event listener for the button element. However, since the indication was only visual, blind or visually impaired users had no way of knowing if the button was disabled or enabled.
Figure 17: Disabled button example

Solutions

The first problem was solved by adding labels to the toggle-buttons using the aria-labelledby attribute, as in the solutions for success criteria 1.3.1 and 3.3.2. A more detailed description can be found in section 5.3.4.

For the second problem, the solution was to programmatically determine the current state of the toggle buttons. This was done using the attribute aria-pressed. By default this attribute was set to true for each button, indicating that the buttons are pressed and thus reflecting the initial state of all toggle-buttons. Subsequently, the attribute was dynamically set to either true or false depending on user interaction with the toggle-buttons. Consequently, assistive technologies such as screen readers can determine the current state of the buttons using the aria-pressed attribute.

The solution to the third problem was similar to that of the second problem, only for this solution the attribute aria-disabled was used instead. This attribute was dynamically set to either true or false depending on the state of the button. The programmatically determined state could then be interpreted by assistive technologies, that in turn can convey the information to sightless users for example.

5.4 Auxiliary Accessibility Improvements

In some instances, simply ensuring conformance to the A and AA levels of WCAG 2.0 did not ensure an accessible application. Guidelines for aiding people with certain cognitive disabilities are lacking in WCAG 2.0 [51], and from testing with assistive technologies, it was found that WCAG failed to identify certain issues regarding screen reader compatibility. The following sections describe additional accessibility improvements implemented where AA level WCAG 2.0 conformance was decided not sufficient for the purpose of the applications.

5.4.1 Visual Presentation

Due to the prevalence of dyslexic users [13] [15] [16] and their improved reading abilities with larger line heights [25] and letter spacing [33] [25], as well as legislation [4] and the importance of equal access and satisfying the needs of the students, rather than conforming to standards within the e-learning domain [44] [52], functionality for increasing the line height and letter spacing was implemented in the accessibility toolbar in addition to the contrast, text size and skip navigation functionality.

The functionality was implemented as toggle-buttons, allowing the user to toggle between the default line height and letter spacing and a larger line height or letter
The specific values for the larger line height and letter spacing were 5 points of letter spacing as concluded beneficial to dyslexics by Zorzi et al. [33] and 2 em line height as recommended by Evett & Brown [30]. The prototype only included controls for changing the line height, which could be changed incrementally with plus and minus buttons. This design was changed to the toggle buttons in favor of usability and due to space constraints in the accessibility toolbar.

The changes were applied to all elements that were not given the class “text-exclude”. This exclusion class was necessary since certain elements would change aspect ratio with increased line height. An example of this would be the circular icons that would become oval-shaped. Elements without this class would upon pressing respective buttons receive classes that increased the line height and letter spacing properties respectively. The entire navbar was excluded from both text and contrast changes for usability purposes, perhaps at the expense of accessibility. The line height and letter spacing features are shown in Figure 18.

![Figure 18: Large line height (left) and large letter spacing (right)](image)

These additions to the accessibility toolbar give it an arguably complete, native solution for allowing to change the visual presentation of the applications, since it achieves conformance to the AAA level success criterion 1.4.8 Visual Presentation. This success criterion consists of five requirements, the first being that foreground and background colors can be selected by the user, achieved by the contrast functionality. The second requirement of having a width of no more than 80 characters can be satisfied since text
is wrapped properly when rescaling the window. Text is not justified, satisfying the third requirement. The fourth requirement is related to line spacing which is satisfied by this part of the solution, and the fifth requirement is similar to the AA success criterion of resizing text, which is also satisfied.

5.4.2 Additional Screen Reader Adaptations

A critical accessibility issue affecting screen reader users was discovered in the multiple-choice quizzes and text input quizzes, not related to any WCAG success criteria. As mentioned in section 5.3.11, these quizzes were designed to allow users to answer several questions in a rapid succession. This was achieved by automatically redirecting the user to the next question after an answer was submitted. During the brief time period before being redirected to the next question, the user was provided with a visual feedback indicating if the submitted answer was correct or wrong. However, as this feedback was only visual, blind or visually impaired users using a screen reader had no way of knowing the result of their answer.

To remedy this problem, an empty div element was added to the underlying templates of the quizzes. Using CSS, the div was kept visually hidden from the screen without hiding the element from assistive technologies. The attribute aria-live was then added to the div with the “assertive” property.

```html
<div class="visually-hidden" aria-live="assertive" id="ariaAnswerContainer"></div>
```

Elements with the aria-live attribute will be monitored by assistive technologies such as screen readers, and if the content of the element changes it will be announced to the user. The content of the div was then updated dynamically using jQuery after an answer was submitted, reflecting if the answer was correct or incorrect. Consequently, the assistive technology monitoring the aria-live element will then provide auditory feedback to the user announcing the result of the answer.

5.5 Accessibility Evaluation of Updated Applications

After the proposed solutions to the accessibility issues discovered in the initial evaluation had been implemented, another accessibility evaluation was performed on the updated applications. In essence, the procedure of the final evaluation was identical to that of the initial evaluation to ensure comparability. However, an additional page (Text input multi-quiz, see Appendix B) was added to the structured sample that was not yet discovered when the first evaluation was conducted.

Nevertheless, the previously defined scope of the website and the conformance target for the evaluation were left unchanged. Likewise, the process of exploring the web application library and selecting the structured sample was the same as for the first evaluation. This resulted in the same sample as in the first evaluation, except for the text input multi-quiz page mentioned earlier that was now discovered due to a deeper knowledge of the applications’ content structure as well as the source code of the entire web application library.
Again, the fourth step of the evaluation was an audit of the entire sample using a combination of automated tools and manual inspection. This resulted in a categorization of each success criterion into the four categories previously stated. Table 2 shows the status of all success criteria categorized as either passed or not present following the evaluation of the updated applications.

Table 2: Passing WCAG success criteria in updated applications

<table>
<thead>
<tr>
<th>Principle</th>
<th>Level A</th>
<th>Level AA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Perceivable</td>
<td>8 / 9</td>
<td>4 / 5</td>
</tr>
<tr>
<td>2. Operable</td>
<td>9 / 9</td>
<td>3 / 3</td>
</tr>
<tr>
<td>3. Understandable</td>
<td>5 / 5</td>
<td>4 / 5</td>
</tr>
<tr>
<td>4. Robust</td>
<td>2 / 2</td>
<td>0 / 0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>24 / 25</strong></td>
<td><strong>11 / 13</strong></td>
</tr>
</tbody>
</table>

All 10 success criteria previously marked as passed in the initial evaluation received the same verdict in the second evaluation. Beyond the criteria that were marked as passed in the evaluation of the baseline applications, an additional 15 success criteria were categorized as passed in the updated version of the applications. Among these, 13 success criteria were previously marked as failed and 2 were previously marked as cannot tell.

In summary, 35 out of the 38 A and AA level success criteria were marked as passed or not present in the second evaluation, and none were marked as cannot tell. Consequently, 3 criteria were categorized as failed in the updated applications. The success criteria that failed are 1.2.3 Audio Description or Media Alternative (Prerecorded), 1.2.5 Audio Description (Prerecorded) and 3.1.2 Language of Parts. Since 1.2.3 is classified as a level A criterion the applications do not fulfill all level A criteria. Thus, the applications do not reach level A conformance. A more comprehensive status of the success criteria in the updated applications can be found in Appendix E.

5.5.1 Success Criteria Marked as Failed

The updated applications, although greatly improved, still contained three success criteria that failed the accessibility evaluation. The following section shortly describes the reason for failure for each success criterion.

**Audio Description or Media Alternative (A, 1.2.3)**

The intent of this success criterion is to provide blind or visually impaired users access to visual information presented in synchronized media such as video content. The W3C states two different ways of passing this success criterion, either by providing a file with
audio description that complements the video content, or by providing a text alternative to the video content that contains all information, both visual and auditory. However, the educational videos included in the applications lack such audio descriptions, and no text alternative is provided. Therefore, this success criterion was marked as failed in the evaluation.

*Audio Description (Prerecorded) (AA, 1.2.5)*

As in success criterion 1.2.3, this criterion requires that all video content be supplemented with an audio description to allow blind or visually impaired users to access the content. In addition, this stricter level AA criterion does not recognize text alternatives as a sufficient solution to this problem. Regardless, this criterion fails for the same reason as success criterion 1.2.3.

*Language of Parts (AA, 3.1.2)*

In order to pass this success criterion, each passage or part of a web page in any other human language than the default language of that web page must be programmatically determined. Otherwise, assistive technologies such as screen readers cannot determine the current language and will therefore default to the main language of the web page, resulting in words being pronounced incorrectly and in other ways making it hard for blind or visually impaired users to understand the content. The programmatically determined default language for all web applications in the study is Swedish. However, the application library contains several web applications used for the purpose of learning a secondary language, such as English, Spanish or German. Naturally, these applications will contain words in another language than Swedish to some extent, sometimes only single words and sometimes whole phrases or passages. The vocabulary quiz for example will present a Swedish word and ask for the translation in that specific language. However, the secondary language of these parts is not programmatically determined to be the correct language, but rather defaults to Swedish, and therefore will not be interpreted correctly by any assistive technology used.
6 Discussion

This chapter discusses the results produced, methods used, and the thesis put in a wider context.

6.1 Results

This section discusses the usefulness of presenting results of accessibility improvements in terms of WCAG conformance. It further discusses alternatives to some of the implemented solutions and elaborates on the relationship between accessibility and e-learning theory.

6.1.1 Usefulness of WCAG Conformance

If the baseline applications and the updated applications are judged strictly according to their WCAG conformance level, it appears that the applications have not been improved at all. Despite implementing solutions for 15 failing success criteria, the applications still fail to achieve level A conformance. Two of the criteria still failing in the updated applications are related to the video content present in the web applications. Possible solutions to these criteria include providing a media alternative or producing alternative audio tracks for every video. However, both solutions are unsolvable for developers and expensive and time consuming for content creators.

Thus, companies already possessing a large library of videos may find it inconceivable to produce alternative audio tracks for every video. This may result in them having no incentives to solve any of their accessibility issues, knowing that they cannot reach any level of WCAG conformance. However, if WCAG conformance status was calculated as a quantitative score instead of fixed conformance levels, more accurately reflecting the true accessibility status of the evaluated website, this problem could be mitigated.

6.1.2 Alternative Solutions

This section discusses how certain decisions about functionality and design could have been made differently.

Contrast Functionality

The contrast tool developed for quickly altering colors to be conformant with WCAG has both pros and cons. It can be used to quickly make a color AA conformant with any other color, or adjusting it to any desired degree, allowing for color combinations with very high or low contrast to be made from two initial colors. If more time was dedicated to this project, the contrast tool could also have been integrated in the company's production script, alerting the content creators if the contrast ratio of the inputted colors is inadequate and prompting them to choose a different color.

However, very light colors that are altered to become darker, or very dark colors altered to become lighter, will appear dull and desaturated. While the contrast tool works for every combination of colors, from a design perspective, it works best when the colors to be altered are already close to conformant, or at least are relatively saturated. Thus,
it is important to create good color schemes from the beginning, if one wants to ensure the visual appeal of the web page. For example, if one were to use white text as in the web applications in this study, yellow background colors have to be avoided, since they will never have a good contrast ratio with white. If they are darkened with the contrast tool in order to become conformant with the text color, they will look muddy and dull.

These cons spawned another idea on how to make any web page conformant with success criterion 1.4.3. There exists a small range of colors that are AA conformant with both black and white. Specifically, those with a relative luminance of between 0.175 and 0.183.

Consider the contrast ratio formula:

\[
\frac{(L_1 + 0.05)}{(L_2 + 0.05)}
\]

where \(L_1\) is the relative luminance of the lighter of the foreground or background colors, and \(L_2\) is the relative luminance of the darker of the foreground or background colors. [68]

For a color to be conformant with black, it needs to have a luminance \(L\) of:

\[
\frac{(L + 0.05)}{(0 + 0.05)} \geq 4.5 \Rightarrow L \geq 0.175
\]

For a color to be conformant with white, it needs to have a luminance \(L\) of:

\[
\frac{(1 + 0.05)}{(L + 0.05)} \geq 4.5 \Rightarrow L \leq 0.183 ...
\]

For a color to be conformant with white and black, it needs to have a luminance \(L\) of:

\[
L \geq 0.175 \quad L \leq 0.183 ...
\]

This implies that any given colour will be AA conformant with at least one of the colors white or black. What this means is that one convenient solution for solving any non-conforming contrast ratios, would be to change the text color of each non-conforming element to either black or white. To black if the relative luminance of the background is more than 0.183, and to white if it is less than 0.175.

This solution may be beneficial to some, especially if the text on the web page is already black or white, as the original background colors can be maintained. However, just above 4.5:1 is the maximum contrast ratio that can be achieved with this method in some instances, and a seemingly arbitrary choice of color for text elements may not be optimal for design and consistency purposes. The bottom line is that both solutions are temporary, and conformant color schemes should be created from the start.

**Design Choices**

Which icon to use for symbolizing accessibility for the expandable accessibility toolbar has been a topic of controversy. How will the users understand which tools are hidden in the menu? The icon used in the prototype hinted at visual or auditory disabilities, but such a distinction should probably be avoided in order not to put labels on people or exclude anyone from using the tools. The symbol for universal access, resembling a person with their arms stretched out, avoids this problem, but the question is how
many people understand what it means, and which tools they expect to find underneath it. The solution that was finally implemented used a generic settings icon, without having the icon reveal that the settings have anything to do with accessibility. On hover or focus of the icon, however, the word “Accessibility” is shown to the user, explaining what is hidden in the menu to people interested in changing the settings, while not being an eye sore for people who are not. Ultimately, tests with actual users would have to be undertaken in order to confirm which icon is the most intuitive and inoffensive.

Likewise, the labeling of the different contrast options were decided to straightforwardly describe the text and background colors, instead of hinting to which type of impairments they were developed for. While the options are useful for users identifying with certain groups, there can be many reasons why people prefer to read text with certain colors and contrast values. Labeling these options as being for sufferers of certain disabilities might exclude those having other reasons for their contrast preferences.

### 6.1.3 Accessibility and E-learning

To not only produce an application conformant with the WCAG criteria, but also ensure that the nature of the applications and characteristics of its users are considered, the applications have also been looked at from an e-learning perspective. While the e-learning properties of the applications of this study only affected the accessibility implementations in a few cases, other comparisons between the accessibility and e-learning perspectives were made during the research for and implementation of this project. This section presents discussion points on possible overlaps and incompatibilities between the accessibility and e-learning perspectives.

**Overlap Between E-learning and Current WCAG**

While some guidelines pertaining to principles in the work by Clark & Mayer [40] are difficult to relate to accessibility, a significant overlap was found between WCAG and the coherence and segmenting principles.

Guideline 1.3 of WCAG 2.0 [70] ensures that third party software can be used to display content in a simpler layout, to be more easily perceivable by some users. This can be related to the coherence principle of e-learning, which for example says to avoid irrelevant graphics, stories and excessively lengthy text [40]. This idea of not having too much clutter on a web page is reflected in the added needs of users with for example ADD or dyslexia, who can have a difficult time perceiving content if there are too many impressions at once [37], and is one of Friedman & Bryen’s [18] most important guidelines for people with cognitive disabilities. There is however a question of how these third-party solutions for constructing a simpler layout considers graphics and animations on a web page. WCAG might not be able to limit the number of distracting elements on a web page, which therefore has to be the responsibility of the e-learning content creators.

The segmenting principle of e-learning says to break down content into smaller chunks [40]. This relates to both text paragraphs and interactive exercises, the latter being
inherent to e-learning and further states that learners should be given options to progress at their own pace [40]. This principle should be extra important for users with ADD or other attention deficit disorders, as it also relates to Friedman & Bryen’s [18] guidelines of using simple text. Guideline 3.2 of WCAG [71] regarding navigation and not initiating unexpected context changes can be related to this principle. Also, while WCAG cannot prevent walls of text or specify the maximum length of a paragraph, long passages of text could be made more readable by using a larger line height or other means of changing the visual presentation, which is translated into success criterion 1.4.8 [72]. This reveals that the intentions are the same, while the methods implied by e-learning and accessibility guidelines are sometimes different. When the issue is related to text content, it is not, and might never be a part of WCAG.

**Incompatibilities Between E-learning and Current WCAG**

There also seem to be some areas in which interpretations of e-learning and accessibility guidelines might be conflicting.

First of all, while both e-learning and accessibility theory agree that both visual and auditory channels should be used to convey information, the reasoning behind the statement differs between the two perspectives. The multimedia principle of e-learning says that audio and video should be used together to convey information in a way that promotes learning [40], while the accessibility requirements related to multimedia are more focused on providing alternatives for people who cannot access information conveyed in a single format. The multimedia principle cannot be utilized if either the visual or auditory channel of a user is limited, which is where WCAG steps in and makes sure the user can perceive the content at all. If a video is made accessible to a blind person by providing narration of what is happening, while simultaneously presenting the normal audio track, it risks overloading the user’s auditory channel, highlighting difficulties disabled people can face even if the web page is made accessible.

In the web applications of this study, the chapter narration and educational videos allow for utilization of both the visual and auditory channels. Video tutorials, the parts of e-learning applications that most often utilize both channels simultaneously, are where the perhaps greatest difficulties in conforming to WCAG arise, and account for two of the three WCAG level A and AA success criteria still marked as failed in the web applications of this study. Since multimedia is such an important aspect of e-learning [40], this problem cannot be ignored. Many e-learning content providers may not have the resources to produce descriptive audio tracks for all their video content, especially not for the large number of videos they may already have produced. Furthermore, developers hired to solve the accessibility issues of a web application can do no more than recommend which efforts that should be made to solve these issues.

One does not have to look far to realize how common video tutorials, lectures and lessons are in the domain of e-learning. Websites offering online courses are becoming increasingly popular, and most of them rely on video content for teaching (for example
KhanAcademy\textsuperscript{17}, Udemy\textsuperscript{18}, OpenLearning\textsuperscript{19} or even tutorials uploaded to YouTube\textsuperscript{20}). Such content seems difficult to make accessible, but it is important that it is not avoided since video is such a convenient format for teaching. Software exists today to automatically generate captions from audio, and with advances in machine learning, it may soon be possible to process video content in a similar manner, automatically generating audio descriptions of what the video is showing. Until then, there seems to be no simple solution to this problem.

Somewhat related is also the redundancy principle, which implies that a student should not be able to choose which media to present the information for learning purposes \[40\]. This is a direct conflict with accessibility guidelines that would disallow disabled people from using their only means of perceiving information. Satisfying both perspectives is seemingly impossible.

The contiguity principle is another principle that can be difficult to adhere to for some users. A common violation of this principle is when scrolling on a page, text and graphics are separated in a way that one comes before the other, and the user has scrolled past the first before seeing the next \[40\]. While Clark & Mayer \[40\] suggest using fixed screens to solve this, this method is outdated and could be solved with a responsive design. For example, keeping the text and picture on top of each other when a device is in portrait mode to accommodate for that aspect ratio, while putting them beside each other for screens in landscape mode. This solves the problem for some users, and works with scaling of content, but if a user has to increase for example the text size significantly, they will still have to scroll, and the picture will be separated from the text, no longer adhering to the contiguity principle. In other words, use of functionality specified in guideline 1.4.4 of WCAG will much likely not be compatible with the contiguity principle.

A more specific issue was identified in theory and also applied to the studied web applications. Kelly et al. \[44\] identified an accessibility issue in e-learning where students were to identify images and connect them to a phrase. This issue was also found in the e-learning application analyzed in this thesis, where in language courses aimed at a younger audience, questions of the type “What’s in the picture?” were asked. Having an ALT attribute describing the image would defeat the purpose of the assignment for people using screen readers, and one could ask if such assignments could ever be accessible by the visually impaired. Conformance to WCAG could be achieved by adding ALT attributes describing the purpose of the picture, but that would not be of much help to anyone, and an alternative exercise would have to be provided.

\textsuperscript{17} https://www.khanacademy.org/
\textsuperscript{18} https://www.udemy.com/
\textsuperscript{19} https://www.openlearning.com/
\textsuperscript{20} https://www.youtube.com/
for the visually impaired, since they still cannot perform the task even with the ALT attribute. This highlights one possible limitation of WCAG for the e-learning domain.

**E-learning Aspects Missing from Current WCAG**

Beyond WCAG’s inability to control the content of a website as previously mentioned, there is an important criterion, regarding text spacing, that is still missing from WCAG 2.0. The text spacing is considered important in an e-learning context since the ability to increase it can improve dyslexic users’ ability to understand and retain the information. In an effort to improve accessibility for different cognitive disabilities, WAI is working on WCAG 2.1, which contains success criterion 1.4.12 Text Spacing [72] for this exact purpose.

While WCAG has come a long way since its first version, there are still some areas in which the guidelines might contradict theories of e-learning, or where users with some cognitive disabilities are not fully considered. It has to be mentioned that it is difficult to give all users the same level of usability of a system, and many principles of e-learning are made more difficult to adhere to if the user suffers from some disability. An example is the contiguity principle that can be made impossible by scaling text or other content. Another is the multimedia principle, which cannot be taken advantage of if the user exhibits limitations in either the visual or auditory channel.

Furthermore, WCAG can only control the layout of a web page to a certain extent, while they cannot do much about the content of an application. This text content aspect might be extra important to the domain of e-learning due to the inherent need for retaining the information presented on a page. This proposes a question of how far one should stretch the definition of accessibility. WCAG can allow anyone to perceive words written on a web page, but they cannot, and probably should not be able to ensure that the content can be understood by everyone. This is an issue of pedagogy and is beyond the competence of developers or accessibility experts. It should rather be the responsibility of the authors of course literature or e-learning content.

### 6.2 Method

This section discusses the limitations of the methods used for this thesis, as well as the replicability, reliability and validity of the specific method implementation. Furthermore, the sources used for the theoretical background of the thesis are also evaluated.

#### 6.2.1 Limitations

Since evaluation of WCAG 2.0 conformance was a prerequisite for this thesis project, the evaluation methods chosen had to be well suited for this purpose. As a result, accessibility evaluation methods that were specifically designed for evaluating conformance to other guidelines were not considered. The method ultimately chosen was the WCAG-EM, described in detail in section 3.5.1. Furthermore, an automated testing tool was chosen to facilitate the audit step of the WCAG-EM. Among the multitude of available options, the tool Axe-Core was selected.
The WCAG-EM was, like WCAG, developed by the W3C. The method provides a detailed step-by-step process of evaluating a website for WCAG conformance. Out of the five steps, the audit step is perhaps the most critical for the outcome produced by the method. The WCAG-EM explicitly states that this step requires deep understanding of WCAG 2.0 and expertise in the web accessibility field [53]. Moreover, studies have shown that novice evaluators achieve poor results when evaluating for WCAG conformance [55] [56], suggesting that novices struggle to interpret the intended purpose of WCAG success criteria.

However, even experts often fail to agree on conformance evaluations [57], indicating that the success criteria can be ambiguous and require subjective interpretation. The WCAG 2.0 often receives critique along these lines, although this was perhaps more characteristic for its predecessor WCAG 1.0 [54] [59] [60].

In addition, the WCAG-EM has been criticized for being ineffective at distinguishing between important and unimportant problems [58]. This inability might be partly due to the fact that the method does not require any testing on users. Instead, involving users in the audit step of the process is listed as optional [53]. However, if the method would require user testing, it would further increase the resources and expertise required to conduct an accessibility evaluation in accordance with WCAG-EM.

On the other hand, using an automated testing tool to assist in the auditing step of the evaluation could help reduce the need for expert evaluators. Then again, only about half of the WCAG success criteria can be tested using automated tools [59]. The automated testing tool chosen to facilitate this evaluation, Axe-Core, performed consistently and well with regards to completeness and correctness in the benchmarking study conducted by Vigo et al. [59]. Axe-Core performed particularly well on correctness, that is, to minimize the number of false positives. In other words, the accessibility issues reported by the tool was indeed true issues in 93% of all cases.

For this reason, high confidence was placed in the results produced by Axe-Core, and no further checks were performed for certain success criteria if the tool did not identify any issues. In summary, Axe-Core did not identify as many types of accessibility issues as the manual inspection. Axe-Core seemed to perform well in this study in identifying issues related to static pages and limitations of the HTML and CSS of the rendered web pages. It was however not good at identifying issues had when completing a process within the applications, and it did not find issues that required interaction with the web page, such as trying to navigate the page with a keyboard or hovering over elements.

Other tools have instead been shown to perform better in terms of completeness but lack in correctness. [59] This trade-off needs to be carefully considered when choosing which tool to use and the tool’s characteristics must be considered when working with the selected tool. If another tool had been selected for this evaluation, without the proper adjustments, the outcome of the evaluation might have been different.

Selecting the sample to be audited is another important factor that will eventually largely impact the outcome of the accessibility evaluation. Knowledge of the application
under evaluation can both increase completeness of the evaluation and at the same time decrease the size of the sample. [63]

The sampling method described in WCAG-EM was used for this evaluation. Using the methodology, coupled with the evaluators’ knowledge of the applications’ structure due to access to the source code, resulted in a relatively small sample with high completeness. However, as already mentioned, one previously undiscovered type of web page was added to the sample for the second evaluation and one was removed since it was discovered that it was represented twice in the sample. Altogether, access to the source code affected the final composition of the sample, ultimately affecting the outcome of the evaluation. Thus, the outcome of the evaluation may have been different without access to the source code.

6.2.2 Research Quality

The reliability of the evaluation methods used is of great importance since the passing of specific WCAG success criteria has been used as an argument to why accessibility issues have been resolved. The results depend on both the coverage and reliability of the evaluation methods chosen, and the evaluators’ ability to use these methods. The required expertise for reliably performing a conformance review as specified in W3C’s WCAG-EM document [53] is ambiguous. The document states that users of the methodology should have a “solid understanding” of WCAG 2.0 evaluation, accessible web design, assistive technologies among others. It further points to a list of resources the evaluators are expected to be “deeply familiar” with. While we, the authors of this thesis, consider ourselves to be deeply familiar with these resources, our limited previous experience in the field may impact the reliability of the evaluations. Theoretical knowledge of how people use assistive technologies, along with experience with using these technologies may be sufficient for being able to claim a “solid understanding” of assistive technologies, but it may not be able to replace the experience of watching people with disabilities use them in real scenarios.

The level of experience may affect the reliability of the evaluation in a sense that different evaluators will have a different pattern for manually checking for accessibility issues, and different abilities to spot these issues. WCAG-EM allows for a methodical and replicable evaluation of every WCAG success criterion, but the number of undetected issues may decrease with experience. While an automated testing tool will produce the same results every time it is run on a web page, manual inspection is still required since the automated tools cannot find all kinds of issues.

Regarding the interpretation of the success criteria and the specific implementations of solutions to these criteria, the applicability of the solutions will vary between applications. Since every application is different, different methods of solving the accessibility issues will work better in other contexts, and exact implementation details will only be applicable to a certain extent. Therefore, the replicability of the specific solutions depends on the application in which they are to be applied. However, the aim of this thesis has not been to determine which solutions might be better than others, or to provide universal solutions to accessibility issues, but rather propose example
solutions that are proven to work in a specific context. These results are therefore valid if the method for evaluating and testing for WCAG success criteria is reliable.

6.2.3 Source Criticism

Peer reviewed articles and widely acclaimed books have been used for the theoretical foundation, while details about WCAG have been acquired from W3C’s website. The list of references contains some sources from another era of the web, the content of which may have decreased in applicability since their respective source’s publication. Most notable are sources on accessibility that reference WCAG 1.0. An effort has been made to find updated sources, but the fundamentals of some of these older sources have been found to be very valuable, even if certain details may be outdated. In cases where older sources have been used, the time relevance of each statement has been evaluated before referencing.

Statistics have been acquired from major statistical bodies without personal interest whenever possible. While mainly primary sources have been used, secondary sources have also been used in some instances, mainly where such sources have provided a simpler interpretation of statistics. Examples of these are the summary of accessibility statistics from Funka [11] and the compilation of disability statistics published by the Swedish Post and Telecom Authority [13], which was commissioned to a third party. These sources are assessed as being less reliable than the primary sources from which those statistics were taken, such as Statistics Sweden, the Public Health Agency of Sweden and the Swedish National Agency for Education.

Further inaccuracy in the statistics cited in this thesis may be present since nation-wide statistics on who does and does not have a disability are in Sweden difficult to produce. This is especially true for school children, because of legislations regulating the use of sensitive personal information [12]. The act of diagnosing a child with a disability is also a sensitive topic since it can lead to exclusion and impact their self-esteem [12]. Consequently, many of the statistics used are on adults, which are not the target audience of the web applications of this study and may differ from corresponding numbers on children.

The input and recommendations from Hampus Sethfors have been valuable and his years of experience in the field of accessibility, having worked for Funka among others, give some credibility to his input. This input has also been in line with theory collected on the topic.

6.3 The Work in a Wider Context

For this study, no user testing was performed, and no personal information of any kind has been recorded, which avoids many issues of ethics commonly faced in similar studies. For future work on these web applications, however, evaluation with human subjects will be relevant, and as the target audience are children, special considerations have to be made.
In addition to the ethical aspects mentioned, the societal significance of accessibility research is reasonable to highlight. It is the intent of this thesis to, in addition to improve the accessibility of an e-learning application used by many, document and present the accessibility issues found and solutions implemented in this e-learning application. This will hopefully give developers concrete examples of such issues and solutions and aid them in their efforts to make their websites accessible. If this in turn can help a disabled person with their access to the web, the aim of the thesis has been achieved. The societal importance of an accessible web cannot be overstated, and it is inspiring to see initiatives in both public and private sectors toward universal web accessibility.
7 Conclusions

This thesis has resulted in example solutions for 15 out of 18 WCAG 2.0 level A and AA success criteria that were not met by the e-learning applications studied in this thesis. Even though significant accessibility improvements have been made, no level of conformance can be claimed for the web applications and, as such, they are seen as equal from a conformance level perspective, which highlights the limitations of this view. For a majority of the accessibility issues found, the fact that it was an e-learning application did not affect their respective solutions. However, a few adaptations were made due to the nature of the applications.

Some extended functionality was considered necessary to allow users to change different aspects of the visual presentation of content. While allowing third party solutions to change colors of text and background is a current AAA success criterion, and to change text spacing is a future WCAG 2.1 success criterion, such functionality was instead built into an accessibility toolbar in the applications. This decision was based on the notion that children in a learning environment should not have to seek third party solutions to be able to use the applications.

The high level of interactivity associated with successful e-learning applications produces issues perhaps less common in other types of applications. Students should be given the same experience in quizzes regardless of whether they can see the screen or not. To achieve some WCAG 2.0 criteria related to the interactive quizzes of the applications, it would have been sufficient to create a separate, perhaps not as usable, quiz intended for disabled people. But since every student deserves to have a similar experience, extensive screen reader adaptations of the quizzes were made instead so that they would be usable by everyone. Some types of quizzes, however, can never be answered by everyone (see section 5.3.1). Solutions exist to satisfy WCAG but, to solve the core issue, alternative types of assignments must be provided.

Issues related to video content, which is of utmost relevance to e-learning, cannot be solved by developers and have to be the responsibility of the content creators. They may however prove such a big challenge to content creators that their related success criteria are ignored. It is important that they are not, and that multimedia content is not avoided for the purpose of WCAG conformance since it is such an important tool for e-learning. The text content of an e-learning application is also highly relevant, but there are few accessibility guidelines that can regulate text content specifically. Ensuring the quality of such content therefore also has to be the responsibility of the e-learning content creators.

The research question proposed for this thesis was:

*How can accessibility issues be identified and solved in an existing e-learning application to improve accessibility in terms of WCAG 2.0 conformance?*

To successfully identify every different kind of accessibility issue, manual inspection methods such as walking through the application with a keyboard or screen reader have
to be employed in addition to using automated evaluation tools. The automated evaluation tools can quickly and reliably detect some but not all types of issues.

The solutions for the identified accessibility issues in the e-learning applications do not, in most cases, have to differ from solutions for any other web application. Such solutions are presented in section 5.3. In some cases, extended functionality connected to the visual presentation of content and quizzes, as detailed in section 5.4, can be desirable for children with, for example, cognitive disabilities. When creating accessible e-learning content, the possible incompatibilities between accessibility and e-learning theories discussed in section 6.1.3 should also be considered.

It is our wish that these answers can be of help to developers of e-learning applications and those who use them.
8 References


Appendix A – WCAG Level A and AA Success Criteria

Principle 1: Perceivable - Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below. (Level A)

- Controls, Input: If non-text content is a control or accepts user input, then it has a name that describes its purpose. (Refer to Guideline 4.1 for additional requirements for controls and content that accepts user input.)
- Time-Based Media: If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content. (Refer to Guideline 1.2 for additional requirements for media.)
- Test: If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content.
- Sensory: If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content.
- CAPTCHA: If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities.
- Decoration, Formatting, Invisible: If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology.

How to Meet 1.1.1 | Understanding 1.1.1

Guideline 1.2 Time-based Media: Provide alternatives for time-based media.

Understanding Guideline 1.2
1.2.1 Audio-only and Video-only (Prerecorded)

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: (Level A)

- Prerecorded Audio-only: An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content.
- Prerecorded Video-only: Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.

1.2.2 Captions (Prerecorded)

Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such. (Level A)

1.2.3 Audio Description or Media Alternative (Prerecorded)

An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such. (Level A)

1.2.4 Captions (Live)

Captions are provided for all live audio content in synchronized media. (Level AA)

1.2.5 Audio Description (Prerecorded)

Audio description is provided for all prerecorded video content in synchronized media. (Level AA)

Guideline 1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

1.3.1 Info and Relationships

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text. (Level A)

How to Meet 1.2.1 | Understanding 1.2.1
How to Meet 1.2.2 | Understanding 1.2.2
How to Meet 1.2.3 | Understanding 1.2.3
How to Meet 1.2.4 | Understanding 1.2.4
How to Meet 1.2.5 | Understanding 1.2.5
How to Meet 1.3.1 | Understanding 1.3.1
1.3.2 Meaningful Sequence

When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined. (Level A)

How to Meet 1.3.2 | Understanding 1.3.2

1.3.3 Sensory Characteristics

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound. (Level A)

Note: For requirements related to color, refer to Guideline 1.4.

How to Meet 1.3.3 | Understanding 1.3.3

Guideline 1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

Understanding Guideline 1.4

1.4.1 Use of Color

Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. (Level A)

Note: This success criterion addresses color perception specifically. Other forms of perception are covered in Guideline 1.3 including programmatic access to color and other visual presentation coding.

How to Meet 1.4.1 | Understanding 1.4.1

1.4.2 Audio Control

If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level. (Level A)

Note: Since any content that does not meet this success criterion can interfere with a user’s ability to use the whole page, all content on the Web page (whether or not it is used to meet other success criteria) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

How to Meet 1.4.2 | Understanding 1.4.2

1.4.3 Contrast (Minimum)

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following: (Level AA)

- Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1;
- Incidental: Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are
part of a picture that contains significant other visual content, have no contrast requirement.

- Logotypes: Text that is part of a logo or brand name has no minimum contrast requirement.

**1.4.4 Resize text**

Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality. (Level AA)

**1.4.5 Images of Text**

If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for the following: (Level AA)

- Customizable: The image of text can be visually customized to the user's requirements;
- Essential: A particular presentation of text is essential to the information being conveyed.

Note: Logotypes (text that is part of a logo or brand name) are considered essential.
Principle 2: Operable - User interface components and navigation must be operable.

Guideline 2.1 Keyboard Accessible: Make all functionality available from a keyboard.

Understanding Guideline 2.1

2.1.1 Keyboard

All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints. (Level A)

Note 1: This exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input but the underlying function (text input) does not.

Note 2: This does not forbid and should not discourage providing mouse input or other input methods in addition to keyboard operation.

How to Meet 2.1.1

2.1.2 No Keyboard Trap

If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away. (Level A)

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

How to Meet 2.1.2

Guideline 2.2 Enough Time: Provide users enough time to read and use content.

Understanding Guideline 2.2

2.2.1 Timing Adjustable

For each time limit that is set by the content, at least one of the following is true: (Level A)

- Turn off: The user is allowed to turn off the time limit before encountering it; or
- Adjust: The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or
- Extend: The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, "press the space bar"), and the user is allowed to extend the time limit at least ten times; or
- Real-time Exception: The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible; or
- Essential Exception: The time limit is essential and extending it would invalidate the activity; or
- 20 Hour Exception: The time limit is longer than 20 hours.

Note: This success criterion helps ensure that users can complete tasks without unexpected changes in content or context that are a result of a time limit. This success criterion should be considered in conjunction with Success Criterion 3.2.1, which puts limits on changes of content or context as a result of user action.

**2.2.2 Pause, Stop, Hide**

For moving, blinking, scrolling, or auto-updating information, all of the following are true: (Level A)

- Moving, blinking, scrolling: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and
- Auto-updating: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

Note 1: For requirements related to flickering or flashing content, refer to Guideline 2.3.

Note 2: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

Note 3: Content that is updated periodically by software or that is streamed to the user agent is not required to preserve or present information that is generated or received between the initiation of the pause and resuming presentation, as this may not be technically possible, and in many situations could be misleading to do so.

Note 4: An animation that occurs as part of a preload phase or similar situation can be considered essential if interaction cannot occur during that phase for all users and if not indicating progress could confuse users or cause them to think that content was frozen or broken.
Guideline 2.3 Seizures: Do not design content in a way that is known to cause seizures.

Understanding Guideline 2.3

2.3.1 Three Flashes or Below Threshold

Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds. (Level A)

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

How to Meet 2.3.1 | Understanding 2.3.1

Guideline 2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are.

Understanding Guideline 2.4

2.4.1 Bypass Blocks

A mechanism is available to bypass blocks of content that are repeated on multiple Web pages. (Level A)

How to Meet 2.4.1 | Understanding 2.4.1

2.4.2 Page Titled

Web pages have titles that describe topic or purpose. (Level A)

How to Meet 2.4.2 | Understanding 2.4.2

2.4.3 Focus Order

If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability. (Level A)

How to Meet 2.4.3 | Understanding 2.4.3

2.4.4 Link Purpose (In Context)

The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general. (Level A)

How to Meet 2.4.4 | Understanding 2.4.4

2.4.5 Multiple Ways

More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process. (Level AA)
2.4.6 Headings and Labels

Headings and labels describe topic or purpose. (Level AA)

2.4.7 Focus Visible

Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible. (Level AA)
Principle 3: Understandable - Information and the operation of user interface must be understandable.

Guideline 3.1 Readable: Make text content readable and understandable.

3.1.1 Language of Page
The default human language of each Web page can be programmatically determined. (Level A)

3.1.2 Language of Parts
The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text. (Level AA)

Guideline 3.2 Predictable: Make Web pages appear and operate in predictable ways.

3.2.1 On Focus
When any component receives focus, it does not initiate a change of context. (Level A)

3.2.2 On Input
Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component. (Level A)

3.2.3 Consistent Navigation
Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user. (Level AA)

3.2.4 Consistent Identification
Components that have the same functionality within a set of Web pages are identified consistently. (Level AA)
Guideline 3.3 Input Assistance: Help users avoid and correct mistakes.

**Understanding Guideline 3.3**

### 3.3.1 Error Identification

If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text. (Level A)

**How to Meet 3.3.1 | Understanding 3.3.1**

### 3.3.2 Labels or Instructions

Labels or instructions are provided when content requires user input. (Level A)

**How to Meet 3.3.2 | Understanding 3.3.2**

### 3.3.3 Error Suggestion

If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content. (Level AA)

**How to Meet 3.3.3 | Understanding 3.3.3**

### 3.3.4 Error Prevention (Legal, Financial, Data)

For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true: (Level AA)

- **Reversible:** Submissions are reversible.
- **Checked:** Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.
- **Confirmed:** A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

**How to Meet 3.3.4 | Understanding 3.3.4**
Principle 4: Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Guideline 4.1 Compatible: Maximize compatibility with current and future user agents, including assistive technologies.

Understanding Guideline 4.1

4.1.1 Parsing

In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features. (Level A)

Note: Start and end tags that are missing a critical character in their formation, such as a closing angle bracket or a mismatched attribute value quotation mark are not complete.

How to Meet 4.1.1 | Understanding 4.1.1

4.1.2 Name, Role, Value

For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies. (Level A)

Note: This success criterion is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criterion when used according to specification.

How to Meet 4.1.2 | Understanding 4.1.2
## Appendix B – Sample

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structured Sample</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Web application library *</td>
<td>A page with links to every web application in the library.</td>
</tr>
<tr>
<td>1</td>
<td>Common chapter navigation</td>
<td>A view with links to book chapters.</td>
</tr>
<tr>
<td>2</td>
<td>Text input quiz</td>
<td>A quiz answered by inputting the answer as free-form text.</td>
</tr>
<tr>
<td>3</td>
<td>Word selection</td>
<td>A page where the words to practice in a vocabulary quiz are selected.</td>
</tr>
<tr>
<td>-</td>
<td>Vocabulary quiz **</td>
<td>A quiz where the user is expected to input a word in a foreign language.</td>
</tr>
<tr>
<td>4</td>
<td>Quiz results</td>
<td>A page with an overview of the questions and answers of a completed quiz.</td>
</tr>
<tr>
<td>5</td>
<td>Words in different languages</td>
<td>A page where certain terminology is explained in different languages.</td>
</tr>
<tr>
<td>6</td>
<td>Multiple choice quiz</td>
<td>A quiz answered by selecting the correct option represented by a button.</td>
</tr>
<tr>
<td>7</td>
<td>What’s in the image quiz</td>
<td>A quiz where an image is presented and is answered by selecting the correct option represented by a button.</td>
</tr>
<tr>
<td>8</td>
<td>Reflection quiz</td>
<td>A quiz with a question and an expandable menu with the correct answer and two buttons with which the user chooses if their answer was correct or incorrect.</td>
</tr>
<tr>
<td>9</td>
<td>Accounting quiz</td>
<td>A quiz answered by inputting debit and credit for different accounts.</td>
</tr>
<tr>
<td>10</td>
<td>Accounting quiz results</td>
<td>A page with an overview of the questions and answers of a completed accounting quiz.</td>
</tr>
<tr>
<td>11</td>
<td>Bookkeeping quiz</td>
<td>A quiz represented as a table where the user practices recording financial transactions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>12</td>
<td>Chapter summary</td>
<td>A page with a summary of a book chapter.</td>
</tr>
<tr>
<td>13</td>
<td>Chapter narration</td>
<td>A page with audio files narrating a book chapter.</td>
</tr>
<tr>
<td>14</td>
<td>Educational video</td>
<td>A page with a video lesson.</td>
</tr>
<tr>
<td>15</td>
<td>Assignment navigation</td>
<td>A view with links to different assignments of a book chapter.</td>
</tr>
<tr>
<td>16</td>
<td>Text input multi-quiz ***</td>
<td>A free-form text quiz with multiple inputs on a single page.</td>
</tr>
<tr>
<td>17</td>
<td>Copyright menu</td>
<td>An expandable menu showing copyright information.</td>
</tr>
<tr>
<td>18</td>
<td>Share menu</td>
<td>An expandable menu where the page can be shared on social media.</td>
</tr>
</tbody>
</table>

**Random Sample**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Spanish word selection</td>
<td>An example of #4.</td>
</tr>
<tr>
<td>20</td>
<td>German vocabulary quiz</td>
<td>An example of #5.</td>
</tr>
</tbody>
</table>

* Omitted from sample since it is not to be considered as part of the web applications.

** Omitted from sample since it was shown to be identical to #3.

*** Not discovered initially but added to second sample.
Appendix C – Code Examples

Color Conversion: Hexadecimal to RGB

```javascript
/**
 * Converts a color in hexadecimal to r,g,b.
 * @param hex - The input color in hexadecimal.
 * @example
 * // returns "128,0,64"
 * hexToRgb("#800040");
 * @returns {string}
 */
function hexToRgb(hex) {
  hex = hex.substring(1);
  var bigint = parseInt(hex, 16);
  var r = (bigint >> 16) & 255;
  var g = (bigint >> 8) & 255;
  var b = bigint & 255;
  return [r, g, b].join();
}
```

Color Conversion: RGB to Hexadecimal

```javascript
/**
 * Converts a color in r,g,b format to hexadecimal.
 * @param rgb - The input color in r,g,b format.
 * @example
 * // returns ":#800040"
 * rgbToHex("128,0,64");
 * @returns {string}
 */
function rgbToHex(rgb) {
  var r = parseInt(rgb.substring(0, rgb.indexOf(",")), 10);
  var g = parseInt(rgb.substring(rgb.indexOf(",") + 1, rgb.lastIndexOf(",")), 10);
  var b = parseInt(rgb.substring(rgb.lastIndexOf(",") + 1, rgb.length), 10);
  return ":#" + ((1 << 24) + (r << 16) + (g << 8) + b) .toString(16).slice(1);
}
```
Lightening or Darkening a Color

```javascript
/**
 * Lightens or darkens a color.
 * @param color - The color in hexadecimal to alter.
 * @param value - A value between -1 and 1 where a negative value darkens the color, and a positive value lightens it.
 * @example
 * // returns "#808080"
 * shadeColor("#FFFFFF", -0.5);
 * @returns {string}
 */
function shadeColor(color, value) {
    var f=parseInt(color.slice(1),16),
        t=value<0?0:255,
        p=value<0?value*-1:value,
        R=f>>16,
        G=f>>8&0x00FF,
        B=f&0x0000FF;
    return "#"+
        0x10000000+(Math.round((t-R)*p)+R)*
        0x10000000+(Math.round((t-G)*p)+G)*
        0x10000000+(Math.round((t-B)*p)+B)).toString(16).slice(1);
}
```

Replacing an SVG Image

This function replaces all SVG icons without the class “exclude” with another with the same name, existing in another directory, “icon-black”.

```javascript
$(document).find('img[src$=".svg"]').not(".exclude")
    .attr('src', function(i, src) {
        return "../img/icon-black/" + src.replace(/^[^/\s]/, '');
    });
```
Calculating Relative Luminance

```javascript
/**
 * Calculates the relative luminance of a color.
 * Formulas from https://www.w3.org/TR/WCAG20-TECHS/G18.html
 * @param r - Red color value between 0 and 255.
 * @param g - Green color value between 0 and 255.
 * @param b - Blue color value between 0 and 255.
 * @example
 * // returns 0.0495935972188209
 * calcLuminance(128,0,64);
 * @returns {number}
 */
function calcLuminance(r, g, b) {
    var components = [ r / 255, g / 255, b / 255 ];
    for(var i = 0; i < 3; i++)
        components[i] = components[i] < 0.03928 ?
            components[i] / 12.92 :
            Math.pow((components[i] + 0.055) / 1.055,
                2.4);
    return 0.2126 * components[0] + 0.7152 * components[1] +
        0.0722 * components[2];
}
```

Calculating Contrast Ratio

```javascript
/**
 * Calculates the contrast ratio between two colors.
 * Formulas from https://www.w3.org/TR/WCAG20-TECHS/G18.html
 * @param r1 - Red color value of first color.
 * @param g1 - Green color value of first color.
 * @param b1 - Blue color value of first color.
 * @param r2 - Red color value of second color.
 * @param g2 - Green color value of second color.
 * @param b2 - Blue color value of second color.
 * @example
 * // returns 1.0575359142260676
 * calcContrastRatio(128, 0, 64, 32, 64, 128);
 * @returns {number}
 */
function calcContrastRatio(r1, g1, b1, r2, g2, b2) {
    var lum1 = calcLuminance(r1, g1, b1);
    var lum2 = calcLuminance(r2, g2, b2);
    return (lum1 >= lum2 ? (lum1 + 0.05) / (lum2 + 0.05) :
        (lum2 + 0.05) / (lum1 + 0.05));
}
## Appendix D – Success Criteria in Baseline Applications

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Level</th>
<th>Success Criterion</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>A</td>
<td>Non-text Content</td>
<td>Failed</td>
<td>Multiple icons and quiz-related images are missing alternative text.</td>
</tr>
<tr>
<td>1.2.1</td>
<td>A</td>
<td>Audio-only and Video-only (Prerecorded)</td>
<td>Cannot tell</td>
<td>It is unclear if audio tracks are sufficiently labeled as alternative to text in corresponding text book.</td>
</tr>
<tr>
<td>1.2.2</td>
<td>A</td>
<td>Captions (Prerecorded)</td>
<td>Cannot tell</td>
<td>It is unclear if audio tracks are sufficiently labeled as alternative to text in corresponding text book.</td>
</tr>
<tr>
<td>1.2.3</td>
<td>A</td>
<td>Audio Description or Media Alternative (Prerecorded)</td>
<td>Failed</td>
<td>Videos are missing an alternative audio track with narration of what the video is showing, and no media alternative is present.</td>
</tr>
<tr>
<td>1.2.4</td>
<td>AA</td>
<td>Captions (Live)</td>
<td>Not present</td>
<td>No live audio is present in the sample.</td>
</tr>
<tr>
<td>1.2.5</td>
<td>AA</td>
<td>Audio Description (Prerecorded)</td>
<td>Failed</td>
<td>Videos are missing an alternative audio track with narration of what the video is showing.</td>
</tr>
<tr>
<td>1.3.1</td>
<td>A</td>
<td>Info and Relationships</td>
<td>Failed</td>
<td>Forms throughout the sample are missing labels.</td>
</tr>
<tr>
<td>1.3.2</td>
<td>A</td>
<td>Meaningful Sequence</td>
<td>Not present</td>
<td>There are no instances in the sample where the sequence affects the meaning of the content.</td>
</tr>
<tr>
<td>1.3.3</td>
<td>A</td>
<td>Sensory Characteristics</td>
<td>Not present</td>
<td>Instructions for understanding and operating content are not present in the sample.</td>
</tr>
<tr>
<td>1.4.1</td>
<td>A</td>
<td>Use of Color</td>
<td>Failed</td>
<td>Correct or incorrect answers to quiz questions are often displayed only by color.</td>
</tr>
<tr>
<td>Test ID</td>
<td>Accessibility Level</td>
<td>Test Description</td>
<td>Result</td>
<td>Status Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------</td>
<td>---------------------------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.4.2</td>
<td>A</td>
<td>Audio Control</td>
<td>Not present</td>
<td>No auto-playing audio is present in the sample.</td>
</tr>
<tr>
<td>1.4.3</td>
<td>AA</td>
<td>Contrast (Minimum)</td>
<td>Failed</td>
<td>Minimum contrast ratio is not fulfilled for many web pages.</td>
</tr>
<tr>
<td>1.4.4</td>
<td>AA</td>
<td>Resize text</td>
<td>Failed</td>
<td>Content does not support text size changes in browser settings.</td>
</tr>
<tr>
<td>1.4.5</td>
<td>AA</td>
<td>Images of Text</td>
<td>Passed</td>
<td>Text is never represented as images in the sample. Logotypes are excluded from this criterion.</td>
</tr>
<tr>
<td>2.1.1</td>
<td>A</td>
<td>Keyboard</td>
<td>Failed</td>
<td>Several interactive elements are inaccessible by keyboard.</td>
</tr>
<tr>
<td>2.1.2</td>
<td>A</td>
<td>No Keyboard Trap</td>
<td>Passed</td>
<td>No keyboard traps have been identified in the sample.</td>
</tr>
<tr>
<td>2.2.1</td>
<td>A</td>
<td>Timing Adjustable</td>
<td>Not present</td>
<td>No time limits are present in the sample.</td>
</tr>
<tr>
<td>2.2.2</td>
<td>A</td>
<td>Pause, Stop, Hide</td>
<td>Not present</td>
<td>No moving, blinking, scrolling or auto-updating information is present in the sample.</td>
</tr>
<tr>
<td>2.3.1</td>
<td>A</td>
<td>Three Flashes or Below Threshold</td>
<td>Not present</td>
<td>No flashing content is present in the sample.</td>
</tr>
<tr>
<td>2.4.1</td>
<td>A</td>
<td>Bypass Blocks</td>
<td>Failed</td>
<td>No mechanism exists to bypass blocks of content.</td>
</tr>
<tr>
<td>2.4.2</td>
<td>A</td>
<td>Page Titled</td>
<td>Passed</td>
<td>Every web application has their own title with the name of the web application, which is sufficient to describe the purpose of the page.</td>
</tr>
<tr>
<td>2.4.3</td>
<td>A</td>
<td>Focus Order</td>
<td>Passed</td>
<td>The focus order is linear and there are no tree structures present.</td>
</tr>
<tr>
<td>2.4.4</td>
<td>A</td>
<td>Link Purpose (In Context)</td>
<td>Passed</td>
<td>Every link has an associated text describing the link destination.</td>
</tr>
<tr>
<td>Section</td>
<td>AA</td>
<td>Test</td>
<td>Result</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.4.5</td>
<td>AA</td>
<td>Multiple Ways</td>
<td>Passed</td>
<td>The web applications are easily navigable in a linear fashion, where subchapters and assignments are rendered on top of main content with backwards links preserved.</td>
</tr>
<tr>
<td>2.4.6</td>
<td>AA</td>
<td>Headings and Labels</td>
<td>Passed</td>
<td>All headings in sample are clear and descriptive.</td>
</tr>
<tr>
<td>2.4.7</td>
<td>AA</td>
<td>Focus Visible</td>
<td>Failed</td>
<td>Keyboard focus indicators are missing from multiple elements of pages in the sample.</td>
</tr>
<tr>
<td>3.1.1</td>
<td>A</td>
<td>Language of Page</td>
<td>Passed</td>
<td>The default human language is programmatically determined for each web page.</td>
</tr>
<tr>
<td>3.1.2</td>
<td>AA</td>
<td>Language of Parts</td>
<td>Failed</td>
<td>For pages with content in a foreign language, all text is programmatically determined to be in Swedish.</td>
</tr>
<tr>
<td>3.2.1</td>
<td>A</td>
<td>On Focus</td>
<td>Passed</td>
<td>No change of context is initiated when any component in the sample receives focus.</td>
</tr>
<tr>
<td>3.2.2</td>
<td>A</td>
<td>On Input</td>
<td>Failed</td>
<td>Context changes automatically and unexpectedly upon submission of answer to quiz question.</td>
</tr>
<tr>
<td>3.2.3</td>
<td>AA</td>
<td>Consistent Navigation</td>
<td>Failed</td>
<td>Navigation buttons are sometimes missing and are not always consistent between different views.</td>
</tr>
<tr>
<td>3.2.4</td>
<td>AA</td>
<td>Consistent Identification</td>
<td>Passed</td>
<td>Icons representing different types of assignments are used consistently. Button usage is consistent.</td>
</tr>
</tbody>
</table>
| 3.3.1   | A  | Error Identification | Not present | No form fields present in the sample except fields for answering quiz questions, where error
<table>
<thead>
<tr>
<th>CRP</th>
<th>A/R</th>
<th>Requirement</th>
<th>Level</th>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>A</td>
<td>Labels or Instructions</td>
<td>Passed</td>
<td>Failed</td>
<td>Purpose of input fields may not be obvious to screen readers.</td>
</tr>
<tr>
<td>3.3.3</td>
<td>AA</td>
<td>Error Suggestion</td>
<td>AA</td>
<td>Not present</td>
<td>No applicable form fields present in the sample. Providing correct answer before submission of quiz input fields would jeopardize the purpose of the content.</td>
</tr>
<tr>
<td>3.3.4</td>
<td>AA</td>
<td>Error Prevention (Legal, Financial, Data)</td>
<td>AA</td>
<td>Not present</td>
<td>No sensitive information can be submitted in or stored by the web applications.</td>
</tr>
<tr>
<td>4.1.1</td>
<td>A</td>
<td>Parsing</td>
<td>Passed</td>
<td>Failed</td>
<td>Duplicate IDs exist in many pages of the sample.</td>
</tr>
<tr>
<td>4.1.2</td>
<td>A</td>
<td>Name, Role, Value</td>
<td>Passed</td>
<td>Failed</td>
<td>UI controls are missing labels/titles.</td>
</tr>
</tbody>
</table>
## Appendix E – Success Criteria in Updated Applications

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Level</th>
<th>Success Criterion</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>A</td>
<td>Non-text Content</td>
<td>Passed</td>
<td>All non-text content in the sample has a text alternative</td>
</tr>
<tr>
<td>1.2.1</td>
<td>A</td>
<td>Audio-only and Video-only (Prerecorded)</td>
<td>Passed</td>
<td>Where chapter narration is present, there is text clearly stating that the audio tracks are alternatives to text in corresponding text book.</td>
</tr>
<tr>
<td>1.2.2</td>
<td>A</td>
<td>Captions (Prerecorded)</td>
<td>Passed</td>
<td>Where chapter narration is present, there is text clearly stating that the audio tracks are alternatives to text in corresponding text book.</td>
</tr>
<tr>
<td>1.2.3</td>
<td>A</td>
<td>Audio Description or Media Alternative (Prerecorded)</td>
<td>Failed</td>
<td>Videos are missing an alternative audio track with narration of what the video is showing, and no media alternative is present.</td>
</tr>
<tr>
<td>1.2.4</td>
<td>AA</td>
<td>Captions (Live)</td>
<td>Not present</td>
<td>No live audio is present in the sample.</td>
</tr>
<tr>
<td>1.2.5</td>
<td>AA</td>
<td>Audio Description (Prerecorded)</td>
<td>Failed</td>
<td>Videos are missing an alternative audio track with narration of what the video is showing.</td>
</tr>
<tr>
<td>1.3.1</td>
<td>A</td>
<td>Info and Relationships</td>
<td>Passed</td>
<td>All relevant information, structure and relationships can be programmatically determined.</td>
</tr>
<tr>
<td>1.3.2</td>
<td>A</td>
<td>Meaningful Sequence</td>
<td>Not present</td>
<td>There are no instances in the sample where the sequence affects the meaning of the content.</td>
</tr>
<tr>
<td>1.3.3</td>
<td>A</td>
<td>Sensory Characteristics</td>
<td>Not present</td>
<td>Instructions for understanding and operating content are not present in the sample.</td>
</tr>
<tr>
<td>Section</td>
<td>Level</td>
<td>Criterion</td>
<td>Status</td>
<td>Reason</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>------------------------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.4.1</td>
<td>A</td>
<td>Use of Color</td>
<td>Passed</td>
<td>Errors are never displayed only by color. An X or a checkmark is displayed along with the color change to illustrate a correct or incorrect answer.</td>
</tr>
<tr>
<td>1.4.2</td>
<td>A</td>
<td>Audio Control</td>
<td>Not present</td>
<td>No auto-playing audio is present in the sample.</td>
</tr>
<tr>
<td>1.4.3</td>
<td>AA</td>
<td>Contrast (Minimum)</td>
<td>Passed</td>
<td>No instances of inadequate contrast ratio were found in the sample.</td>
</tr>
<tr>
<td>1.4.4</td>
<td>AA</td>
<td>Resize text</td>
<td>Passed</td>
<td>The user can resize the text with the built-in accessibility controls up to 200% without loss of content or functionality.</td>
</tr>
<tr>
<td>1.4.5</td>
<td>AA</td>
<td>Images of Text</td>
<td>Passed</td>
<td>Text is never represented as images in the sample. Logotypes are excluded from this criterion.</td>
</tr>
<tr>
<td>2.1.1</td>
<td>A</td>
<td>Keyboard</td>
<td>Passed</td>
<td>All interactive elements are now accessible by keyboard.</td>
</tr>
<tr>
<td>2.1.2</td>
<td>A</td>
<td>No Keyboard Trap</td>
<td>Passed</td>
<td>No keyboard traps have been identified in the sample.</td>
</tr>
<tr>
<td>2.2.1</td>
<td>A</td>
<td>Timing Adjustable</td>
<td>Not present</td>
<td>No time limits are present in the sample.</td>
</tr>
<tr>
<td>2.2.2</td>
<td>A</td>
<td>Pause, Stop, Hide</td>
<td>Not present</td>
<td>No moving, blinking, scrolling or auto-updating information is present in the sample.</td>
</tr>
<tr>
<td>2.3.1</td>
<td>A</td>
<td>Three Flashes or Below Threshold</td>
<td>Not present</td>
<td>No flashing content is present in the sample.</td>
</tr>
<tr>
<td>2.4.1</td>
<td>A</td>
<td>Bypass Blocks</td>
<td>Passed</td>
<td>The markup is structured with &lt;nav&gt;, &lt;main&gt; and &lt;footer&gt; landmarks, and a &quot;skip navigation&quot; link exists in the beginning of the</td>
</tr>
</tbody>
</table>
accessibility toolbar, skipping to the first element in the `<main>` tag.

<table>
<thead>
<tr>
<th>Section</th>
<th>Level</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.2</td>
<td>A</td>
<td>Page Titled</td>
<td>Passed</td>
</tr>
<tr>
<td>2.4.3</td>
<td>A</td>
<td>Focus Order</td>
<td>Passed</td>
</tr>
<tr>
<td>2.4.4</td>
<td>A</td>
<td>Link Purpose (In Context)</td>
<td>Passed</td>
</tr>
<tr>
<td>2.4.5</td>
<td>AA</td>
<td>Multiple Ways</td>
<td>Passed</td>
</tr>
<tr>
<td>2.4.6</td>
<td>AA</td>
<td>Headings and Labels</td>
<td>Passed</td>
</tr>
<tr>
<td>2.4.7</td>
<td>AA</td>
<td>Focus Visible</td>
<td>Passed</td>
</tr>
<tr>
<td>3.1.1</td>
<td>A</td>
<td>Language of Page</td>
<td>Passed</td>
</tr>
<tr>
<td>3.1.2</td>
<td>AA</td>
<td>Language of Parts</td>
<td>Failed</td>
</tr>
<tr>
<td>3.2.1</td>
<td>A</td>
<td>On Focus</td>
<td>Passed</td>
</tr>
<tr>
<td>3.2.2</td>
<td>A</td>
<td>On Input</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Every web application has their own title with the name of the web application, which is sufficient to describe the purpose of the page.

The focus order is linear and there are no tree structures present.

Every link has an associated text describing the link destination.

The web applications are easily navigable in a linear fashion, where subchapters and assignments are rendered on top of main content with backwards links preserved.

All headings in sample are clear and descriptive.

All interactive elements in the sample now have a visible keyboard focus indicator.

The default human language is programmatically determined for each web page.

For pages with content in a foreign language, all text is programmatically determined to be in Swedish.

No change of context is initiated when any component in the sample receives focus.

When a screen reader user is clicks a button to be redirected to the next question in a quiz, they now
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Grade</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.3</td>
<td>AA</td>
<td>Consistent Navigation</td>
<td>Passed</td>
</tr>
<tr>
<td>3.2.4</td>
<td>AA</td>
<td>Consistent Identification</td>
<td>Passed</td>
</tr>
<tr>
<td>3.3.1</td>
<td>A</td>
<td>Error Identification</td>
<td>Not present</td>
</tr>
<tr>
<td>3.3.2</td>
<td>A</td>
<td>Labels or Instructions</td>
<td>Passed</td>
</tr>
<tr>
<td>3.3.3</td>
<td>AA</td>
<td>Error Suggestion</td>
<td>Not present</td>
</tr>
<tr>
<td>3.3.4</td>
<td>AA</td>
<td>Error Prevention (Legal, Financial, Data)</td>
<td>Not present</td>
</tr>
<tr>
<td>4.1.1</td>
<td>A</td>
<td>Parsing</td>
<td>Passed</td>
</tr>
<tr>
<td>4.1.2</td>
<td>A</td>
<td>Name, Role, Value</td>
<td>Passed</td>
</tr>
</tbody>
</table>