

# ACCESSIBILITY FOR SOFTWARE DEVELOPERS: A STUDY OF THE ONLINE TOOLS

MASTER THESIS

ABDUL SAMAD KAKAR

MASTER IN APPLIED COMPUTER SCIENCE

ØSTFOLD UNIVERSITY COLLEGE

# I. ABSTRACT

---

Since the approval of the UN Convention on the Rights of Persons with Disabilities, it has been a global initiative to work towards inclusivity in all aspects to make daily tools accessible for people with disabilities. In most countries, applying universal designs to a website is a legal obligation. Accessibility and inclusivity are not exclusive to people with disabilities, as they can affect other people with different barriers, which may lead to a complicated user experience. This study aims to test the accessibility of DevOps tools in order to promote accessibility for developers and to find answers for the research question "To what extent, the online DevOps tools are accessible according to WCAG guidelines?". The selected eight online DevOps tools are based primarily on two criteria: I) current online availability and II) open and free to use. DevOps tools are analyzed by Accessibility Conformance Evaluation Methodology (WCAG-EM) according to WCAG 2.1 conformance level AA. The tests show a 27% failure using DevOps tools; hence, the selected online DevOps tools need improvement to be fully accessible for developers with disabilities.

## II. TABLE OF CONTENTS

---

I.	ABSTRACT.....	I
II.	TABLE OF CONTENTS .....	II
III.	LIST OF TABLES .....	V
IV.	LIST OF FIGURES .....	VI
1	INTRODUCTION .....	1
1.1	RESEARCH QUESTION .....	4
1.2	STRUCTURE OF THE THESIS.....	4
2	FUNDAMENTALS AND RELATED WORKS .....	5
2.1	ACCESSIBILITY .....	5
2.1.1	Definition of Accessibility.....	5
2.1.2	Different Types of Disabilities and Barriers .....	5
2.1.3	Components of Web Accessibility .....	7
2.1.4	Assistive Technologies Used for Web Browsing.....	8
2.2	WEB ACCESSIBILITY STANDARDS .....	9
2.2.1	World Wide Web Consortium .....	9
2.2.2	WCAG 2.0/2.1.....	9
2.2.3	Website Accessibility Evaluation .....	11
2.2.4	Implementing Recommendations from Web Accessibility Guidelines .....	12
2.3	DEVELOPMENT AND OPERATIONS (DEVOPS) .....	12
2.4	RELATED WORKS .....	16
2.4.1	The Agency for Public Management and eGovernment (DIFI) and Web accessibility conformance testing	17
2.4.2	DIFI surveys .....	17
3	DESIGN AND IMPLEMENTATION .....	19
3.1	WEBSITE ACCESSIBILITY EVALUATION METHODOLOGY (WCAG-EM):.....	20
3.2	WEB ACCESSIBILITY GUIDELINES 2.1.....	24
3.2.1	Principle 1. Perceivable.....	24
3.2.2	Principle 2. Operable .....	30
3.2.3	Principle 3. Understand .....	34
3.2.4	Principle 4. Robust.....	37
3.3	TEST ENVIRONMENT.....	38

3.4	EVALUATION TOOLS:	38
3.4.1	<i>Firefox Developer Tools</i>	39
3.4.2	<i>HTML validator</i>	39
3.4.3	<i>Wappalyzer</i>	39
3.4.4	<i>WebAIM</i>	39
3.4.5	<i>WAVE</i>	39
3.4.6	<i>AChecker</i>	39
3.4.7	<i>Paciello Group Contrast Analyzer</i>	40
3.4.8	<i>A11Y Color Contrast Accessibility Validator</i>	40
3.4.9	<i>AXE: Accessibility for Development Teams</i>	40
3.4.10	<i>Accessibility Bookmarklets</i>	40
3.4.11	<i>JAWS</i>	40
3.4.12	<i>NVDA</i>	40
3.5	SELECTION OF THE SAMPLE PAGES	41
3.6	SCORING PROCEDURE	41
<b>4</b>	<b>EVALUATION</b>	<b>41</b>
4.1	GITHUB	42
4.1.1	<i>Evaluation Scope</i>	42
4.1.2	<i>Web technologies relied upon</i>	42
4.1.3	<i>Representative Sample</i>	42
4.1.4	<i>Testing the samples</i>	42
4.1.5	<i>Evaluation Findings</i>	46
4.2	GITLAB	48
4.2.1	<i>Evaluation Scope</i>	48
4.2.2	<i>Web technologies relied upon</i>	48
4.2.3	<i>Representative Sample</i>	48
4.2.4	<i>Testing the samples</i>	48
4.2.5	<i>Evaluation Findings</i>	52
4.3	BITBUCKET	54
4.3.1	<i>Evaluation Scope</i>	54
4.3.2	<i>Web technologies relied upon</i>	54
4.3.3	<i>Representative Sample</i>	55
4.3.4	<i>Testing the samples</i>	55
4.3.5	<i>Evaluation Findings</i>	59
4.4	SONARQUBE	61
4.4.1	<i>Evaluation Scope</i>	61
4.4.2	<i>Web technologies relied upon</i>	61

4.4.3	<i>Representative Sample</i> .....	61
4.4.4	<i>Testing the samples</i> .....	61
4.4.5	<i>Evaluation Findings</i> .....	65
4.5	TRAVIS CI .....	67
4.5.1	<i>Evaluation Scope</i> .....	67
4.5.2	<i>Web technologies relied upon</i> .....	67
4.5.3	<i>Representative Sample</i> .....	67
4.5.4	<i>Testing the samples</i> .....	67
4.5.5	<i>Evaluation Findings</i> .....	70
4.6	SLACK .....	72
4.6.1	<i>Evaluation Scope</i> .....	72
4.6.2	<i>Web technologies relied upon</i> .....	73
4.6.3	<i>Representative Sample</i> .....	73
4.6.4	<i>Testing the samples</i> .....	73
4.6.5	<i>Evaluation Findings</i> .....	77
4.7	TRELLO .....	79
4.7.1	<i>Evaluation Scope</i> .....	79
4.7.2	<i>Web technologies relied upon</i> .....	79
4.7.3	<i>Representative Sample</i> .....	79
4.7.4	<i>Testing the samples</i> .....	79
4.7.5	<i>Evaluation Findings</i> .....	83
4.8	HEROKU .....	85
4.8.1	<i>Evaluation Scope</i> .....	85
4.8.2	<i>Web technologies relied upon</i> .....	86
4.8.3	<i>Representative Sample</i> .....	86
4.8.4	<i>Testing the samples</i> .....	86
4.8.5	<i>Evaluation Findings</i> .....	90
5	<b>DISCUSSION</b> .....	<b>92</b>
5.1	LIMITATIONS AND FUTURE WORK .....	116
6	<b>CONCLUSION</b> .....	<b>116</b>
7	<b>APPENDIX A</b> .....	<b>117</b>
7.1	WCAG 2.1 .....	117
8	<b>APPENDIX B</b> .....	<b>123</b>
8.1	THE 2014 DIFI SURVEY ON THE UNIVERSAL DESIGN OF WEBSITES IN THE PRIVATE AND PUBLIC SECTOR .....	123
9	<b>REFERENCES</b> .....	<b>126</b>

### III. LIST OF TABLES

---

TABLE 1: LIST OF ALL DEVOPS TOOLS .....	16
TABLE 2: GITHUB RESULTS.....	45
TABLE 3: GITLAB RESULTS.....	52
TABLE 4: BITBUCKET RESULTS .....	58
TABLE 5: SONARQUBE RESULTS .....	65
TABLE 6: TRAVIS CI RESULTS .....	70
TABLE 7: SLACK RESULTS .....	76
TABLE 8: TRELLO RESULTS.....	83
TABLE 9: HEROKU RESULTS .....	89
TABLE 10: OVERALL DEVOPS TOOLS RESULTS .....	93
TABLE 11: ALL CRITERIA FOR DEVOPS TOOLS.....	99
TABLE 12: RESULT OF ALL CRITERIA BASED ON THE LOWEST SCORE.....	105
TABLE 13: RESULT OF ALL CRITERIA BASED ON TOTAL SCORE .....	109
TABLE 14: RESULT OF ALL CRITERIA BASED ON TOTAL SCORE COUNTING NOT-PRESENT AS PASS.....	114
TABLE 15: COMPARISON OF OVERALL, AGGREGATE LOWEST SCORE AND AGGREGATE LOWEST SCORE COUNTING NOT-PRESENT AS PASS RESULTS.....	114
TABLE 16: SUMMARY OF WCAG 2.0 AND WCAG 2.1 .....	122

## IV. LIST OF FIGURES

---

FIGURE 1: DEVOPS TOOLS CATEGORIES .....	14
FIGURE 2: EVALUATION PROCEDURE .....	20
FIGURE 3: GITHUB LOWEST SCORE .....	46
FIGURE 4: GITHUB RESULT COUNTING NO-PRESENT AS PASS .....	47
FIGURE 5: GITHUB OVERALL RESULT .....	47
FIGURE 6: GITLAB LOWEST SCORE .....	53
FIGURE 7: GITLAB RESULT COUNTING NO-PRESENT AS PASS .....	53
FIGURE 8: GITLAB OVERALL RESULT .....	54
FIGURE 9: BITBUCKET LOWEST SCORE .....	59
FIGURE 10: BITBUCKET RESULT COUNTING NO-PRESENT AS PASS .....	60
FIGURE 11: BITBUCKET OVERALL RESULT .....	60
FIGURE 12: SONARQUBE LOWEST SCORE .....	65
FIGURE 13: SONARQUBE RESULT COUNTING NO-PRESENT AS PASS .....	66
FIGURE 14: SONARQUBE OVERALL RESULT .....	66
FIGURE 15: TRAVIS CI LOWEST SCORE .....	71
FIGURE 16: TRAVIS CI RESULT COUNTING NO-PRESENT AS PASS .....	71
FIGURE 17: TRAVIS CI OVERALL RESULT .....	72
FIGURE 18: SLACK LOWEST SCORE .....	77
FIGURE 19: SLACK RESULT COUNTING NO-PRESENT AS PASS .....	78
FIGURE 20: SLACK OVERALL RESULT .....	78
FIGURE 21: TRELLO LOWEST SCORE .....	84
FIGURE 22: TRELLO RESULT COUNTING NO-PRESENT AS PASS .....	84
FIGURE 23: TRELLO OVERALL RESULT .....	85
FIGURE 24: HEROKU LOWEST SCORE .....	90
FIGURE 25: HEROKU RESULT COUNTING NO-PRESENT AS PASS .....	91
FIGURE 26: HEROKU OVERALL RESULT .....	91
FIGURE 27: OVERALL DEVOPS TOOLS LOWEST SCORE PERCENTAGE .....	94
FIGURE 28: OVERALL DEVOPS TOOLS LOWEST SCORE .....	94
FIGURE 29: OVERALL DEVOPS TOOLS COUNTING NO-PRESENT AS PASS .....	95

FIGURE 30: DEVOPS TOOLS AGGREGATE LOWEST SCORE.....	100
FIGURE 31: DEVOPS TOOLS RESULT COUNTING NO-PRESENT AS PASS FOR AGGREGATE LOWEST SCORE.....	100
FIGURE 32: DEVOPS TOOLS AGGREGATE LOWEST SCORE RESULTS.....	101
FIGURE 33: COMPARISON OF OVERALL, AGGREGATE LOWEST SCORE AND AGGREGATE LOWEST SCORE COUNTING NOT-PRESENT AS PASS RESULTS.....	115



# 1 INTRODUCTION

---

Technology has been continuously developing to encourage cultural progress and facilitate the digital lifestyle. Web accessibility promotes accessing web content to all people regardless of disabilities, and it is mandatory regulation in many European countries [1]. Thus it should be provided equally to all users to perceive the valuable information from the web contents through their senses [2].

The World Wide Web Consortium (W3C) has produced the widely applied standards for web improving accessibility known as the Web Content Accessibility Guidelines (WCAG) [2]. The standards help to make the websites useful for all users, not only users with disabilities. The W3C is in cooperation with organizations, communities, and individuals worldwide. The W3C upholds the principles proposed by Tim Berners-Lee, creator of the web and founder and director of W3C. Here, accessibility is at the core.

*"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect".* Tim Berners-Lee

W3C is also in line with the UN convention principles, including digital accessibility for all people with or without disabilities [3]. Therefore, digital accessibility is a worldwide initiative. As highlighted by Bruce Betsy, author of the book Dreamweaver CS4 in 24 Hours:

*"Believe it or not, some people who may surf to your Web pages; that are visually impaired and have speech synthesizers that read the contents of Web pages. For all of these viewers, you should add alternative text to your images" [4].*

Despite the popularity of WCAG guidelines, they have received criticism regarding unsatisfactory user experience [5] and difficulty to evaluate [6]. In 2010, according to an extensive study of web accessibility carried out by authors at the University of Lisbon [7], only 4% of all 28 million pages were fully accessible. In 2018, the European Commission adopted two implementing decisions to enhance this situation, ensuring uniform conditions

for implementing the Web Accessibility Directive [8]. The Commission also provides financial support to research on web accessibility, accessible technology, and solutions deployment.

People with disabilities are estimated to make up 15% of the world's population [9].

However, there is a wide range of disabilities such as blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited motor movements, speech disabilities, photosensitivity, and any combinations of the impairments. According to WHO, people with disabilities recently were acknowledged as the largest minority group in the world [10]. However, web accessibility is essential for improving the accessibility perception for people with disabilities and helps people without disabilities in different user experiences and circumstances. For example, people using different devices (mobile phone, tablet, smart TV, and other devices with varying dimensions of screen), people with temporary disabilities, people with situational limitations (low or bright light), slow internet connection, or elderly [11]. Therefore, the rationale for organizations to address accessibility goes beyond laws requiring digital accessibility [12].

IT industry is not exempt from software practitioners with disabilities, and it could benefit from accessibility features. Stack Overflow surveyed 64,000 practitioners in 2017; the results revealed that 3.6% of the respondents recognize themselves as having a particular physical disability [13]. Similar results were obtained in 2018 [14] and 2019 [15], which involved around 100,000 and 90,000 respondents. It means approximately 2.8% to 2.9% of the respondents recognized themselves as having a particular disability. These results highlight that there are software practitioners with disabilities, and thus, providing accessibility is an essential feature for them. However, as mentioned before, people with disabilities are around 15% then the number of software practitioners with disabilities could be bigger than the 3% reported in these surveys.

Indeed, accessibility has gained increasing attention from large corporations in the software industry. For instance, IBM created the IBM Team Able consisting of employees with disabilities that spend their extra time to create accessible technology [16]. Their effort has promoted better understanding and better results when creating inclusive products. Thus, some IBM products get feedback from Team Able volunteers. Google is also committed to making its tools accessible for everyone. They do so by hiring people with disabilities, such

as Victor Tsaran, a Technical Program Manager [17], and offering Google Accessibility support to get feedback from users with disabilities [18]. In this way, it anticipates that individuals have opportunities to benefit from the internet, and it consequently helps to improve new technologies by employing equally accessibility perception. According to the article diversity in software engineering [19], people with disabilities can improve technology. However, improvements in technology and the internet also depend on overcoming the potential accessibility issues of online tools [20].

Global survey regarding Pandemic Programming and how COVID-19 affects software developers shows that 8% of the developers have a disability that affects their work and productivity [21]. The survey proposes that to improve employee productivity, companies must make an effort for their well-being and home office ergonomics situation, particularly for disabled women and parents [21].

In general, diversity in the IT workforce promotes a better dynamic amongst employees, improved teamwork, and efficiency [19]. However, the term diversity refers to gender and race and (permanent or temporal) disabilities. According to the Stack Overflow survey in the software industry, key roles are full-stack developer, back-end developer, front-end developer, desktop application developer, mobile developer, quality assurance developer, database administrator, designer, DevOps specialist, system administrator, and others [15]. In this context, one of the main obstacles that software practitioners with disabilities face in the workplace is the tools used for software development. Therefore, to succeed in the workplace, they need ramps to make physical spaces [22], assistive technologies such as screen reading software, and accessible tools, including online tools used to develop software.

On the other hand, software companies succeed based on their ability to innovate faster than their competitors; hence there is an increasing interest in DevOps from academics and practitioners [23]. DevOps is a software approach that promotes the integration and collaboration of development and operations while providing continuous delivery in response to the increasing velocity demands [24]. In this approach, developer tools are vital since they improve product quality, speed, lower the cost, and speed up the product release [25]. Therefore, accessibility plays a vital role in this context as well.

Despite accessibility being a universal topic and the research efforts made in this field, to the best of the author's knowledge, the accessibility of the tools used by software developers is little known. Consequently, this exploratory study will focus on the accessibility of developer tools; in particular, DevOps tools have their growing importance in the software industry. Therefore, some online DevOps tools will test their level of accessibility according to the accessibility guidelines 2.1.

## 1.1 RESEARCH QUESTION

This study aims to test the accessibility of DevOps tools to promote accessibility for developers. Therefore, based on that objective, this study formulates the following research question:

**"To what extent, the online DevOps tools are accessible according to according to WCAG guidelines?"**

The selected DevOps tools are GitHub, GitLab Bitbucket, SonarQube, Travis Ci, Slack, Trello, and Heroku. The selection was based primarily on two criteria: I) current online availability and II) open and free to use. Website Accessibility Conformance Evaluation Methodology (WCAG-EM) [26] is used to analyze the level of conformance of these DevOps tools according to WCAG 2.1. Moreover, the following tools were used: I) Firefox for code inspection, II) WAVE AXE, NVDA from the Elsevier accessibility checklist [27], and III) AChecker, HTML validator from WAI-Web Accessibility Evaluation Tools [28].

## 1.2 STRUCTURE OF THE THESIS

The thesis is divided into five main chapters:

- 1 "Fundamental and Related works" present fundamentals of accessibility, WCAG, DevOps, and related works.
- 2 "Design and Implementation" describes the research approach based on the Website Accessibility Conformance Evaluation Methodology (WCAG-EM).
- 3 "Evaluation" presents the testing process used to evaluate DevOps tools according to WCAG 2.1.

- 4 "Discussion" gives insight into the significance of the result obtained and describing the current work limitations and ideas for future work.
- 5 "Conclusion" summarizes the research and concluding the collected data.

## 2 FUNDAMENTALS AND RELATED WORKS

---

The topics regarding web accessibility are not a new concept in the continuously developing world of technology. In many countries worldwide, especially in European countries, regulations are mandated by law to apply the WCAG 2.0 to both the public and private sectors [1].

Given that there is not enough attention to the topic of accessibility of online tools for developers. The focus of the relevant literature for the research is based on aspects regarding accessibility, DevOps tools, and related works.

### 2.1 ACCESSIBILITY

#### 2.1.1 Definition of Accessibility

**ISO 25000** is a series of standards known as Software Product Quality Requirements and Evaluation, or shortly known as *SQuaRE*. According to the ISO 25000 series, accessibility is defined as a degree that measures how much data is accessible in a particular context of use towards the people who need technological solutions due to disabilities. These series determine the software product quality, **ISO 25010**, quality of data product, **ISO 25012**, and quality evolution of software products, **ISO 25040** [29].

#### 2.1.2 Different Types of Disabilities and Barriers

Disability can be defined as the physical or mental condition limiting a person's movement, sensation, or activities. There are diverse types of disabilities, in which the degree a person experience varies. These disabilities are grouped into six categories: auditory, cognitive, learning and neurological, physical, speech, and visual [30]. Each disability has its challenges and barriers experienced by users when "accessible" options are not available on the web.

People with auditory disabilities experience obstacles on the web, primarily through media content and the lack of captions and transcripts [30]. In some circumstances, the lack of volume control and caption adjustability options can be challenging for users.

Auditory disability includes primarily the volume control problem that leads toward a problematic experience in obtaining information from the web since information may be complex for them to comprehend [30]. In practice, auditory disabled people should have easy access to correct captions and transcripts on media-based content and the option to adjust the captions through text size and color. The volume control problem is categorized as one that can be catered through technological tools.

Cognitive, learning, and neurological disabilities, are a wide range of neurological disorders, including behavioral and mental health disorders [30]. The web may provide various solutions for web accessibility for people under mentioned category. However, they still experience barriers that may hinder them from using the web effectively [30]. The barriers vary as this category is quite diverse and broad. For instance, the challenges that arise when using the web may be similar to other problems stated above. Complexity in navigating through a page and complex page layouts and text content may provide an unpleasant experience for these users [30]. Websites with continually moving or flickering content and audio that cannot shut down could also be barriers. A clear and concise page layout with consistent visual design and simple text can give the user a better experience.

Physical disabilities need a combination of specialized hardware and software solutions to use the web effectively [30]. The lack of providing full keyboard support and insufficient time to finish tasks are some barriers that arise. A page layout that is inconsistent and difficult to navigate through may also provide difficulty for users. A better page layout and support for specialized hardware and software devices may provide a better user experience [30].

Speech disabilities neglected by web-based services that depend solely on voice communication, such as only through phone numbers and written communication such as through e-mail and chat, might be deemed to be a challenge for users with speech disabilities [30]. Providing written communication services will help users.

Users with visual disabilities may experience challenges if there is a lack of text alternatives. Including the text, images, and page layouts that cannot be resized or lose information

when resized. Media contents that do not provide audio alternatives, including the subtitles or description of audio considered as a barrier by users [30]. The page layout may also be a source of difficulty if it does not provide a consistent, accessible design and interface.

Accessible design of the page layout that offers various ways of presentation and interaction; may help with the user's experience.

Aside from the disabilities mentioned above, other groups of people without disabilities may benefit from web accessibility. For instance, users who use devices with a smaller or broader screen, age-related challenges or users that experience change in their abilities to independently navigate through the web, various health conditions or temporary disabilities, users with multiple disabilities, or even users with limitations caused by a specific situation such as slow internet speed [30].

In fact, according to W3C, accessible web design for every kind of user will give an increased and better overall experience for the user. Consideration should also be shown for older people since older people have age-related impairments such as vision, hearing, cognitive, or physical disabilities [35]. Since 2010 accessibility, usability, and inclusion are recognized as related aspects to create websites that work for everyone; bonding these three leads to a more accessible, usable, and inclusive web experience [12]. As mentioned before, accessibility means that people with disabilities can easily access the web without limitations. Usability is more about the product and its effectiveness, efficiency, and satisfactory results. Inclusion is the design for all or universal design, including people with disabilities, elderly/youth, people with low literacy or not fluent in the language, different device users, and more.

### 2.1.3 Components of Web Accessibility

To be able to provide adequate accessibility on the web, specific essential components are taken into consideration. In what follows a brief description of these components [32]:

- Content pertains to the information provided on the page or application. The content may include the physical text and underlying code that users use to provide the overall structure and presentation of the layout.
- User agents are web browsers or media players that the user uses.

- Assistive technologies are specialized tools that provide an assisting hand for the user with a disability. These can be alternative keyboards or switches, software for scanning or speech assistance, or screen readers.
- Developers are also essential as they must provide and contribute to the digital content provided.
- Authoring tools are software that creates or publishes a website.
- Evaluation tools are needed in terms of web accessibility to check if they meet the user's needs. HTML and CSS validators are also included in this.

#### 2.1.4 Assistive Technologies Used for Web Browsing

Disabled People has his way of adapting the solutions to develop a way to interact with the web. However, one thing is sure that the lack of an assisting tool can create new challenges for disabled people. Assistive technologies and adaptive strategies are commonly used for users with disabilities to interact with the web. Perception, presentation, input, and interaction are vital when using assistive technologies and adaptive methods [33].

Perception is the way a user hears, feels, or sees. To perceive and comprehend the content, some users require converting it into another form. The Braille system and sign language are examples of ways the user can convert content so they will be able to perceive it. Strategies that can be incorporated are screen readers, voice browsers, or Braille display [33]. These strategies will help in assisting the user according to their needs. Developers should consider presenting the content on the web page to make it easier for users to distinguish and comprehend the overall page. Some features are customized content through the fonts, colors, and outline, more straightforward text content that would be easier to comprehend for the user, screen magnification without compromising the layout or content, sign language, and symbols.

Strategies that can be adapted are automated blockers for pop-up windows or animations, assistive software that provides help reading or scanning through the page, screen magnifier, and easy access to volume control [33]. There are numerous assistive technologies and adaptive strategies, such as keyboard and mouse customization and filters or compatible alternative keyboard and mouse. Tools that check the spelling and grammar



and predictive texts may help users when writing. Interaction involves navigation and finding content on the page [33]. In this part, the overall page layout is vital to provide proper assistance and the chance for adaptation [33].

## 2.2 WEB ACCESSIBILITY STANDARDS

### 2.2.1 World Wide Web Consortium

World Wide Web Consortium, or W3C for short, is led by Tim Berners-Lee (director), the inventor of the world wide web, and Jeffrey Jaffe (CEO) with other international organizations and staff to create and develop web standards [34]. The mission of W3C is to ensure that the web grows to its full potential. Hence in 2012, W3C developed principles that include two major principles that are Web for all and Web on Everything. The *design principle of web for all and Web on Everything*. "Web for all" means everyone, including people with disabilities, can access web pages without any difficulties. At the same time, *Web on Everything* means to have web pages accessible through any device. To sum up, the vision for the web is to improve trust on a global scale and the participation of people worldwide for sharing information and knowledge [35].

### 2.2.2 WCAG 2.0/2.1

The first WCAG (WCAG 1.0) was developed and published in 1999. Then, this version was updated to WCAG 2.0 in 2008. The WCAG 2.0 acts as a vital source for international standards and guides for web designers to create and design inclusive and accessible tools. It is intended for user experience, web designers, web content creators, and developers. Based on four principles for web accessibility: Perceivable, Operable, Understand, and Robust [36]. In 2018, an updated version of WCAG (WCAG2.1) was published.

Web Accessibility Initiative launched in 1997, intending to improve accessibility on the web. The WAI has introduced three core guidelines or Web accessibility standards[37]:

- Web Content Accessibility Guidelines or WCAG, which focuses and tackles the information provided by a website. This information pertains to the text and graphics (images, sounds) on a website.

- Authoring Tool Accessibility Guidelines, or *ATAG* in short, is a list of software and services used by developers to create content on the web.
- User Agent Accessibility Guidelines, or *UAAG* in short, is a set of guidelines allocated for web browsers, media players, and user agents.

The WCAG was primarily created for web designers and developers, including policymakers, teachers, students, and purchasing agents. The principles and techniques presented in the WCAG may be used and applied directly to website development. Four layers of guidance include principles, guidelines, success criteria, sufficient and advisory techniques are provided to benefit the users, especially persons with disabilities

The top layer of guidance is the principles for web accessibility. It comprises of 4 principles that are: *perceivable*, *operable*, *understandable*, and *robust*. The **POUR** principles are easy to apply to the design and help the designer or developer create an accessible website.

The next layer of guidance is the success criteria. Each of the thirteen guidelines consists of testable success criteria that are to be used to verify a guideline's accessibility and conformance to a website. Testing conformance is required in cases such as in legal regulations, agreements, and design specifications. There are three levels of conformance for each criterion: A (low level), AA (middle level), AAA (high level). Meeting conformance level A, the web page meets all the requirements of the level A success criteria. For levels AA and AAA, the web page must meet all the success criteria for previous levels. There are five WCAG conformance requirements: I) conformance level, II) full pages, III) complete processes, IV) only accessibility-supported Ways of Using Technologies, and V) non-Interference [36, p. 2].

Lastly, for every guideline and success criteria, various informative techniques are provided in the WCAG. There are two categories of the different techniques, namely, sufficient and advisory. Sufficient techniques meet the success criteria, while advisory techniques may exceed what is needed by each success criteria and allow the designer/developer to improve further, acknowledging the guidelines. Some instructional techniques included insufficient techniques to acknowledge barriers for accessibility that are not covered by the testable success criteria.

For web accessibility—making the World Wide Web similarly accessible for all users, without counting the physical and cognitive levels of the users—WCAG 2.0 by the World Wide Web Consortium (W3C) is the best quality level. Its AA level of consistency is the principal reference point for accessibility standards worldwide, including the United Nations, European Union, the United States, and others. Be that as it may, WCAG 2.0 was published in 2008, and the web has changed significantly in 10 years. WCAG 2.1 started in 2017, with the most recent update (as of this composition) distributed in June 2018. WCAG 2.1 delivers changes to the web and how innovations can be executed, so all users have equivalent access. Seventeen new success criteria are added in WCAG 2.1, as shown in Table 16 and discussed in the design and implementation section.

### 2.2.3 Website Accessibility Evaluation

The research shows that over-reliance on automated testing is not recommendable, leading to harmful and misleading results [38]. Web accessibility evaluation tools can assist only so that human judgment is required [39]. In addition to the mentioned standard methods, a different approach is called the Barrier walkthrough method [40]. The barrier walkthrough method proposes to introduce barriers, defect, users, affected, and consequences. Standard accessibility evolution methods are those methods that are used to access the performance. Standard accessibility evolution methods are cost-effective, more comfortable to obtain results, and more reliable than the Barrier walkthrough method. If the sound barriers are introduced, this method can generate more valuable results [41]. There are various methods to determine the level of accessibility of the website. Website Accessibility Evaluation Methodology (WCAG-EM) is a method to determine whether a website conforms to WCAG. It is essential because the derived form of this concept is related to the research. WCAG-EM provides the following five steps for conformance evaluation [41]:

1. Define the scope of the evaluation: Defining what is included in the evaluation, the goal of the evaluation, and the WCAG conformance level (A, AA, AAA).
2. Explore the website: Identify the key web pages, key functionality, types of web content, designs, functionality, required web technologies, and more.

3. Select a representative sample: Guidance on structured and randomly selected web pages or evaluate every web page on a website if feasible.
4. Evaluate the selected sample: Determining successes and failures in meeting WCAG, accessibility support for website features, and recording evaluation steps.
5. Report the evaluation findings: aggregate and report evaluation findings, make evaluation statements and calculate overall scores.

WCAG provided the template for Accessibility Evaluation Reports and WCAG-EM Report Tool, which is helpful in the planning and testing phase.

## 2.2.4 Implementing Recommendations from Web Accessibility

### Guidelines

Implementing the WCAG 2.0 on websites benefits both user groups: disabled and non-disabled users. Therefore, the higher conformance to WCAG 2.0, the better the result in benefits for both groups. Many standard measurements could be used to determine the efficiency of the website and its usage. The measures were task completion rate and time, usability, positive and negative effect user experience. The tests were conducted on 55 non-disabled and visually impaired participants to verify the results [42]. Another study suggests that many will assume that applying WCAG 2.0 does not help non-disabled users or negatively impact [2]. The negative consequences mean that instead of contributing positively, it might increase the issues as it becomes even more challenging to use. The same test was tested on 61 non-disabled users, and the results show that the higher the conformance of the website, the better the participants' performance. It improved the users' ratings or improved users' feedback on aspects regarding the principles, such as perceived usability, aesthetics, workload, and trustworthiness [42].

## 2.3 DEVELOPMENT AND OPERATIONS (DEVOPS)

Research has been conducted to analyze the influence of Web Accessibility deeply, and it was concluded that there are three primary motivators for it. User advocacy, Web

Accessibility demand from users, self-perceptions as a specialist are essential solutions for this issue.

Web Accessibility is a part of the web developer's job for improving the quality of the product [2].

In an article on IBM's blog, about "DevOps: Three Ways to Improve DevOps to Ensure Accessibility," the author introduced three core strategies to ensure any application is personalized according to the users' needs [43]. These three core strategies provide accessibility for an integral design function, conduct automated testing in development, and gather stakeholder feedback [43]. The first two strategies involve accessibility at the beginning of the development process instead of considering it an intrusion that may lead to compromises. Including accessibility as a function helps to save time and the budget during the development and evaluation process. The third strategy promotes public participation, which involves gathering from people with disabilities. By adopting the approach, the developers suggested that we produce products that provide the optimal user experience for all users, including users with disabilities. These three strategies should be the norm for developers to ensure accessibility for all types of users. The work and input that the IBM Team Able endorses inclusivity in the IT industry. The results are so effective that they should be an example followed by most companies, especially in the IT industry.

Developer tools are vital in the process of programming DevOps. DevOps is a new and modern approach to software development. This the reason behind the fact that there is currently no specific and distinct definition for "DevOps"; however, it has been used in multiple frameworks [24] [44].

Agile software development approaches have gotten progressively pervasive, and several organizations have discovered they could not accomplish frequent release cadence to a great extent because of various departmental capacities. With an end goal to expel these shortcomings, organizations have moved towards DevOps. As digitalization proceeds, organizations progressively actualize DevOps [45]

DevOps promote the integration and collaboration of development and operations. According to [24], DevOps promotes team members' active cooperation with diverse

interdisciplinary backgrounds, working continuously on operational feature deliveries instead of performing the work separately, which can be time-consuming. Therefore, collaboration in DevOps promotes efficiency through fast and constant work, boosts quick problem resolution, and reduces problems resulting from miscommunication [24].

DevOps, the mix of Development and Operations, is another perspective in the product designing space that got much consideration. Given that DevOps is another term and novel idea presented, no normal comprehension of what it involves has been accomplished at this point [46].

DevOps intends to lessen the time between submitting a framework change and putting the changes into the existing product while guaranteeing high quality [47].

As shown in Figure 1, the DevOps tools are categorized in the versioning control system, code management and analyzing, build system, continuous integration (CI), testing, configuration and provisioning, and continuous deployment (CD) [48]. In addition to the aforementioned, collaboration tools are also included [44].

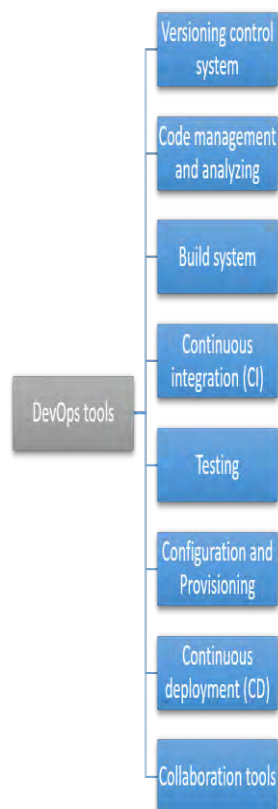


Figure 1: DevOps tools categories

In this thesis, eight tools are selected based primarily on the criteria: current online availability and ii) open and free to use. These kinds of DevOps tools are usually for versioning control systems and collaboration. The others typically are not free or require installation on a computer. A summary of the selected DevOps tools shown in Table 1. DevOps tools are selected from three sources: the journal article "Continuous Integration, Delivery and Deployment" by Mojtaba Shahin and Muhammad Ali Babar, the journal article "DevOps: Concepts, Practices, Tools, Benefits and Challenges" by Georges Bou Gbantous and Asif Gill, and the website of XebiaLabs, and a website' software company that specializes in DevOps [49].

Tools	Category	Information
GitHub <a href="https://github.com/">https://github.com/</a>	Versioning Control	GitHub is a free online versioning control and code management Git repository for developers. It offers both free and paid versions.
GitLab <a href="https://about.gitlab.com/">https://about.gitlab.com/</a>	Versioning Control	GitLab is an online application used for code management, monitoring, and development.
Bitbucket <a href="https://bitbucket.org/">https://bitbucket.org/</a>	Versioning Control	Bitbucket is similar to GitHub.
SonarQube <a href="https://sonarcloud.io">https://sonarcloud.io</a>	Code Analysis	SonarQube is a code analyzer and management tool which can either be installed on a computer or used online. It supports more than 25 programming languages.
Travis CI <a href="https://travis-ci.com/">https://travis-ci.com/</a>	Integration and Development	Travis CI is a continuous integration and deployment tool integrated with GitHub, which supports various languages.

Slack <a href="https://slack.com/">https://slack.com/</a>	Management and Collaboration	Slack is a cloud-based collaboration and communication tool. This tool can transfer files from Trello, GoogleDrive, DropBox, GitHub.
Trello <a href="https://trello.com/">https://trello.com/</a>	Management and Collaboration	Trello is a web-based project management and collaboration application.
Heroku <a href="https://www.heroku.com">https://www.heroku.com</a>	Platform as a Service	Heroku is a cloud platform as a service (PaaS) supporting several programming languages.

Table 1: List of all DevOps tools

## 2.4 RELATED WORKS

It was challenging to find articles related to the research question: "To what extent are online DevOps tools for developers accessible according to WCAG guidelines.". Tried different search keywords containing WCAG and DevOps (Developer tools) using Google scholar. In current programming improvement, developers regularly allude to web search engines as they look for relevant data from many online assets, for example, online instructional exercises, technology blogs, API records, and many more. Search engines, for example, Google, Bing, and Baidu, are among the most well-known and significant tools for developers. These tools are used to finish various coding tasks, such as fixing bugs/exceptions and understanding new code or APIs [50]. There is a lack of research focusing on the accessibility of online tools for software work. In other words, the extent to which DevOps tools comply with accessibility standards as WCAG. Although there are many studies about web accessibility conformance tests in other fields, there is no published study about DevOps tools. For instance, in Norway, the Agency for Public Management and eGovernment (DIFI) carried out annual surveys. The purpose of the survey is to identify the digital barriers in the current modern digital world. One of the surveys' motives was to ensure that the websites with many users addressing the common goals and interests offer an equal opportunity for disabled persons to interact with them [51].



### 2.4.1 The Agency for Public Management and eGovernment (DIFI) and Web accessibility conformance testing

The Agency for Public Management and eGovernment (DIFI) in Norway is vital in the field of Web Accessibility as it is the government agency assigned for the country's information and communication technology (ICT) solutions. They are responsible for improving the quality and accessibility of information and services offered to the public on the internet [52]. After evaluating the website's quality and accessibility, DIFI gives its approval to the websites. According to Norway's action plan of 2018, for universal design, these evaluations occur annually and are conducted according to 35 criteria relating to accessibility, usage quality, and content. The current and updated version of the plan in 2017 introduces 14 measures that must be done and implemented by the governmental agencies and bodies with primary responsibility.

In comparison to the action plan from 2009, which had only five measures for ICT solutions, it is notable that it has been thoroughly expanded. Furthermore, it shows more government bodies working together to achieve a universal design in ICT solutions. It reveals that accessibility and inclusion for people with disabilities is an important topic that should be tackled in a well-thought manner.

In conformance with the UN Convention on the Rights of Persons with Disabilities, Norway released an initial report in 2009 stating the country's actions in response to the convention [52]. On June 21, 2013, regulations for universal design of information and communication technology (ICT) solutions were approved in Norway. These regulations have a legal basis according to Act no. 61, the Anti-discrimination and Accessibility Act [53], recently updated in 2017. According to Norwegian law, the approval of this regulation in 2014 means converting it into a universal design is a legal responsibility. It means that new ICT solutions should be universally designed and existing ICT solutions by 2021. DIFI is responsible for overseeing that these regulations are met and considered.

### 2.4.2 DIFI surveys

DIFI conducts annual surveys of 700 websites to check accessibility, usage quality, and content. The focus areas for evaluations are public administration websites and websites

with particular responsibility for following up. The result of evaluations is posted on Norge.no's website [52].

The selection of web solutions is based on the risk of non-compliance, wherein the risk of non-compliance to the regulation is high. Consequently, a breach in the regulations has consequences for many users. Therefore, DIFI's primary target groups are:

- Banking/finance/insurance
- Travel and accommodation
- Retail trade
- Media
- Education
- Central and local administration.

Level A and AA compliance conform to most legal regulations. In Norway, level AA is the standard for the universal design of a website. There are 12 WCAG 2.0 guidelines, and for each of them, there are several success criteria. For measuring universal design, DIFI's authority has introduced 50 indicators to measure the compliance or non-compliance of the success criteria in WCAG 2.0. The primary purpose of indicators is to provide an overview of the universal design status for a website. Either a score of 1 for compliance or 0 for non-compliance is awarded for each indicator. However, each success criteria have several indicators. Afterward, the total score of each success criteria is added together to find the total sum of the scores of the website [54].

The total average score for both the public and private sectors achieved is 51% in 2014. In total, the results show that 49% from the private sector and 54% from the public sector were given the total number of points possible. Therefore, websites from the public sector are more accessible than those in the private sector. There is a notable difference in the results as the scores may vary from 18-78% for several individual organizations [55].

In 2018 DIFI evaluated 278 websites and given a total of 60% [56]. The score from the recent evaluation show increase in score.

### 3 DESIGN AND IMPLEMENTATION

---

Design and implementation are essential for finding out the answer to the mentioned research question. The website in Table 1 is evaluated by these five procedures listed in Figure 2. Each of these steps is not in chronological order, and it can be altered depending on the type of website (small website, web applications, a website with separable areas, website in multiple versions, website using the responsive design) [26]. For small websites, the entire website is evaluated because of a reduced amount of web pages. Therefore, the evaluation starts from step three. Web applications are dynamic and complex, and for evaluating these types of websites, the sample must be more significant to cover different contents, functions, and processes. Websites with separable areas are considered separate websites while evaluating, except common pages.

Suppose a website has multiple versions depending on different scenarios, such as mobile versions and other languages. These websites are considered separate websites during evaluation. Responsive websites change the flow, order, or behavior of content according to the user's device. These websites will not be considered separately; only the web pages state considered in the evaluation scope [26].

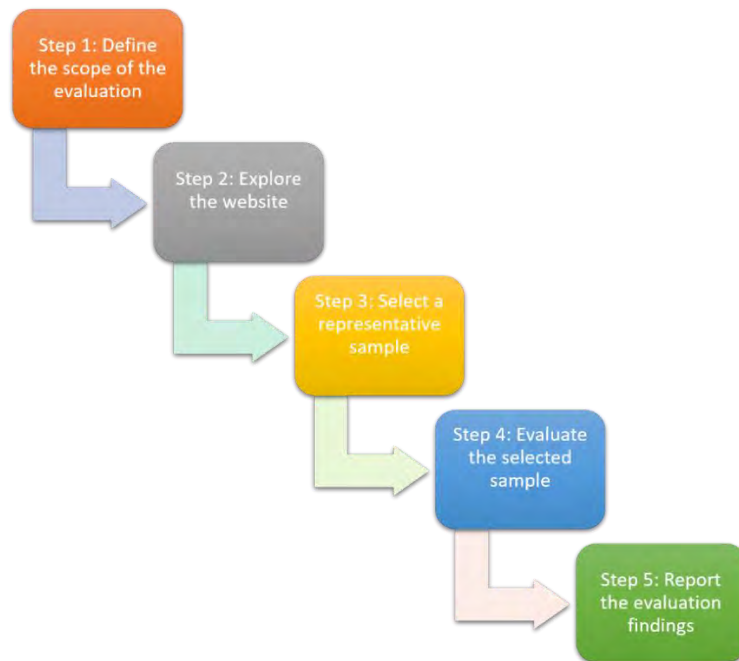


Figure 2: Evaluation procedure

### 3.1 WEBSITE ACCESSIBILITY EVALUATION METHODOLOGY (WCAG-EM):

#### 1. Define the scope of the evaluation:

Defining the scope of the evaluation is the first step for finding out the overall scope of the evaluation.

##### a. Define the scope of the websites:

The targeted website and selected web pages are included in this evaluation process. A website may have different versions such as language, online shop, mobile must be included in this phase.

##### b. Define the conformance target:

The conformance level (A, AA, AAA) of the target website is specified. Conformance level AA is generally recommended and will be used.

##### c. Define an accessibility support baseline:

Accessibility support baseline such as the web browser such as Firefox, Chrome, assistive technologies such as screen readers, alternative keyboards, and other user agents such as media players and plug-ins are defined.

##### d. Define additional evaluation requirements (optional):

Determine additional requirements that the evaluator wants to evaluate the target website for, such as involving users with disabilities to test the targeted website and so on.

## **2. Explore the website:**

The target website is explored for finding out the information/understanding of the website's purpose and functionality. After steps 3 and 4, it might be possible to come back to this step.

### **a. Identify common web pages of the website:**

Explore to identify common web pages of the target website. Common web pages are the most relevant web pages on the entire website, usually shown in the homepage header, footer, and navigation sections.

### **b. Identify the essential functions of the website:**

Explore the target website to identify the website's essential and most essential functions, such as buying a product and filling the form. However, not including all the functionality of the target website.

### **c. Identify the variety of web pages types:**

Identify the variety of web pages and web page states such as different styles, structure, functionality, contents(forms, tables, lists, headings), author of the content or website, change of appearance and behavior depending on the device, user agents such as the browser and dynamic content such as dialog-boxes, pop-ups, error messages, and many more.

### **d. Identify web technologies relied upon:**

Find out the web technologies relied upon to build the target website. Technologies such as HTML, CSS, JavaScript, jQuery, SVG, PDF, and in addition to that, finding out the CMS, CMS version will also help to find out about the technologies that were relied upon to build the target website.

### **e. Identify other relevant web pages:**

Identify other web pages and web page states relevant to people with disabilities and the accessibility of the website. Usually, these pages should be included in step 2.a; if not, they should be included.

## **3. Select a representative sample:**

Selecting a representative sample of a target website (web pages, web page states)

is essential before evaluation. It is crucial to choose valuable samples to give the best evaluation results. If the whole website is evaluated, at that point, this step can be skipped; otherwise, selecting samples of web pages depends on this below factor and divided into two sections; selecting a larger sample or smaller. Larger samples need to be selected when a website has more web pages, is old, complex (many interactions, dynamic), too many varieties (web page types, functionality, technologies, coding), or more confidence in evaluation results. A smaller sample was selected when well-known tools such as content management systems, consistency in coding and development, or website were evaluated before.

**a. Include a structured sample:**

Include a structured sample from all the web pages and web pages state that has been included in previous steps 2.a, 2.b, 2.c, 2.d, 2.e.

**b. Include a randomly selected sample:**

Include a randomly selected sample that has not been selected in step 3.a.

These samples will be at least 10% of the selected structured sample.

Randomly selecting these web pages can be done by finding out the pages on search engines, tools/scripts for traversing, and selecting pages or other methods.

**c. Include complete processes:**

Include all the complete processes in the website sample. If some part of the web page and web page states of the entire process is included in steps 3.a and 3.b, then include the rest of the complete process web pages from the beginning until the end of the complete process.

**4. Evaluate the selected sample:**

The selected sample in step 3 is evaluated according to the chosen conformance level, then compares the evolution of both structured and randomly selected samples.

**a. Check all initial web pages:**

Evaluate structured samples only according to targeted conformance level, and complete processes and randomly selected samples are omitted.

**b. Check all completed processes:**

Evaluate completed processes only according to the targeted conformance

level. It includes all interactions with the form, entering data, error messages, feedback dialog boxes are also evaluated, which is part of the complete process.

c. **Compare structured and random samples:**

Evaluate random samples according to the targeted conformance level, then compare them with results from step 4.a. if the comparison shows new content or new evaluation finding, then the evaluator starts from step 3.

5. **Report the evaluation findings:**

The WCAG-EM report generator tool can be used to report the finding. However, the reporting can be based on WCAG 2.0 version, not 2.1. Therefore, it can be used for generating/creating a new reporting mechanism [57].

a. **Document the outcomes of each step:**

Document the outcomes of each step of WCAG-EM, including the sub-sections.

b. **Record the evaluation specifics (optional):**

All the web pages and web page states are archived, and all the evaluation tools, user-agents, assistive technologies, and different methods are recorded. Recording can be copies of files of the web pages, screenshots, URL addresses, names, and versions of the tool, and so on.

c. **Provide an evaluation statement (optional):**

Provide a statement for the result of the evaluations.

d. **Provide an aggregated score (optional):**

Providing an aggregated score is sometimes misleading; however, it can help understand the evaluation result more efficiently. If the score is used, then the scoring approach must be documented for understanding the aggregated score.

e. **Provide Machine-Readable Reports (optional):**

Provide machine-readable reports for describing test results. The result of this can be used in other programs for generating conclusions and manipulation of the results. EARL (Evaluation and Report Language) is a Vocabulary designed for this purpose [58].

## 3.2 WEB ACCESSIBILITY GUIDELINES 2.1

All principles, guidelines, success criteria, and techniques with conformance level AA will be discussed below. Info below is retrieved from Elsevier accessibility checklist for testing each success criteria [27], WebAIM's WCAG 2 Checklist [59], WCAG 2.1 [60, p. 1] for info about each guideline, success criteria, and WCAG 2.1 meet-list for each success criteria [61].

### 3.2.1 Principle 1. Perceivable

All the information and the component present in a website should be easily perceived by users. It consists of 4 guidelines and 20 success criteria.

#### *Guideline 1.1 Text Alternatives*

Provide text alternatives for any non-text content so it can be perceivable for people with disabilities.

#### **Success Criterion 1.1.1 Non-text content Level A**

Provide a text alternative for non-text content such as images so the users with visual disabilities can understand.

- Provide all images with a descriptive ALT attribute tag or empty alt-tag string for the decorative images.
- Provide a descriptive TITLE attribute for all embedded audio/video, non-image charts, Flash, form elements, and other components that require a textual explanation to be understood.
- Use CAPTCHA for the identification of a person or a robot only. An alternative form of CAPTCHA is provided for people with visual disabilities. Do not use CAPTCHA that relies on visual identification only.
- Use CSS for decorative images and not directly embed those images in HTML.

**Testing:** Use WAVE and AXE tools for finding out all the images on a web page. Look for missing and descriptive alt-text and make sure that it is understandable.



## ***Guideline 1.2 Time-based Media***

Provide alternatives for time-based media such as audio and video to perceive for blind and deaf users.

### **Success Criterion 1.2.1. Audio-only and Video-only      Level A**

Transcript provided for pre-recorded audio-only and video-only. Video without an audio must-have textual transcript.

**Testing:** Check manually if pre-recorded audio-only and video-only has a transcript.

### **Success Criterion 1.2.2. Captions (Pre-recorded audio)      Level A**

Provide captions for pre-recorded audio that should be available.

**Testing:** Check manually if prerecorded audio is present and captions are available.

### **Success Criterion 1.2.3. Audio Description or Media Alternative (Pre-recorded)**

#### **Level A**

All videos with audio should have text or audio descriptions.

**Testing:** Check manually if all videos with audio should have text or audio descriptions.

### **Success Criterion 1.2.4. Captions (Live)      Level AA**

Providing captions for synchronized audio or video.

**Testing:** Check manually if captions available for captions for synchronized audio or video.

### **Success Criterion 1.2.5. Audio Description (Pre-recorded)      Level AA**

Provide an audio description for the pre-recorded video.

**Testing:** Check manually if the audio description available for pre-recorded video.

## ***Guideline 1.3 Adaptable***

Web page content can be presented in different ways without losing information or structure.

### Success Criterion 1.3.1. Info and Relationships      Level A

Information, structure, and relationships can be programmatically determined. Removing all CSS styling should not prevent an understanding of the content. HTML elements such as headings, lists, tables, and other elements convey the content's structure and meaning.

**Testing:** Use web developer toolbar or Wave for removing all CSS styling and test whether it prevents understanding of the content. Use Wave or check manually for HTML markup is used for elements such as tables, headings, and lists.

### Success Criterion 1.3.2. Meaningful Sequence      Level A

The reading sequence is programmatically determined. Removing all CSS styling does not affect the reading sequence, and the tabbing order is logical. Use tab index HTML property to enforce tabbing order.

**Testing:** Use the web developer toolbar to remove all CSS styling and test whether it prevents understanding the content. Check the tabbing order manually, and the elements are in logical reading order.

### Success Criterion 1.3.3. Sensory Characteristics      Level A

Identify the element not only by its sensory characteristics such as shape, size, position on the page.

**Testing:** Check manually and find all the references on the web page that mentions characteristics such as shape, size, location and find out whether it is identifiable without these characteristics

### Success Criterion 1.3.4 Orientation      Level AA

Content is not restricted to a single display orientation, such as portrait or landscape, unless a specific orientation is essential such as bank checks, piano applications, slides in a projector/TV.

**Testing:** Check manually if the content can be displayed in both orientations (portrait/landscape).

### Success Criterion 1.3.5 Identify Input Purpose      Level AA

The input purpose of the elements is identifiable.

**Testing:** Manually check if the input purpose of the elements is identifiable and all the text input HTML elements have autocomplete-tag.

### *Guideline 1.4 Distinguishable*

Content is easy to read and see for the users, and there is enough color contrast separating foreground and background.

### Success Criterion 1.4.1 Use of Color    Level A

Make sure that the color is not used to convey information. If a color is used to convey information, make sure text is also available to convey the same type of information. A link should be easy to identify and also should not rely solely on color for identification. Make links bold, underline the icon next to links, or find other solutions other than color. Finally, forums should not rely on colors too. Forums should have text labels for identifying required fields or fields with errors.

**Testing:** Check manually. Make sure the web page does not rely solely on colors to convey information.

### Success Criterion 1.4.2 Audio Control Level A

Audio controls (play, play, stop, volume change) available for all audio clips or for clips that play automatically for more than 3 seconds.

**Testing:** Do manual checking to find out the result if present on a page.

### Success Criterion 1.4.3 Contrast (Minimum) Level AA

Text or images of text has to have a contrast ratio of at least 4.5:1. For small text, the ratio is 4.5:1, and for larger, the minimum ratio is 3:1. A text which is part of a logo or brand has no contrast requirement.

**Testing:** Use one of these tools such as WebAIM Color Contrast Checker, Paciello Group, or A11Y Color Contrast Accessibility Validator for testing.

#### Success Criterion 1.4.4 Resize text Level AA

Text resized up to 200% without assistive technology except for images of text and captions.

**Testing:** Check manually.

#### Success Criterion 1.4.5 Images of Text Level AA

Text is used rather than images of text, except the visual presentation is essential such as in logos.

**Testing:** Check manually if images of text are present and determine if they are used for essential visual presentation. If not, then this criterion fails.

#### Success Criterion 1.4.10 ReflowLevel AA

If the content of the web page is zoomed in to a large degree, then it should be able to preserve the content without the loss of information or functionality and without requiring scrolling in both directions (horizontal, vertical); except if the content requires both directions (images, maps, diagrams, video, games, presentations, data tables). Pixels for vertical and horizontal scrolling should be at a width equivalent to 320 pixels and a height equivalent to 256 pixels.

**Testing:** Check manually.

#### Success Criterion 1.4.11 Non-Text Contrast Level AA

Visual objects should have a minimum contrast ratio of 3:1 against adjacent colors, except if the visual object is an essential part of conveying information. Visual objects are user interface components, graphical objects.

**Testing:** Use Paciello Group Color Contrast Analyzer or WebAim Contrast Checker and Wave for testing.

#### Success Criterion 1.4.12 Text Spacing Level AA

Text-based CSS styling can be changed to lower numbers without loss of content or functionality.

CSS styling:

- Line height (line spacing) to at least 1.5 times the font size.
- Space following paragraphs to at least two times the font size.
- Letter spacing (tracking) to at least 0.12 times the font size.
- Word spacing to at least 0.16 times the font size.

**Testing:** Use the following bookmarklet to activate the minimum CSS values to a site: [Text Spacing](#). Add a link to the bookmarks bar on a browser. While on a page, click on the bookmark bar (Text Spacing). Check if these styling changes made content unreadable or cut-off. If bookmarklet did not work on a page, insert the below code in the head-tag.

```
<style>
* {
  line-height: 1.5 !important;
  letter-spacing: 0.12em !important;
  word-spacing: 0.16em !important;
}
</style>
```

#### Success Criterion 1.4.13 Content on Hover or Focus      Level AA

Additional content that appears on mouse, pointer, or keyboard focus must be dismissible, hoverable, and persistent. Dismissible means that the user can close the content without moving the mouse or keyboard focus (Esc button). Hoverable means that the pointer can move over the additional content without disappearing. Persistent means that the additional content is visible until the user closes or the focus is moved away. Additional content can be sub-menus, pop-ups, tooltips, or other content that appears on hover or focus.

**Testing:** Check manually if the content of hover or focus is dismissible, hoverable, and persistent.

## 3.2.2 Principle 2. Operable

All components present on a website should be easily operable by users. It consists of 5 guidelines and 17 success criteria.

### *Guideline 2.1 Keyboard Accessible*

All functionality should be accessible by the keyboard.

#### **Success Criterion 2.1.1. Keyboard                      Level A**

All functionality should be accessible by keyboard except for drawing, handwriting, or similar activities. People with low motoring skills or any visual disability cannot access the mouse. Therefore, all clickable components must be accessible by keyboard also. Every component on a web page which is accessible by mouse should also be accessible by keyboard. If drag and drop functionality are available by mouse, it should be accessible by copy and paste method on keyboard.

**Testing:** Check manually if web page interactive elements are accessible by keyboard tabbing.

#### **Success Criterion 2.1.2. No Keyboard Trap    Level A**

No keyboard trap should be available while surfing or navigating through content, a section of a web page. Users must be able to exit by keyboard from different types of plug-ins such as java applets, flash files, and other plug-ins.

**Testing:** Check for the keyboard trap manually.

#### **Success Criterion 2.1.4. Character Key Shortcuts    Level A**

If a single Keyboard shortcut (Letters, numbers, symbols, punctuations) is available in web content, the shortcut should be able to turn off, remap, or active only while the component is on focus.

**Testing:** Check manually.

### *Guideline 2.2 Enough Time*

There should be enough time for users to read and use the content.

### Success Criterion 2.2.1 Timing Adjustable Level A

If a time limit is present on a web page, users should be warned that the time limit is longer than 20 hours or time limit is essential to an activity, event such as auction, exam, and other activities. Otherwise, the users should be able to turn off, extend the time limit beforehand, or be warned at least 20 seconds before expiring.

**Testing:** Test manually.

### Success Criterion 2.2.2 Pause, Stop, Hide Level A

Users should have the ability to pause, stop, or hide the moving, scrolling, or auto-updating content on a web page. This can cause trouble for people with disabilities that cannot read quickly or track the movement of the moving object (text, video, audio, or animation) on a web page. The Pause, stop, hide controls are correct for a moving, blinking, scrolling object that starts automatically and lasts more than 5 seconds. If auto-updating is not part of the essential activity, the user should pause, stop, or hide it.

**Testing:** Manually check if objects present in a web page that is moving, blinking, or auto-updating; if so, test if it follows the rules.

## *Guideline 2.3 Seizures and Physical Reactions*

A website should not have content that causes users seizures or physical reactions.

### Success Criterion 2.3.1 Three Flashes or Below Threshold Level A

Flashing can cause trouble with people with disabilities (epilepsy, photosensitive); therefore, no more than three flashes per 1 second or it should be below the flash threshold.

**Testing:** Manually check if flashing occurs on a web page. If it does, then check if it is complying.

## *Guideline 2.4 Navigable*

Users should be able to navigate, find content and their location.

### Success Criterion 2.4.1 Bypass Blocks Level A

Users should bypass the repetitive blocks such as menus and headers to go to the main content. Bypassing the repetitive block on a web page can be a link such as "Go to main

content" on top, information on using a screen reader, or other tools used for reading content.

**Testing:** Use the web developer toolbar to remove all CSS styling and test whether it bypasses to main content.

#### **Success Criterion 2.4.2 Page Titled    Level A**

Every page must have a title describing the purpose or topic of the web page. A title tag is used in a header section of the HTML page

**Testing:** Check if the title tag available in a header section of the HTML page and describes the topic or the purpose of the web page. It also can be checked by seeing the page tab on the browser or by bookmarking it.

#### **Success Criterion 2.4.3 Focus Order    Level A**

Users should be able to navigate sequentially without deterring the meaning and operability of the web page. Keyboard tabbing is used to navigate through web pages. Tabindex is used to enforce the tabbing order on a web page, but this should not deter the meaning and operability of the web page. If a user gets a dialog box, the focus should be back on the previous section rather than starting.

**Testing:** Use keyboard tabbing for navigating sequentially through a web page, and it preserves the meaning and operation of the web page. Users should be able to close the dialog box with the keyboard only, and the focus should be back on the previous section before the dialog box was opened.

#### **Success Criterion 2.4.4 Link Purpose (In Context)    Level A**

Link purpose should be easily understood from the text link or context. If link text is not clear enough, then use title property inside the link tag, for example: `<a href="Messi.html" title="View more details about best player on planet Earth">Lionell Messi</a>`

**Testing:** Check manually if the link text is clearly labeled.



### Success Criterion 2.4.5 Multiple Ways      Level AA

A website should have more than one way to navigate, such as sitemap, table of content, search box, and others, except where the web page results from a step in a process.

**Testing:** Manually check if there are multiple ways to navigate to other pages.

### Success Criterion 2.4.6 Headings and Labels Level AA

Make sure that the heading and label are clear and effortlessly describes their purpose.

**Testing:** Test the label tags and heading tags that are clear and describes the purpose. If a label tag is required in a form, then required must be in parenthesis and easily identifiable. The label is not required to submit, reset, image, and hidden or script buttons for input type.

### Success Criterion 2.4.7 Focus Visible    Level AA

Visible focus indicators are available on a web page where the keyboard's current page element is on focus. Use CSS styling when an element receives focus.

**Testing:** check manually if a visible focus indicator is available when elements on the web page receive focus.

## *Guideline 2.5 Input Modalities*

Other input modalities available for a user to operate functionality a web page beyond the keyboard.

### Success Criterion 2.5.1 Pointer Gestures      Level A

All operations on a web page must be simple gestures such as one single click or touch. If more complex operations exist, make sure single click/touch is available except if complex gestures such as multi-touch or signature are essential.

**Testing:** Test web page manually if complying with the success criterion.

### Success Criterion 2.5.2 Pointer Cancellation Level A

Users should be able to recover from accidental or wrong pointer click and touch.

- No down-event of the pointer is used to execute part of the function.

- Able to abort or undo before completion of the function.
- Up reverse available to reverse the outcome of the down event of the pointer.
- No pointer Cancellation needed If function completion on the down-event is essential

**Testing:** Check manually.

### Success Criterion 2.5.3 Label in Name Level A

User interface components with labels including text, images of text, and the name contain the visually presented text.

**Testing:** Check manually. Check manually. For all UI components with a visible label, the accessible name of the control contains the same letters in the same order as the visible label.

### Success Criterion 2.5.4 Motion Actuation Level A

Other input types like buttons can do motion inputs such as orientation change, tilting, and shaking. Motion activation can be disabled if enabled by accident. Exceptions are made if motion is available by accessibility supported interface or is an essential part of an activity.

## 3.2.3 Principle 3. Understand

All the information and the component present in a website should be easily understandable and readable by users. It consists of 3 guidelines and 10 success criteria.

### *Guideline 3.1 Readable*

Text content should be easily understandable and readable by users.

### Success Criterion 3.1.1 Language of Page Level A

The language of the page must be determined to allow users with disabilities that use assistive tools to read the content of the page. Use lang-property in HTML tag for specifying the language. Example: `<html lang="en">`

**Testing:** Use AXE, WAVE, or HTML inspection for testing.

### Success Criterion 3.1.2 Language of Parts    Level AA

Mention the language for a phrase, quote, text which is not part of the entire web page.

Example: `<blockquote lang="de">Text her</blockquote>`.

### *Guideline 3.2 Predictable*

Predictability is necessary to be able to operate effortlessly on a web page for a user.

### Success Criterion 3.2.1 On Focus    Level A

When user interface components receive focus from the keyboard or a mouse, they should not behave unexpectedly, such as form submitted mistakenly, open a new window, and re-shift the focus to other components. Behave unexpectedly makes it hard for all users, especially for the blinds.

**Testing:** Check manually for testing unexpected actions from user interface components such as form elements.

### Success Criterion 3.2.2 On Input    Level A

Any changes to user interface components should not cause changes in context unless the user has been notified beforehand. Changes can include turning on or off the checkbox, radio buttons, select items from the dropdown menu, and filling the text field. Unexpected actions can be a new window opens or the content of the page changes.

**Testing:** Check manually for testing unexpected actions from user interface components such as form elements.

### Success Criterion 3.2.3 Consistent Navigation    Level AA

Navigation menus or any repetitive items on a web page such as login, "skip to content" link, or search box should be in the exact location on the entire website.

**Testing:** check manually if menus or any repetitive items appear, are located, and work the same on every page.

### Success Criterion 3.2.4 Consistent Identification    Level AA

User interface components with the same functionality in a set of web pages should be identified consistently on every page without any confusion.

**Testing:** Check manually if user interface components are identified consistently across every page.

### *Guideline 3.3 Input Assistance*

Input Assistance helps users to avoid and correct mistakes.

#### **Success Criterion 3.3.1 Error Identification    Level A**

Input error from users is easily identified and described to the user for correcting. Informing the user by describing the error as a text message and marking it by color or image.

**Testing:** Test all the form elements by clicking the submit button and check whether text and mark feedback is received while form elements have incorrect data.

#### **Success Criterion 3.3.2 Labels or Instructions        Level A**

Form input elements are clearly labeled, or instruction provided.

- Use for property in label tag for associating the labels to the form input elements for example: `<label for="element-id">Name</label>`.
- All form elements should have clear labels and near to their labels.
- Provide text labels and colors to identify the mandatory field in a form.
- Provide an example for correct input, such as date format.

**Testing:** Use AChecker for form elements and manually check if all labels are clear or instructions provided.

#### **Success Criterion 3.3.3 Error Suggestion        Level AA**

Suggestions are provided if users input incorrect data.

**Testing:** Check manually; enter wrong input for form elements and check whether valid input suggestion provided.

#### **Success Criterion 3.3.4 Error Prevention (Legal, Financial, Data)        Level AA**

For a web page that causes legal, financial, or data commitments, input data can be review and corrected before final submission, and the user can revert the submission.

**Testing:** Manually check if it is complying with the success criterion.

### 3.2.4 Principle 4. Robust

All the information and components present in a website should be robust and easily interpreted by user agents and assistive technologies. It consists of 1 guideline and three success criteria.

#### *Guideline 4.1 Compatible*

Maximize the compatibility of the content for the current and future user agents, for example, web browsers and assistive technologies.

#### Success Criterion 4.1.1 Parsing Level A

Content using markup languages such as HTML, elements should follow these criteria:

- Every ID should be unique.
- Elements have complete start and end tags and are nested accordingly to their specifications.
- Elements do not contain duplicate attributes.
- HTML code should pass the HTML validation test.

**Testing:** Use HTML-validator for HTML code validation and AXE or Wave for duplicate IDs.

#### Success Criterion 4.1.2 Name, Role, Value Level A

Programmatically determined all user interface components such as form elements, links, and other components generated by scripts, name, role, and value.

If using non-standard controls, avoid the following:

- Non-standard controls, such as those created by Flash, Java, other plug-ins, or components created by scripts.
- Avoid clickable <div>s and <span>s.
- If non-standard controls are used, make sure that it is keyboard accessible and receive focus, and change the state while using the keyboard.

**Testing:** Use axe to check for clickable <div>s and <span>s. check if not-standard controls have unexpected actions.

### Success Criterion 4.1.3 Status Messages      Level AA

Status messages can be programmatically determined by user agents, assistive technologies through role or properties. The status message is a visual message that informs the user about the success or outcome of action on a web page on waiting, progress state, or occurred error. Status roles can include: role="status", role="marquee", role="progress bar", role="timer", role="alert dialog", and many more.

Avoid these below rules otherwise, will result in failing the success criterion:

- Using role="alert" or aria-live="assertive" on content which is not important and time-sensitive (future link).
- Avoid visibility change events to hide or display a document without switching the document's live regions between active and inactive (future link).

**Testing:** Use NVDA software. Manually check if it is complying with the success criterion.

## 3.3 TEST ENVIRONMENT

The testing environment includes these below setups:

- Operative system: Windows 10 on a local PC.
- Evaluation tools: Detail description of the tools used is in the section below.
- Data storage: First, the testing data will be stored on a local PC then transfer into GitHub online repository (<https://github.com/samadhiof/samadhiof.git>). Stored data consist of, Web pages and CSV files, screenshots, the word file for reporting summary.
- Reporting: First, the data will be stored in an Excel file, generating the CSV from an Excel file for machine-Readable reports, and further used in the evaluation and the conclusion sections.

## 3.4 EVALUATION TOOLS:

Evaluating some success criterion requires specific auditing tools. These tools are retrieved from the Elsevier accessibility checklist [27], WAI-Web Accessibility Evaluation Tools List [28].

### 3.4.1 Firefox Developer Tools

Firefox developer tools are powerful inspection tools used for debugging, testing, and editing HTML, CSS, and JavaScript. Most of the success criteria are checked manually. Therefore, Firefox is used inspection tool for this purpose.

Website link: [Firefox Developer Tools](#)

### 3.4.2 HTML validator

HTML validator developed by W3C checks if HTML markup language is used correctly on a web page and gives feedback to correct the errors.

Website link: [HTML validator](#)

### 3.4.3 Wappalyzer

Wappalyzer is an open-source web technology analyzer and used to find web technologies relied upon on a website.

Website link: [Wappalyzer](#)

### 3.4.4 WebAIM

WebAim is used for checking color contrast; it provides awareness about web accessibility. This tool is used for testing success criteria 1.4.3 and 1.4.11.

Website link: [WebAIM](#)

### 3.4.5 WAVE

The WebAIM community developed WAVE. WAVE is a browser plugin installed on Firefox or Chrome for testing success criteria 1.1.1, 3.1.1, 4.1.1, 1.4.3, and 1.4.11.

Website link: [WAVE](#)

### 3.4.6 AChecker

AChecker online tool is used for many purposes, such as HTML/CSS validation, WCAG check, and user feedback. Only one page can be tested at a time.

Website link: [AChecker](#)

### 3.4.7 Paciello Group Contrast Analyzer

Paciello is a program installed on the operative system to check color contrast, graphical controls, visual indicators, and a color blindness simulator. It tests success criteria 1.4.3 and 1.4.11.

Website link: [Paciello Group Contrast Analyzer](#)

### 3.4.8 A11Y Color Contrast Accessibility Validator

An A11Y Color is an online tool used for checking color contrast based on success criterion 1.4.3.

Website link: [A11Y Color Contrast Accessibility Validator](#)

### 3.4.9 AXE: Accessibility for Development Teams

AXE is a browser plugin similar to WAVE, installed on Firefox or Chrome for testing success criteria 1.1.1, 1.4.3, 3.1.1, 4.1.1, and 4.1.2.

Website link: [AXE: Accessibility for Development Teams](#)

### 3.4.10 Accessibility Bookmarklets

Accessibility Bookmarklets makes it easier to highlight accessibility features of web pages such as landmarks, headings, list, images, and forms.

Website link: [Accessibility Bookmarklets](#)

### 3.4.11 JAWS

JAWS is a screen reader software installed on an operative system. It can evaluate web pages and emulate how people with vision impairment or learning disabilities use a website.

Website link: [JAWS](#)

### 3.4.12 NVDA

NVDA is similar to JAWS, which is free and open-source screen reader software.



Website link: [NVDA](#)

### 3.5 SELECTION OF THE SAMPLE PAGES

Since the focus of the thesis is testing the DevOps tool, and most relevant pages for the DevOps are the pages after signing in page. Testing a much larger sample is time-consuming; therefore, the sample for testing is a maximum of five pages, from the homepage to the signing in page and the pages afterward. Other pages that are shown in menus, footer, the sitemap will be excluded.

### 3.6 SCORING PROCEDURE

To better understand the results from each DevOps tools, numbers are introduced for meeting each success criterion; 4 for a pass, 2-3 for according to the degree of compliance, when the web page complies with the success criterion, 1 as not-present when the success criterion is not-present on a page, and 0 as fail when web pages do not comply with the success criterion. The lowest and total score presents the lowest score chosen from all the pages and the sum of all pages.

## 4 EVALUATION

---

Each DevOps tool in Table 1 will be evaluated based on 50 success criteria from WCAG 2.1. Success criteria cover the 30 conformance level A and 20 conformance level AA, using the five procedures listed in Figure 2. The test environment is described in section 3.3 and further discussed in this section.

Additional data are available in the [GitHub repository](#) for further information. Documents include word files (result and summary of testing), screenshots of evaluation tools (AXE, WAVE, HTML validator), and HTML pages used for testing. Visit the repository for more testing results. Evaluation tools used are Firefox Developer Tools, AChecker, NVDA, AXE, WAVE, HTML validator, and Accessibility Bookmarklets.

## 4.1 GITHUB

GitHub is a free online versioning control and code management Git repository for developers. It offers both free and paid versions.

### 4.1.1 Evaluation Scope

**Scope of the website:** Web pages in <https://github.com/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.1.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies includes: HTML5, CSS, JavaScript, JQuery, Bootstrap, WordPress, ReCAPTCHA, Java, AngularJS.

### 4.1.3 Representative Sample

Web pages in <https://github.com/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

### 4.1.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Homepage.

Link: <https://github.com/>

#### *Page 2*

Sign up page.

Link: <https://github.com/join>

### Page 3

Login page.

Link: <https://github.com/login>

### Page 4

GitHub after login page. Testing the different functions for creating a new repository.

Link: <https://github.com/>

### Page 5

GitHub after login page. Getting inside the created repository and testing different sections inside the repository, such as code, issues, pull requests, actions, projects, security, insights, and settings.

Links: <https://github.com/samadhiof/samadhiof> and <https://github.com/samadhiof>

### Overall result:

Table 2 shows the overall result for all the representative samples.

Success criteria	Page 1	Page 2	Page 3	Page 4	Page 5	Lowest	Total score	Conformance level
1.1.1 Non-text content	4	4	4	4	3	3	19	A
1.2.1. Audio-only and Video-only	1	1	1	1	1	1	5	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	1	5	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	1	1	1	1	1	5	A
1.2.4. Captions (Live)	1	1	1	1	1	1	5	AA
1.2.5. Audio Description (Pre-recorded)	1	1	1	1	1	1	5	AA
1.3.1. Info and Relationships	4	4	4	0	4	0	16	A

1.3.2. Meaningful Sequence	4	0	4	0	4	0	12	A
1.3.3. Sensory Characteristics	4	4	4	4	4	4	20	A
1.3.4 Orientation	4	4	4	4	4	4	20	AA
1.3.5 Identify Input Purpose	4	4	4	4	4	4	20	AA
1.4.1 Use of Color	4	4	4	4	4	4	20	A
1.4.2 Audio Control	1	1	1	1	1	1	5	A
1.4.3 Contrast (Minimum)	2	4	4	3	2	2	15	AA
1.4.4 Resize text	4	4	4	4	4	4	20	AA
1.4.5 Images of Text	4	4	4	4	4	4	20	AA
1.4.10 Reflow	4	4	4	0	0	0	12	AA
1.4.11 Non-Text Contrast	2	4	4	2	1	2	13	AA
1.4.12 Text Spacing	4	4	4	4	4	4	20	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	20	AA
2.1.1. Keyboard	4	0	4	4	4	0	16	A
2.1.2. No Keyboard Trap	4	0	4	4	4	0	16	A
2.1.4. Character Key Shortcuts	1	1	1	1	1	1	5	A
2.2.1 Timing Adjustable	1	1	1	1	1	1	5	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	5	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	5	A
2.4.1 Bypass Blocks	4	4	4	4	4	4	20	A
2.4.2 Page Titled	4	4	4	4	4	4	20	A
2.4.3 Focus Order	4	0	4	4	4	0	16	A

2.4.4 Link Purpose (In Context)	2	3	4	4	4	2	17	A
2.4.5 Multiple Ways	4	4	4	4	4	4	20	AA
2.4.6 Headings and Labels	3	3	3	4	3	3	16	AA
2.4.7 Focus Visible	4	4	4	4	4	4	20	AA
2.5.1 Pointer Gestures	4	4	4	4	4	4	20	A
2.5.2 Pointer Cancellation	4	4	4	4	4	4	20	A
2.5.3 Label in Name	4	4	4	4	4	4	20	A
2.5.4 Motion Actuation	1	1	1	1	1	1	5	A
3.1.1 Language of Page	4	4	4	4	4	4	20	A
3.1.2 Language of Parts	1	1	1	1	1	1	5	AA
3.2.1 On Focus	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	AA
3.2.4 Consistent Identification	4	4	4	4	4	4	20	AA
3.3.1 Error Identification	4	4	4	4	4	4	20	A
3.3.2 Labels or Instructions	4	4	4	4	4	4	20	A
3.3.3 Error Suggestion	4	4	4	4	4	4	20	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	4	4	20	AA
4.1.1 Parsing	2	3	2	4	2	2	13	A
4.1.2 Name, Role, Value	4	4	4	4	4	4	20	A
4.1.3 Status Messages	0	4	0	1	1	0	6	AA

Table 2: GitHub results

### 4.1.5 Evaluation Findings

As per the findings, GitHub has a lowest score of 126 (68 A, 58 AA) out of 200, which is 63% (57% A, 73% AA), and total score of 747 out of 1000, which is 74.7%, as shown in Figure 5. The total number of pass criteria is 25 (13 A, 12 AA), 6 (3 A, 3 AA) half-pass, 12 (9 A, 3 AA) not-present, 7 (5A, 2 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 3, the pass percentage is 50% (43.3% A, 60% AA), 12% (10% A, 15% AA) half-pass, 24% (30% A, 15% AA) not-present, and 14% (16.67% A, 10% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of the pass is 37 (22 A, 15 AA) and fail as 13 (8 A, 5 AA). In percentage, as shown in Figure 4, pass is 74% (75% A, 74% AA) and fail as 26% (25% A, 26% AA).

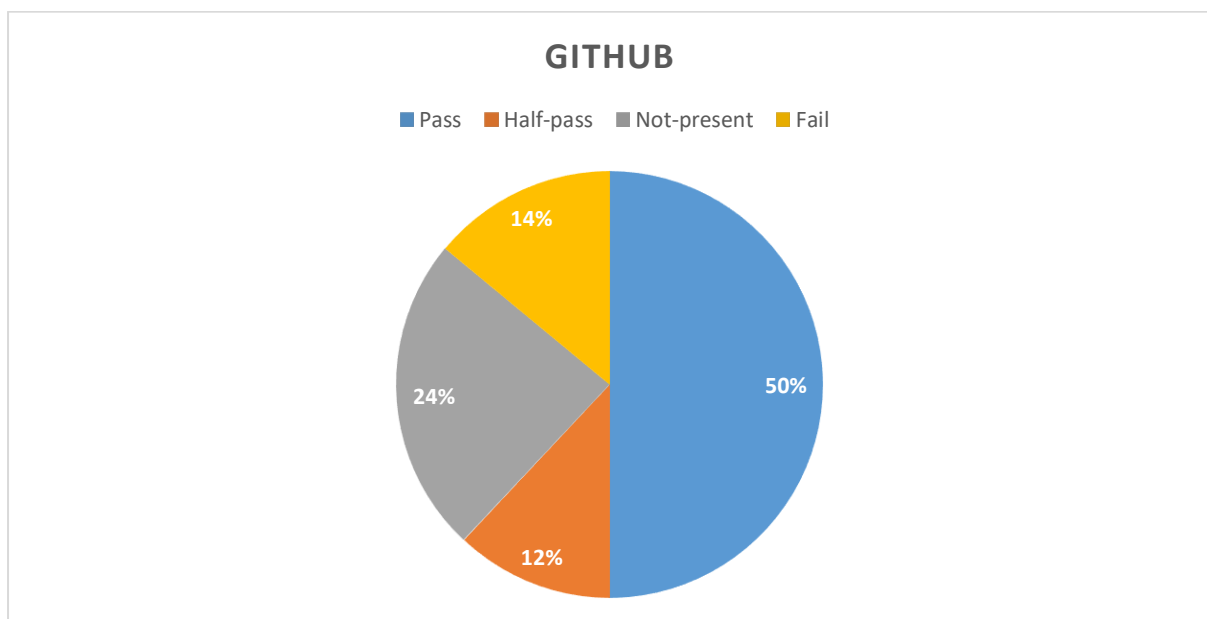


Figure 3: GitHub lowest score

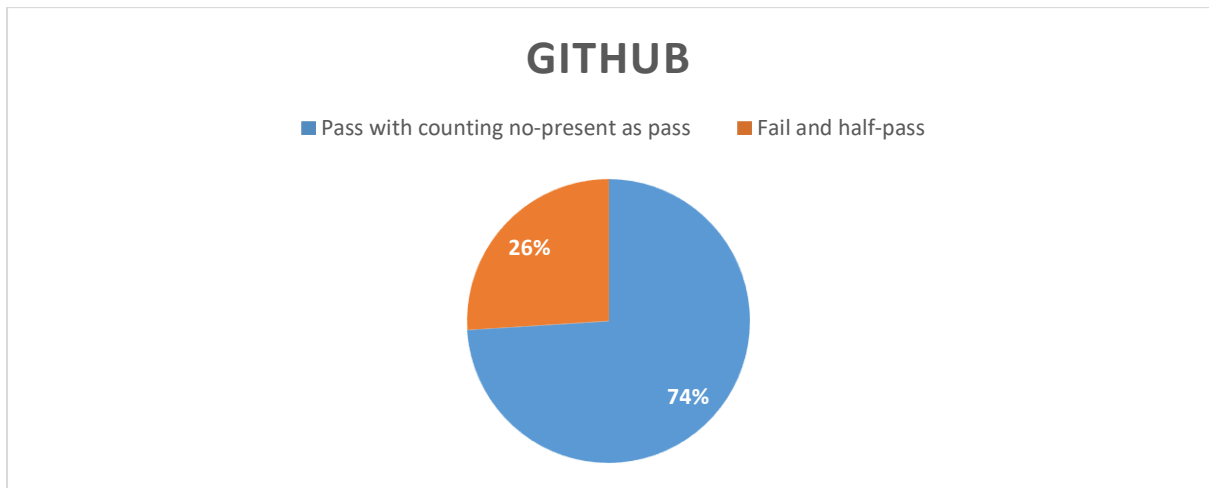


Figure 4: GitHub result counting no-present as pass

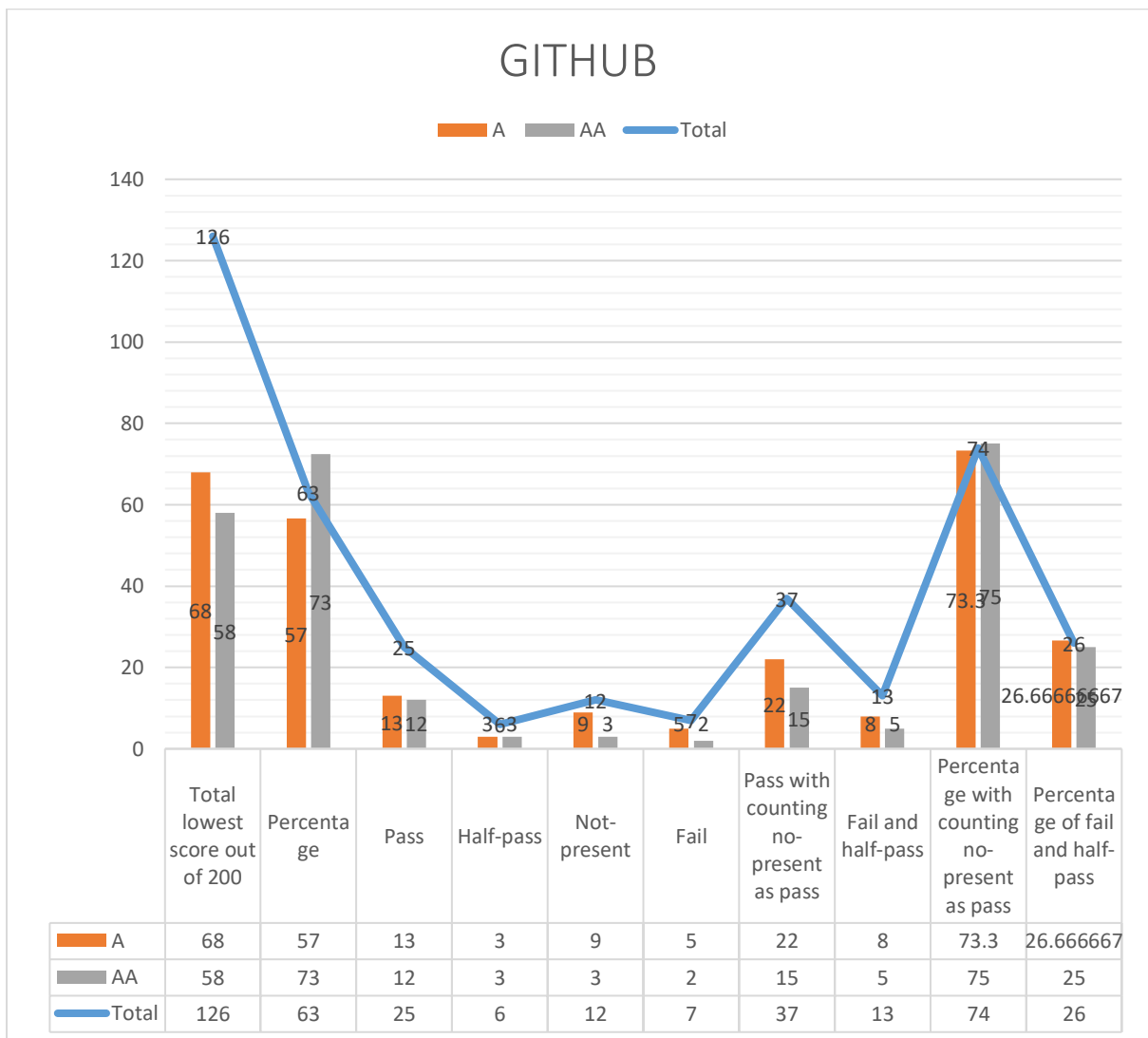


Figure 5: GitHub overall result

## 4.2 GITLAB

GitLab is an online application used for code management, monitoring, and development.

### 4.2.1 Evaluation Scope

**Scope of the website:** Web pages in <https://about.gitlab.com/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.2.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies include HTML5, CSS, JavaScript, JQuery, Bootstrap, Ruby, Font Awesome.

### 4.2.3 Representative Sample

Web pages in <https://about.gitlab.com/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

### 4.2.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Homepage.

Link: <https://about.gitlab.com/>

#### *Page 2*

Sign up page.

Link: [https://gitlab.com/users/sign\\_up](https://gitlab.com/users/sign_up)

#### *Page 3*

Login page.



Link: [https://gitlab.com/users/sign\\_in](https://gitlab.com/users/sign_in)

#### Page 4

GitLab after login page. Test different functionality for creating a new repository.

Links: <https://gitlab.com/> and <https://gitlab.com/projects/new>

#### Page 5

GitLab after login page. Getting inside the created repository and testing different sections inside the repository such as code, issues, pull requests, actions, projects, security, insights, and settings.

Link: <https://gitlab.com/samadhiof/test-gitlab>

#### Overall result:

Table 3 shows the overall result for all the representative samples.

Success criteria	Pag e 1	Pag e 2	Pag e 3	Pag e 4	Pag e 5	Lowest	Total score	Conformanc e level
1.1.1 Non-text content	3	4	4	0	2	0	13	A
1.2.1. Audio-only and Video-only	0	1	1	1	1	0	4	A
1.2.2. Captions (Pre- recorded audio)	1	1	1	1	1	1	5	A
1.2.3. Audio Description or Media Alternative (Pre- recorded)	0	1	1	1	1	0	4	A
1.2.4. Captions (Live)	1	1	1	1	1	1	5	AA
1.2.5. Audio Description (Pre-recorded)	0	1	1	1	1	0	4	AA

1.3.1. Info and Relationships	1	3	4	0	0	0	8	A
1.3.2. Meaningful Sequence	4	4	4	4	4	4	20	A
1.3.3. Sensory Characteristics	4	4	4	4	4	4	20	A
1.3.4 Orientation	4	4	4	4	4	4	20	AA
1.3.5 Identify Input Purpose	3	4	4	4	4	3	19	AA
1.4.1 Use of Color	4	4	4	4	4	4	20	A
1.4.2 Audio Control	4	1	1	1	1	4	8	A
1.4.3 Contrast (Minimum)	2	4	2	2	2	2	12	AA
1.4.4 Resize text	4	4	4	4	4	4	20	AA
1.4.5 Images of Text	4	4	4	4	4	4	20	AA
1.4.10 Reflow	4	4	4	4	4	4	20	AA
1.4.11 Non-Text Contrast	2	4	2	2	1	2	11	AA
1.4.12 Text Spacing	4	4	4	4	4	4	20	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	20	AA
2.1.1. Keyboard	4	4	4	4	4	4	20	A
2.1.2. No Keyboard Trap	4	4	4	4	4	4	20	A

2.1.4. Character Key Shortcuts	1	1	1	1	1	1	5	A
2.2.1 Timing Adjustable	1	1	1	1	1	1	5	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	5	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	5	A
2.4.1 Bypass Blocks	0	0	0	4	4	0	8	A
2.4.2 Page Titled	4	4	4	4	4	4	20	A
2.4.3 Focus Order	4	4	4	4	4	4	20	A
2.4.4 Link Purpose (In Context)	3	4	4	3	2	2	16	A
2.4.5 Multiple Ways	4	4	4	4	4	4	20	AA
2.4.6 Headings and Labels	1	4	3	1	1	3	10	AA
2.4.7 Focus Visible	2	4	4	4	4	2	18	AA
2.5.1 Pointer Gestures	4	4	4	4	4	4	20	A
2.5.2 Pointer Cancellation	4	4	4	4	4	4	20	A
2.5.3 Label in Name	4	4	4	4	4	4	20	A
2.5.4 Motion Actuation	1	1	1	1	1	1	5	A
3.1.1 Language of Page	4	0	0	4	4	0	12	A
3.1.2 Language of Parts	1	1	1	1	1	1	5	AA
3.2.1 On Focus	4	4	4	4	4	4	20	A

3.2.2 On Input	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	AA
3.2.4 Consistent Identification	4	4	4	4	4	4	20	AA
3.3.1 Error Identification	4	4	4	4	4	4	20	A
3.3.2 Labels or Instructions	0	4	0	0	0	0	4	A
3.3.3 Error Suggestion	4	4	4	4	4	4	20	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	4	4	20	AA
4.1.1 Parsing	2	3	2	1	2	2	10	A
4.1.2 Name, Role, Value	2	4	4	3	2	2	15	A
4.1.3 Status Messages	1	4	4	1	1	4	11	AA

Table 3: GitLab results

## 4.2.5 Evaluation Findings

As per findings, GitLab has a lowest score of 130 (68 A, 62 AA) out of 200, which is 65% (57% A, 78% AA), and total score of 707 out of 1000, which is 70.7%, as shown in Figure 8. The total number of pass criteria is 26 (14 A, 12 AA), 8 (3 A, 5 AA) half-pass, 8 (6 A, 2 AA) not-present, 8 (7A, 1 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 6, pass percentage is 52% (46.6% A, 60% AA), 16% (10% A, 25% AA) half-pass, 16% (20% A, 10% AA) not-present, and 16% (23.3% A, 5% AA) fail. However, if we count not-present criteria as pass and half-pass as failing, then the total number of the pass is 34 (20 A, 14 AA) and fail as 16 (10 A, 6 AA). In percentage, as shown in Figure 7, pass percentage is 68% (66.7% A, 70% AA) and fail as 32% (33.3% A, 30% AA).

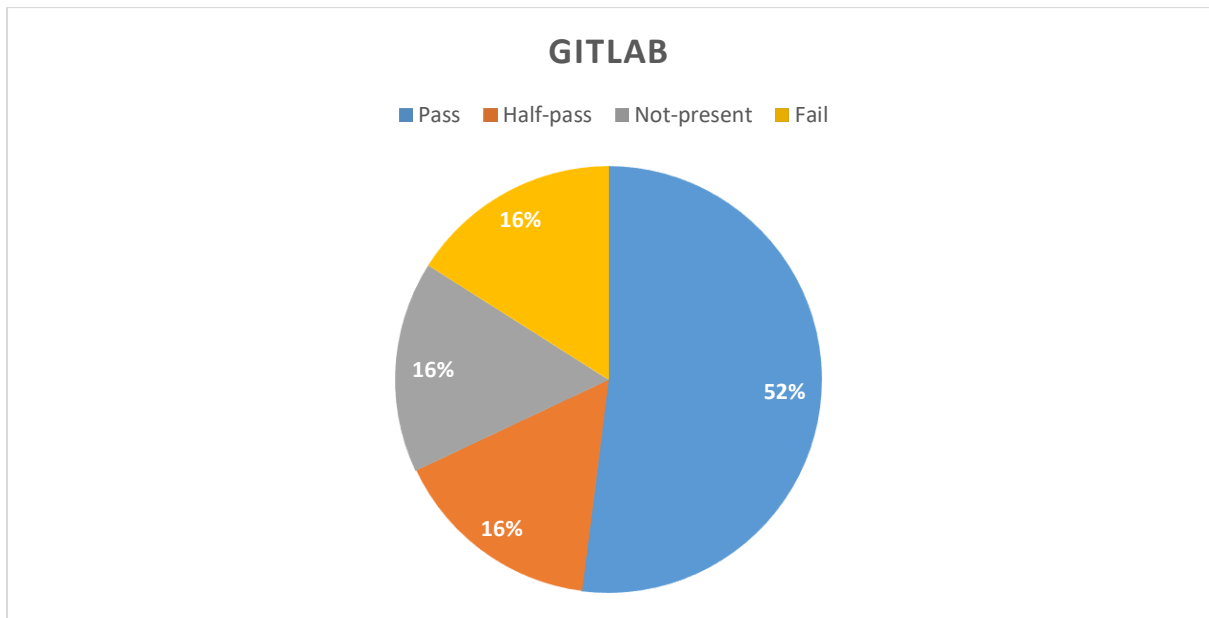


Figure 6: GitLab lowest score

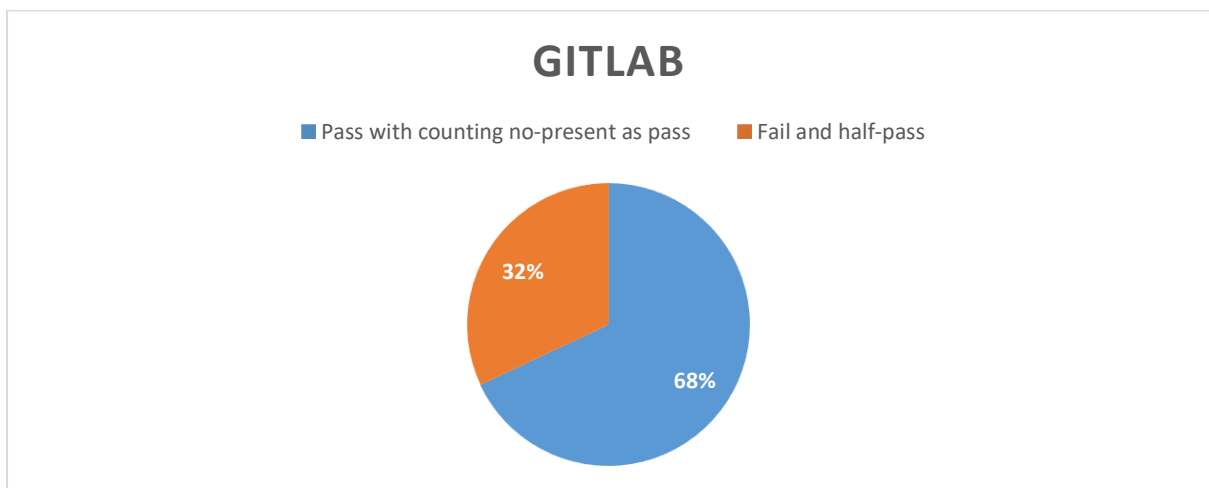


Figure 7: GitLab result counting no-present as pass

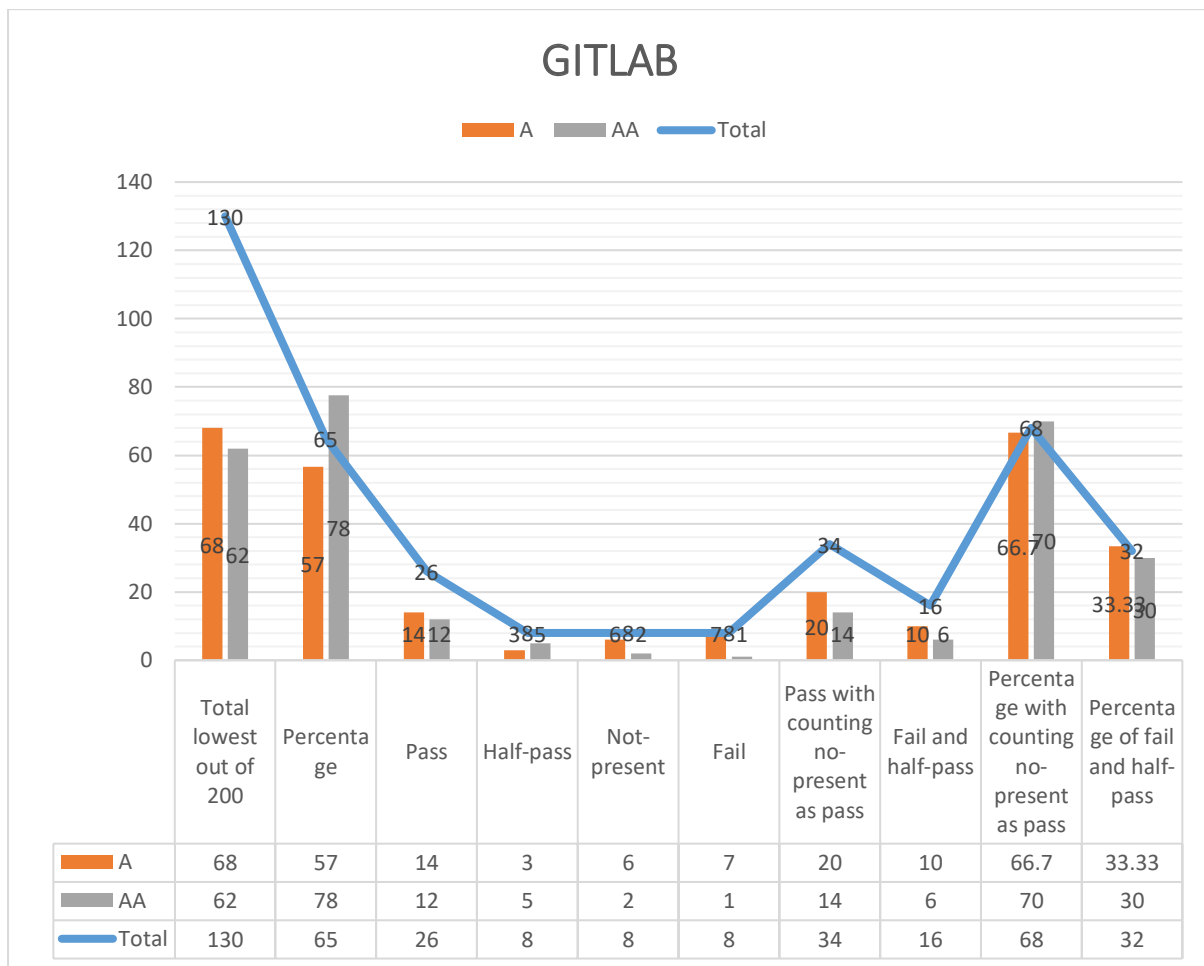


Figure 8: GitLab overall result

## 4.3 BITBUCKET

Bitbucket is similar to GitHub, used for code management, collaboration, and testing.

### 4.3.1 Evaluation Scope

**Scope of the website:** Web pages in <https://bitbucket.org/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.3.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies includes: HTML5, CSS, JavaScript, JQuery, Bootstrap, WordPress, ReCAPTCHA, Java, Node.js.

### 4.3.3 Representative Sample

Web pages in <https://bitbucket.org/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

### 4.3.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Homepage.

Link: <https://bitbucket.org/>

#### *Page 2*

Sign up page.

Link: <https://bitbucket.org/account/signup/>

#### *Page 3*

Login page.

Link: <https://bitbucket.org/account/signin/>

#### *Page 4*

BitBucket after login page. Test different functionality for creating a new repository.

Links: <https://bitbucket.org/dashboard/overview> and <https://bitbucket.org/repo/create>

#### *Page 5*

BitBucket after login page. Getting inside the created repository and testing different sections inside the repository, such as code, issues, pull requests, actions, projects, security, insights, and settings.

Links: <https://bitbucket.org/kakarsam/test/src/master/>

**Overall result:**

Table 4 shows the overall result for all the representative samples.

Success criteria	Pag e 1	Pag e 2	Pag e 3	Pag e 4	Pag e 5	Lowest	Total score	Confor mance level
1.1.1 Non-text content	3	4	4	1	1	3	13	A
1.2.1. Audio-only and Video-only	1	1	1	1	1	1	5	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	1	5	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	1	1	1	1	1	5	A
1.2.4. Captions (Live)	1	1	1	1	1	1	5	AA
1.2.5. Audio Description (Pre-recorded)	1	1	1	1	1	1	5	AA
1.3.1. Info and Relationships	4	4	4	2	2	2	16	A
1.3.2. Meaningful Sequence	4	0	4	0	0	0	8	A
1.3.3. Sensory Characteristics	4	4	4	4	4	4	20	A
1.3.4 Orientation	4	4	4	4	4	4	20	AA
1.3.5 Identify Input Purpose	4	4	4	4	4	4	20	AA
1.4.1 Use of Color	4	4	4	4	4	4	20	A
1.4.2 Audio Control	1	1	1	1	1	1	5	A



1.4.3 Contrast (Minimum)	3	2	4	3	3	2	15	AA
1.4.4 Resize text	4	4	4	4	4	4	20	AA
1.4.5 Images of Text	4	4	4	4	4	4	20	AA
1.4.10 Reflow	4	4	4	4	4	4	20	AA
1.4.11 Non-Text Contrast	3	2	3	3	3	2	14	AA
1.4.12 Text Spacing	4	4	4	4	4	4	20	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	20	AA
2.1.1. Keyboard	4	4	4	4	4	4	20	A
2.1.2. No Keyboard Trap	4	4	4	4	4	4	20	A
2.1.4. Character Key Shortcuts	1	1	1	1	1	1	5	A
2.2.1 Timing Adjustable	1	1	1	1	1	1	5	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	5	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	5	A
2.4.1 Bypass Blocks	0	0	0	0	0	0	0	A
2.4.2 Page Titled	4	4	4	4	4	4	20	A
2.4.3 Focus Order	4	4	4	4	4	4	20	A
2.4.4 Link Purpose (In Context)	3	3	3	2	4	2	15	A
2.4.5 Multiple Ways	4	4	4	4	4	4	20	AA

2.4.6 Headings and Labels	2	3	3	4	2	2	14	AA
2.4.7 Focus Visible	2	4	4	4	4	2	18	AA
2.5.1 Pointer Gestures	4	4	4	4	4	4	20	A
2.5.2 Pointer Cancellation	4	4	4	4	4	4	20	A
2.5.3 Label in Name	4	4	4	4	4	4	20	A
2.5.4 Motion Actuation	1	1	1	1	1	1	5	A
3.1.1 Language of Page	4	4	4	4	4	4	20	A
3.1.2 Language of Parts	1	1	1	1	1	1	5	AA
3.2.1 On Focus	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	AA
3.2.4 Consistent Identification	4	4	4	4	4	4	20	AA
3.3.1 Error Identification	4	4	4	4	4	4	20	A
3.3.2 Labels or Instructions	0	0	0	0	0	0	0	A
3.3.3 Error Suggestion	4	4	4	4	4	4	20	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	4	4	20	AA
4.1.1 Parsing	2	2	2	2	3	2	11	A
4.1.2 Name, Role, Value	3	4	4	4	3	3	18	A
4.1.3 Status Messages	1	4	4	1	1	4	11	AA

Table 4: Bitbucket results

### 4.3.5 Evaluation Findings

As per findings, BitBucket has a lowest score of 136 (73 A, 63 AA) out of 200, which is 68% (61% A, 79% AA), and total score of 713 out of 1000, which is 71.3%, as shown in Figure 11. The total number of pass criteria is 26 (13 A, 13 AA), 9 (5 A, 4 AA) half-pass, 12 (9 A, 3 AA) not-present, 3 (3 A, 0 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 9, pass percentage is 52% (43.3% A, 65% AA), 18% (16.6% A, 20% AA) half-pass, 24% (30% A, 15% AA) not-present, and 6% (10% A, 0% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of the pass is 38 (22 A, 16 AA) and fail as 12 (8 A, 4 AA). In percentage, as shown in Figure 10, the pass percentage is 76% (73.4% A, 80% AA) and fail as 24% (26.6% A, 20% AA).

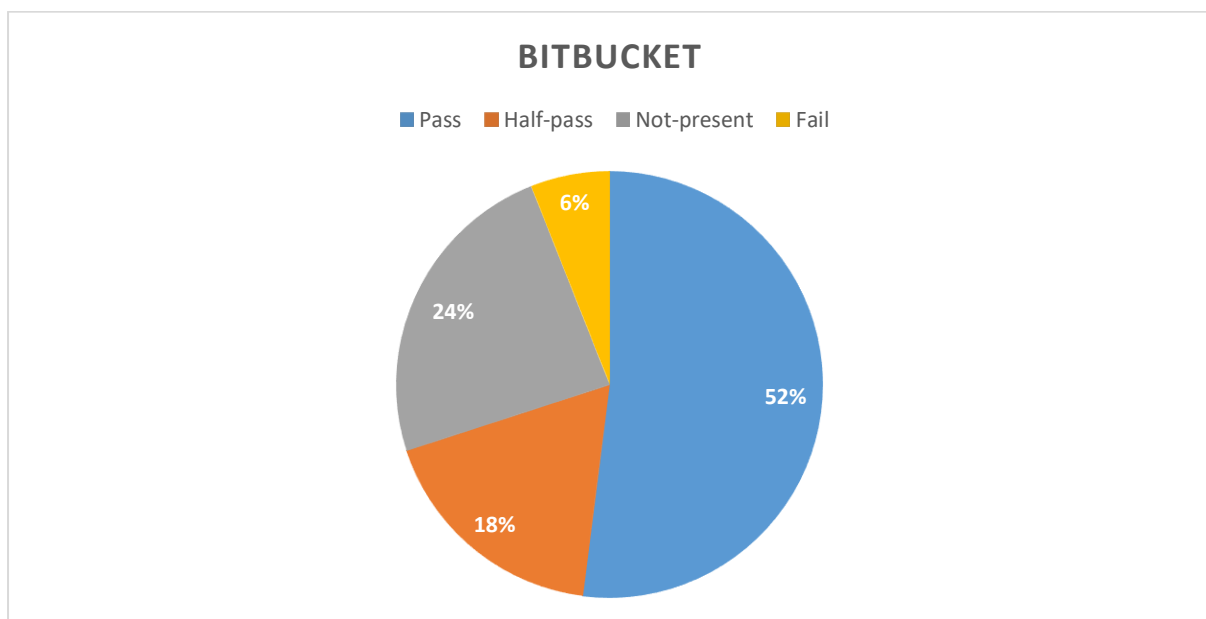


Figure 9: BitBucket lowest score

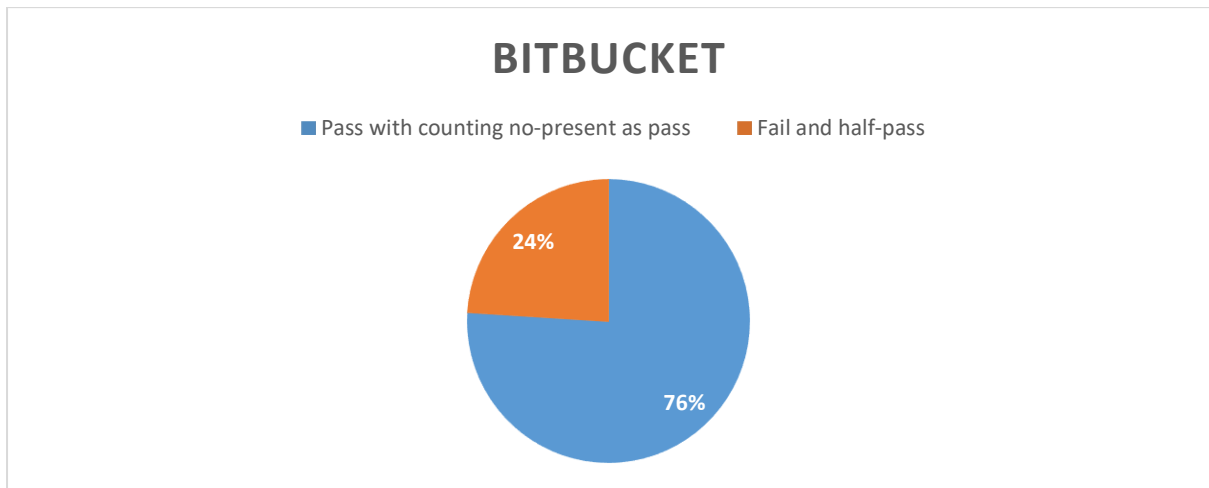


Figure 10: BitBucket result counting no-present as pass

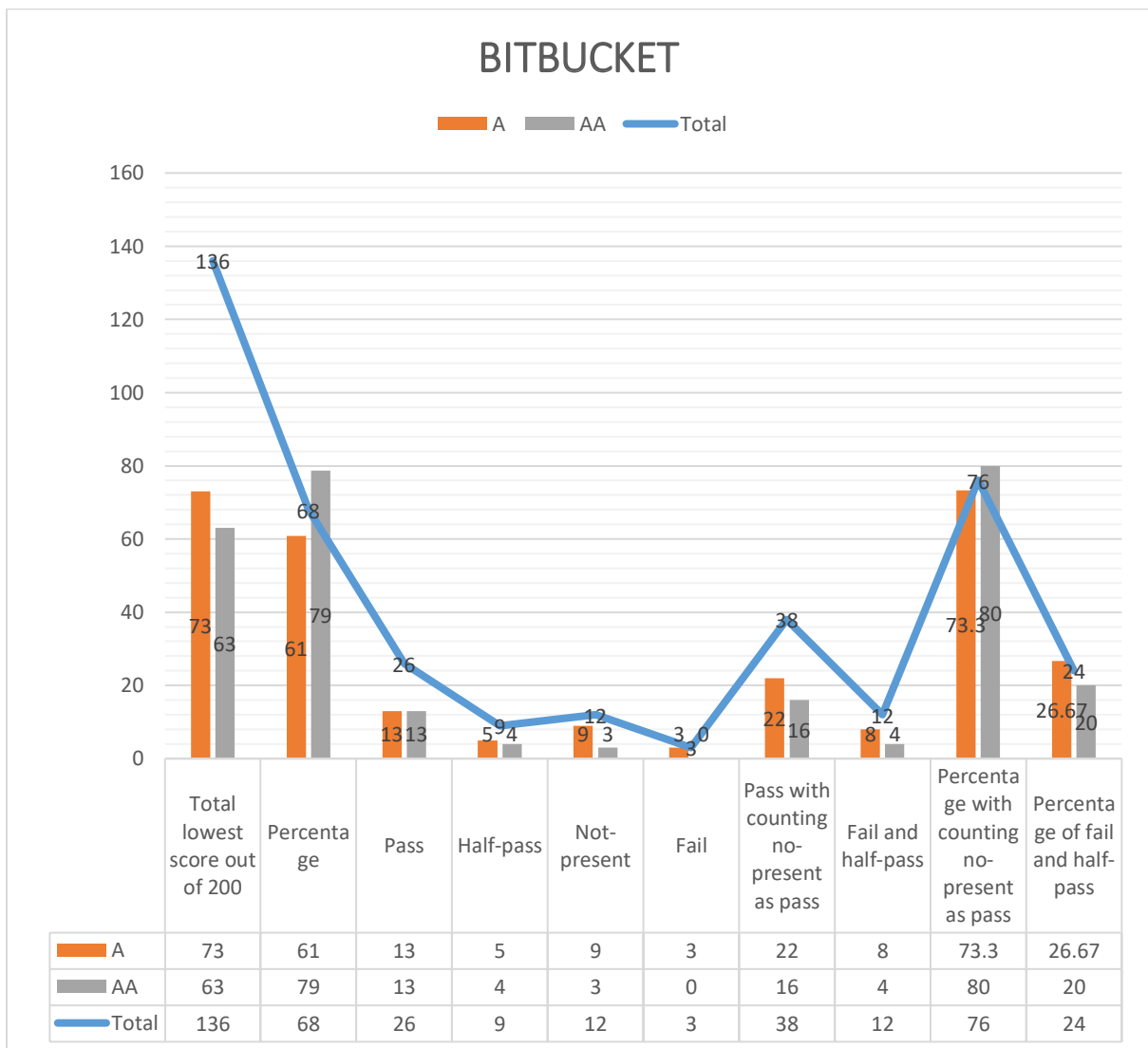


Figure 11: BitBucket overall result

## 4.4 SONARQUBE

SonarQube is a code analyzer and management tool which can either be installed on a computer or used online. It supports more than 25 programming languages.

### 4.4.1 Evaluation Scope

**Scope of the website:** Web pages in <https://sonarcloud.io>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.4.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies includes: HTML5, CSS, JavaScript, JQuery, WebPack, Lodash, React.

### 4.4.3 Representative Sample

Web pages in <https://sonarcloud.io/> from home to signing in pages and afterward. Sign in and sign up are omitted since it uses another website to log in, such as GitHub, Bitbucket, Gitlab, Azure DevOps.

List of all web pages: Samples are represented in the below section.

### 4.4.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Homepage.

Link: <https://sonarcloud.io/>

#### *Page 2*

Page 2 is the first page after signing in.

Links: <https://sonarcloud.io/projects> and <https://sonarcloud.io/projects/create>

### Page 3

Inside the project.

Link: <https://sonarcloud.io/dashboard>

### Overall result:

Table 5 shows the overall result for all the representative samples.

Success criteria	Page 1	Page 2	Page 3	Lowest	Total score	Confor mance level
1.1.1 Non-text content	4	4	4	4	12	A
1.2.1. Audio-only and Video-only	1	1	1	1	3	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	3	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	1	1	1	3	A
1.2.4. Captions (Live)	1	1	1	1	3	AA
1.2.5. Audio Description (Pre- recorded)	1	1	1	1	3	AA
1.3.1. Info and Relationships	4	2	3	2	9	A
1.3.2. Meaningful Sequence	4	4	4	4	12	A
1.3.3. Sensory Characteristics	4	4	4	4	12	A
1.3.4 Orientation	4	4	4	4	12	AA
1.3.5 Identify Input Purpose	4	4	4	4	12	AA

1.4.1 Use of Color	4	4	4	4	12	A
1.4.2 Audio Control	1	1	1	1	3	A
1.4.3 Contrast (Minimum)	2	2	2	2	6	AA
1.4.4 Resize text	4	4	4	4	12	AA
1.4.5 Images of Text	1	1	1	1	3	AA
1.4.10 Reflow	2	2	2	2	6	AA
1.4.11 Non-Text Contrast	1	1	1	1	3	AA
1.4.12 Text Spacing	4	2	2	2	8	AA
1.4.13 Content on Hover or Focus	4	4	4	4	12	AA
2.1.1. Keyboard	4	4	4	4	12	A
2.1.2. No Keyboard Trap	4	4	4	4	12	A
2.1.4. Character Key Shortcuts	1	4	4	4	9	A
2.2.1 Timing Adjustable	1	1	1	1	3	A
2.2.2 Pause, Stop, Hide	1	1	1	1	3	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	3	A
2.4.1 Bypass Blocks	0	4	4	0	8	A
2.4.2 Page Titled	4	4	4	4	12	A
2.4.3 Focus Order	4	4	4	4	12	A
2.4.4 Link Purpose (In Context)	3	4	4	3	11	A

2.4.5 Multiple Ways	4	4	4	4	12	AA
2.4.6 Headings and Labels	4	2	4	2	10	AA
2.4.7 Focus Visible	2	4	4	2	10	AA
2.5.1 Pointer Gestures	4	4	4	4	12	A
2.5.2 Pointer Cancellation	4	4	4	4	12	A
2.5.3 Label in Name	4	4	4	4	12	A
2.5.4 Motion Actuation	1	1	1	1	3	A
3.1.1 Language of Page	4	4	4	4	12	A
3.1.2 Language of Parts	1	1	1	1	3	AA
3.2.1 On Focus	4	4	4	4	12	A
3.2.2 On Input	4	4	4	4	12	A
3.2.2 On Input	4	4	4	4	12	AA
3.2.4 Consistent Identification	4	4	4	4	12	AA
3.3.1 Error Identification	4	4	4	4	12	A
3.3.2 Labels or Instructions	4	0	0	0	4	A
3.3.3 Error Suggestion	4	4	4	4	12	AA
3.3.4 Error Prevention (Legal, Financial, Data)	1	1	1	1	3	AA
4.1.1 Parsing	3	1	2	2	6	A
4.1.2 Name, Role, Value	3	1	1	3	5	A



4.1.3 Status Messages	1	4	1	4	6	AA
-----------------------	---	---	---	---	---	----

Table 5: SonarQube results

## 4.4.5 Evaluation Findings

As per findings, SonarQube has a lowest score of 134 (82 A, 52 AA) out of 200, which is 67% (68% A, 65% AA), and total score of 416 out of 600, which is 69.3%, as shown in Figure 14. The total number of pass criteria is 25 (16 A, 9 AA), 9 (4 A, 5 AA) half-pass, 14 (8 A, 6 AA) not-present, 2 (2 A, 0 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 12, pass percentage is 50% (53.3% A, 45% AA), 18% (13.3% A, 25% AA) half-pass, 28% (26.6% A, 30% AA) not-present, and 4% (6.6% A, 0% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of the pass is 39 (24 A, 15 AA) and fail as 11 (6 A, 5 AA). In percentage, as shown in Figure 13, pass percentage is 78% (80% A, 75% AA) and fail as 22% (20% A, 25% AA).

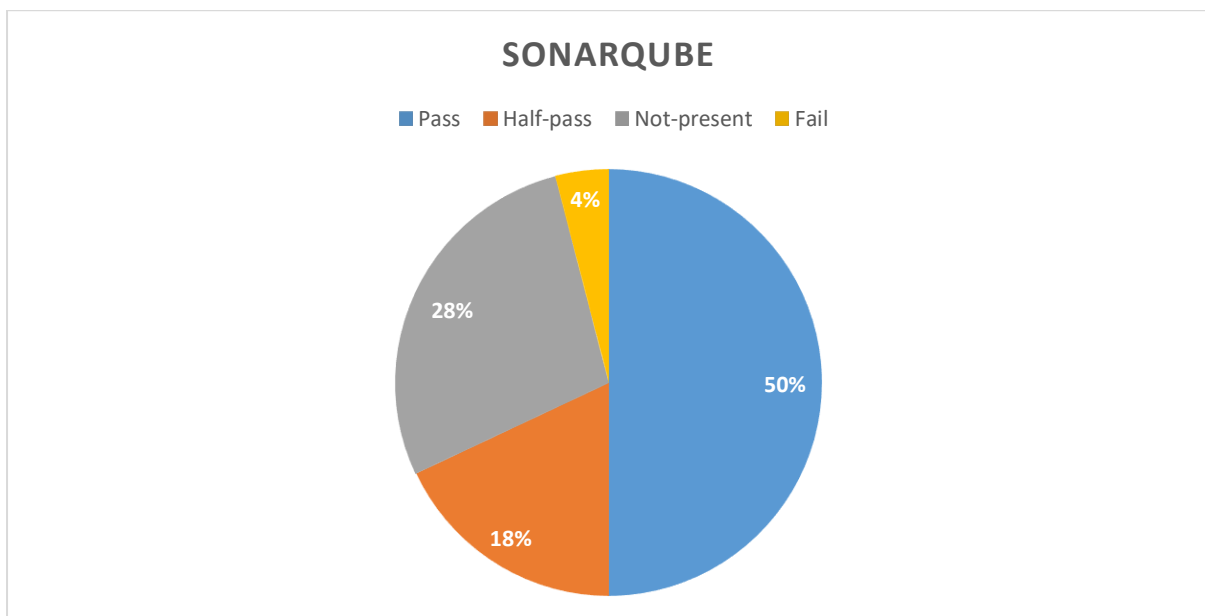


Figure 12: SonarQube lowest score

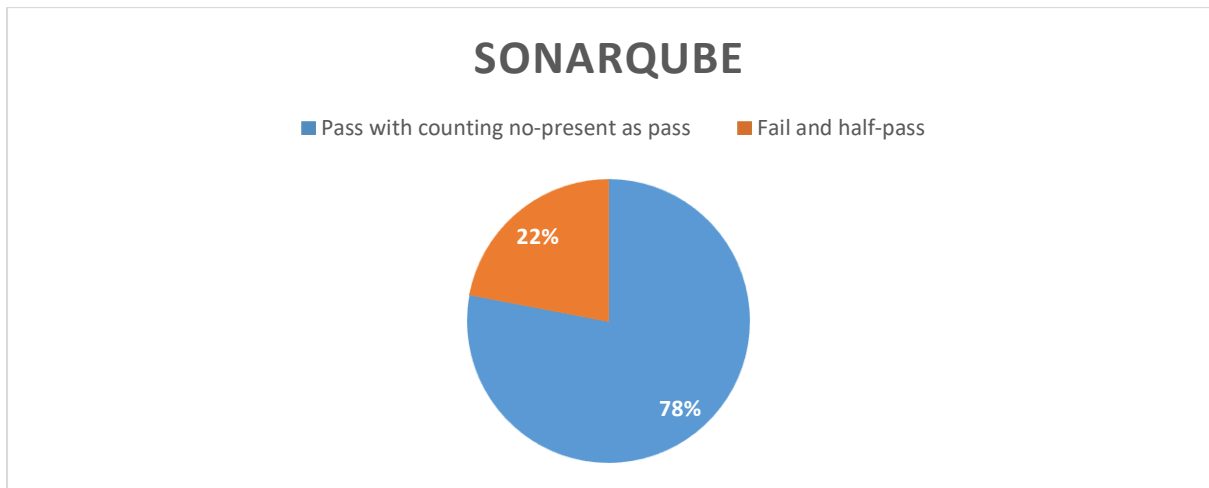


Figure 13: SonarQube result counting no-present as pass

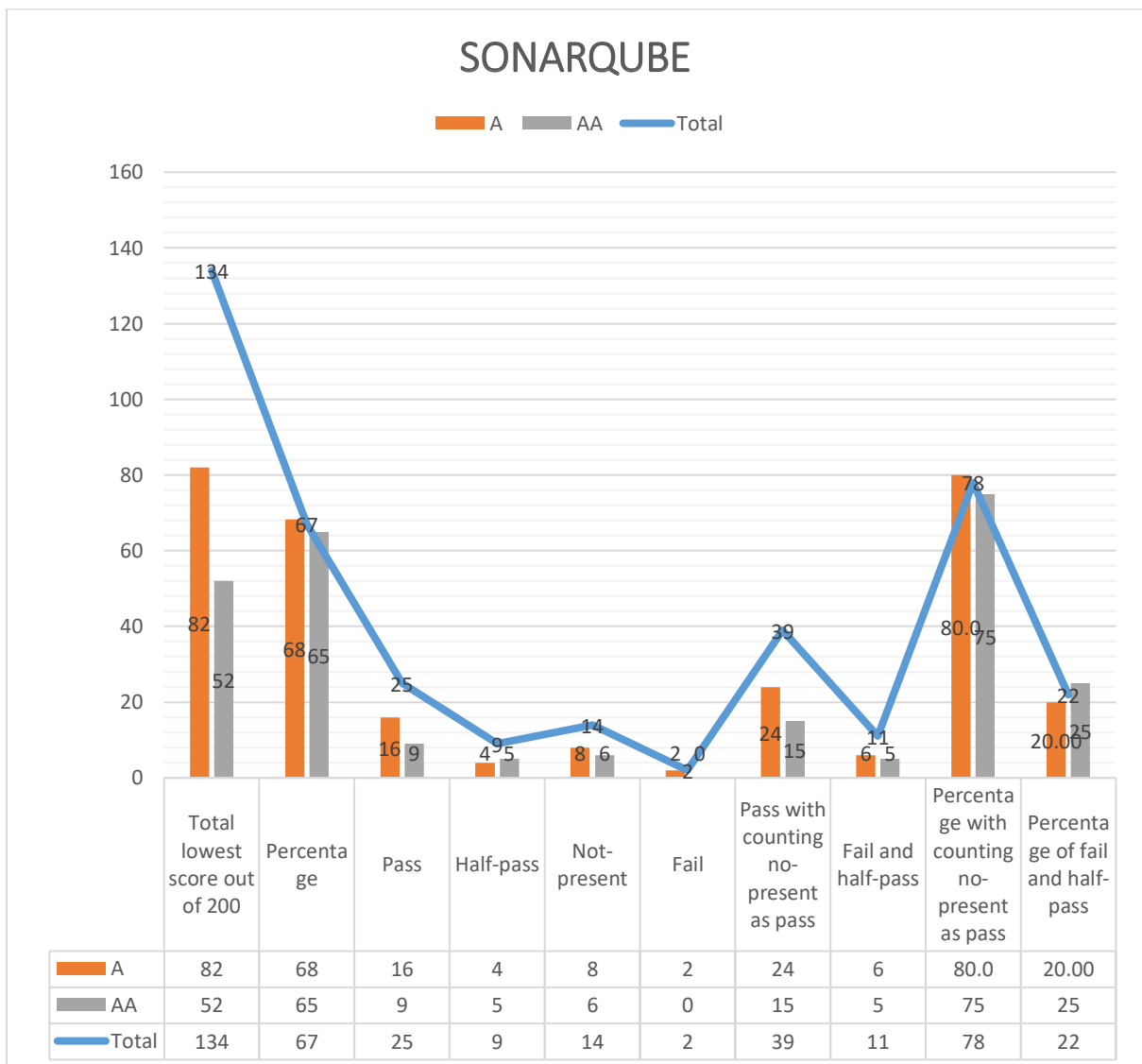


Figure 14: SonarQube overall result

## 4.5 TRAVIS CI

Travis CI is a continuous integration and deployment tool, can be integrated with GitHub, which supports various languages.

### 4.5.1 Evaluation Scope

**Scope of the website:** Web pages in <https://travis-ci.com/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.5.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies include HTML5, CSS, JavaScript, JQuery, Google Font API, Ember.js, Prism, Moments.js.

### 4.5.3 Representative Sample

Web pages in <https://travis-ci.com/> from home to signing in pages and afterward. No need to go to sign in and sign up pages since it uses GitHub.

List of all web pages: Samples are represented in the below section.

### 4.5.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Travis CI Home page.

Link: <https://travis-ci.com/>

#### *Page 2*

The first page after signing in.

Link: <https://travis-ci.com/dashboard>

### Page 3

Inside create a repository for functionality testing.

Link: <https://travis-ci.com/samadhiof/samadhiof>

### Overall result:

Table 6 shows the overall result for all the representative samples.

1.1.1 Non-text content	1	4	4	4	9	A
1.2.1. Audio-only and Video-only	1	1	1	1	3	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	3	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	1	1	1	3	A
1.2.4. Captions (Live)	1	1	1	1	3	AA
1.2.5. Audio Description (Pre-recorded)	1	1	1	1	3	AA
1.3.1. Info and Relationships	3	2	3	2	8	A
1.3.2. Meaningful Sequence	4	4	4	4	12	A
1.3.3. Sensory Characteristics	4	4	4	4	12	A
1.3.4 Orientation	4	4	4	4	12	AA
1.3.5 Identify Input Purpose	4	4	4	4	12	AA
1.4.1 Use of Color	4	4	4	4	12	A
1.4.2 Audio Control	1	1	1	1	3	A
1.4.3 Contrast (Minimum)	2	2	2	2	6	AA

1.4.4 Resize text	4	4	4	4	12	AA
1.4.5 Images of Text	4	4	4	4	12	AA
1.4.10 Reflow	4	4	4	4	12	AA
1.4.11 Non-Text Contrast	2	2	2	2	6	AA
1.4.12 Text Spacing	4	4	4	4	12	AA
1.4.13 Content on Hover or Focus	4	4	4	4	12	AA
2.1.1. Keyboard	4	4	4	4	12	A
2.1.2. No Keyboard Trap	4	4	4	4	12	A
2.1.4. Character Key Shortcuts	1	1	1	1	3	A
2.2.1 Timing Adjustable	1	1	1	1	3	A
2.2.2 Pause, Stop, Hide	1	1	1	1	3	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	3	A
2.4.1 Bypass Blocks	0	0	0	0	0	A
2.4.2 Page Titled	4	4	4	4	12	A
2.4.3 Focus Order	4	4	4	4	12	A
2.4.4 Link Purpose (In Context)	4	3	3	3	10	A
2.4.5 Multiple Ways	4	4	4	4	12	AA
2.4.6 Headings and Labels	3	3	3	3	9	AA
2.4.7 Focus Visible	3	4	4	3	11	AA

2.5.1 Pointer Gestures	4	4	4	4	12	A
2.5.2 Pointer Cancellation	4	4	4	4	12	A
2.5.3 Label in Name	4	4	4	4	12	A
2.5.4 Motion Actuation	1	1	1	1	3	A
3.1.1 Language of Page	0	0	0	0	0	A
3.1.2 Language of Parts	1	1	1	1	3	AA
3.2.1 On Focus	4	4	4	4	12	A
3.2.2 On Input	4	4	4	4	12	A
3.2.2 On Input	4	4	4	4	12	AA
3.2.4 Consistent Identification	4	4	4	4	12	AA
3.3.1 Error Identification	4	4	4	4	12	A
3.3.2 Labels or Instructions	4	4	0	0	8	A
3.3.3 Error Suggestion	4	4	4	4	12	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	12	AA
4.1.1 Parsing	4	3	2	2	9	A
4.1.2 Name, Role, Value	4	3	3	3	10	A
4.1.3 Status Messages	1	1	1	1	3	AA

Table 6: Travis CI results

## 4.5.5 Evaluation Findings

As per findings, Travis CI has a lowest score of 137 (75 A, 62 AA) out of 200, which is 68.5% (63% A, 78% AA), and total score of 425 out of 600, which is 70.8%, as shown in Figure 17.

The total number of pass criteria is 26 (14 A, 12 AA), 8 (4 A, 4 AA) half-pass, 13 (9 A, 4 AA) not-present, 3 (3 A, 0 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 15, pass percentage is 52% (46.6% A, 60% AA), 16% (13.3% A, 20% AA) half-pass, 26% (30% A, 20% AA) not-present, and 6% (10% A, 0% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of pass is 39 (23 A, 16 AA) and fail as 11 (7 A, 4 AA). In percentage, as shown in Figure 16, pass percentage is 78% (76.7% A, 80% AA) and fail as 22% (23.3% A, 20% AA).

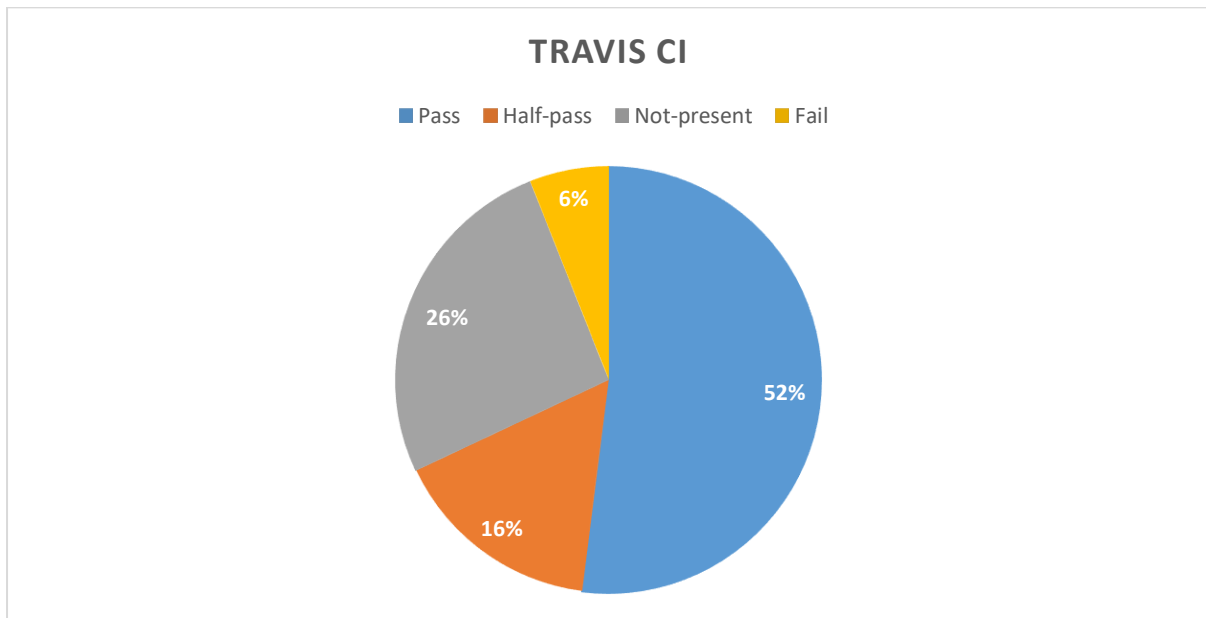


Figure 15: Travis CI lowest score

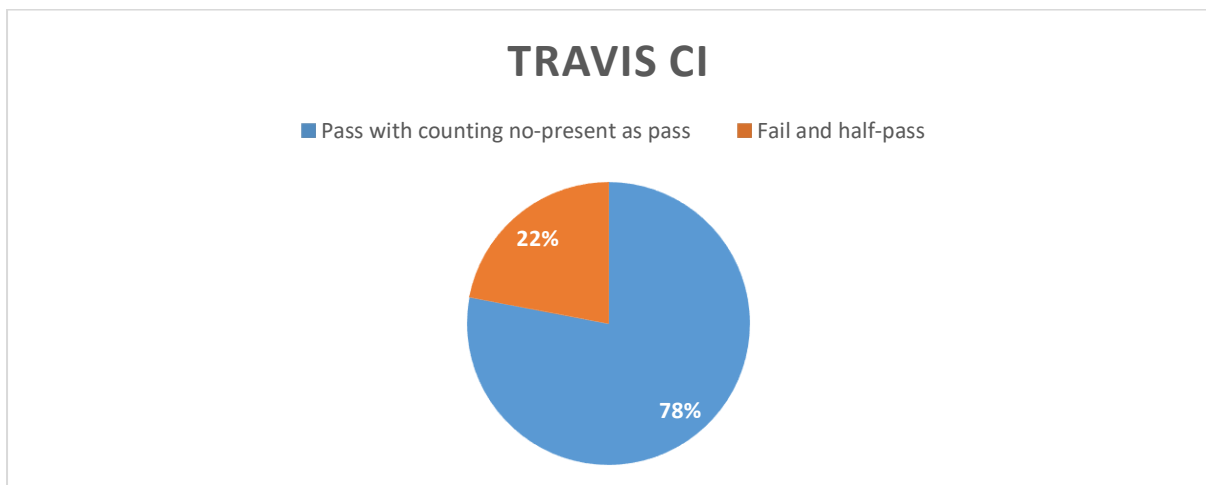


Figure 16: Travis CI result counting no-present as pass

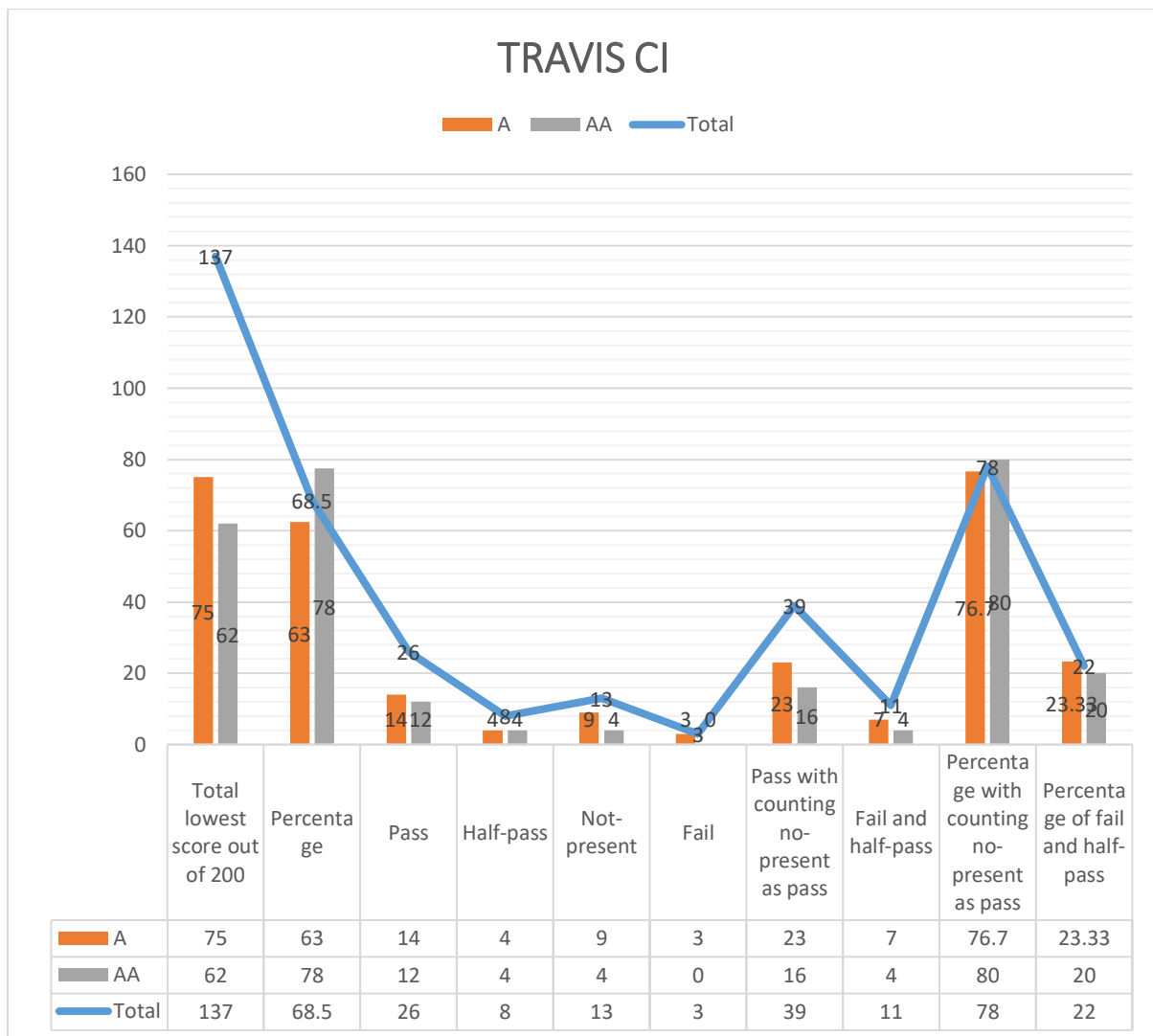


Figure 17: Travis CI overall result

## 4.6 SLACK

Slack is a cloud-based collaboration and communication tool. This tool can transfer files from Trello, GoogleDrive, DropBox, GitHub.

### 4.6.1 Evaluation Scope

**Scope of the website:** Web pages in <https://slack.com/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)



## 4.6.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies include HTML5, CSS, JavaScript, JQuery, Webpack, Pardot, Amazon Web services.

## 4.6.3 Representative Sample

Web pages in <https://slack.com/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

## 4.6.4 Testing the samples

The samples are evaluated according to conformance level AA.

### *Page 1*

Slack Home page.

Link: <https://slack.com/>

### *Page 2*

Sign up page. The first page clicks on to create a Slack workspace. Enter Email. Enter the received confirmation code.

Links: <https://slack.com/get-started#/create> and <https://slack.com/intl/en-no/create#confirmemail>

### *Page 3*

Login page. Enter workspace's **Slack URL**. Then username and password.

Links: <https://slack.com/signin> and <https://devops-yhw7199.slack.com/>

### *Page 4*

The first page after the signing-in page. Functionality testing.

Link: <https://app.slack.com/client/>

**Overall result:**

Table 7 shows the overall result for all the representative samples.

Success criteria	Pag e 1	Pag e 2	Pag e 3	Page 4	Lowest	Total score	Confor mance level
1.1.1 Non-text content	4	3	4	4	3	15	A
1.2.1. Audio-only and Video-only	0	1	1	1	0	3	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	4	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	0	1	1	1	0	3	A
1.2.4. Captions (Live)	1	1	1	1	1	4	AA
1.2.5. Audio Description (Pre-recorded)	0	1	1	1	0	3	AA
1.3.1. Info and Relationships	4	4	4	4	4	16	A
1.3.2. Meaningful Sequence	4	4	4	0	0	12	A
1.3.3. Sensory Characteristics	4	4	4	4	4	16	A
1.3.4 Orientation	4	4	4	4	4	16	AA
1.3.5 Identify Input Purpose	4	4	4	4	4	16	AA
1.4.1 Use of Color	4	4	4	4	4	16	A
1.4.2 Audio Control	4	1	1	1	4	7	A

1.4.3 Contrast (Minimum)	2	2	2	2	2	8	AA
1.4.4 Resize text	4	4	4	4	4	16	AA
1.4.5 Images of Text	4	4	4	4	4	16	AA
1.4.10 Reflow	4	4	4	4	4	16	AA
1.4.11 Non-Text Contrast	2	1	2	1	2	6	AA
1.4.12 Text Spacing	4	4	4	4	4	16	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	16	AA
2.1.1. Keyboard	4	4	4	4	4	16	A
2.1.2. No Keyboard Trap	4	4	4	0	0	12	A
2.1.4. Character Key Shortcuts	1	1	1	1	1	4	A
2.2.1 Timing Adjustable	1	1	1	1	1	4	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	4	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	4	A
2.4.1 Bypass Blocks	4	0	0	0	0	4	A
2.4.2 Page Titled	4	4	4	4	4	16	A
2.4.3 Focus Order	0	4	2	4	0	10	A
2.4.4 Link Purpose (In Context)	2	4	2	3	2	11	A
2.4.5 Multiple Ways	4	4	4	4	4	16	AA
2.4.6 Headings and Labels	4	2	0	0	0	6	AA

2.4.7 Focus Visible	4	4	4	4	4	16	AA
2.5.1 Pointer Gestures	4	4	4	4	4	16	A
2.5.2 Pointer Cancellation	4	4	4	4	4	16	A
2.5.3 Label in Name	4	4	4	4	4	16	A
2.5.4 Motion Actuation	1	1	1	1	1	4	A
3.1.1 Language of Page	4	4	4	4	4	16	A
3.1.2 Language of Parts	1	1	1	1	1	4	AA
3.2.1 On Focus	4	4	4	4	4	16	A
3.2.2 On Input	4	4	4	4	4	16	A
3.2.2 On Input	4	4	4	4	4	16	AA
3.2.4 Consistent Identification	4	4	4	4	4	16	AA
3.3.1 Error Identification	4	4	4	4	4	16	A
3.3.2 Labels or Instructions	4	0	0	0	0	4	A
3.3.3 Error Suggestion	4	4	4	4	4	16	AA
3.3.4 Error Prevention (Legal, Financial, Data)	1	1	1	1	1	4	AA
4.1.1 Parsing	1	2	4	3	2	10	A
4.1.2 Name, Role, Value	4	2	2	0	0	8	A
4.1.3 Status Messages	1	4	4	1	4	10	AA

Table 7: Slack results

### 4.6.5 Evaluation Findings

As per finding, Slack has a lowest score of 124 (65 A, 59 AA) out of 200, which is 62% (54% A, 74% AA), and total score of 552 out of 800, which is 69%, as shown in Figure 20. The total number of pass criteria is 26 (13 A, 13 AA), 5 (3 A, 2 AA) half-pass, 9 (6 A, 3 AA) not-present, 10 (8 A, 2 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 18, pass percentage is 52% (43,3% A, 65% AA), 10% (10% A, 10% AA) half-pass, 18% (20% A, 15% AA) not-present, and 20% (26.6% A, 10% AA) fail. However, if we count not-present criteria as pass and half-pass as failing, then the total number of the pass is 35 (19 A, 16 AA) and fail as 15 (11 A, 4 AA). In percentage, as shown in Figure 19, pass percentage is 70% (63.3% A, 80% AA) and fail as 30% (36.6% A, 20% AA).

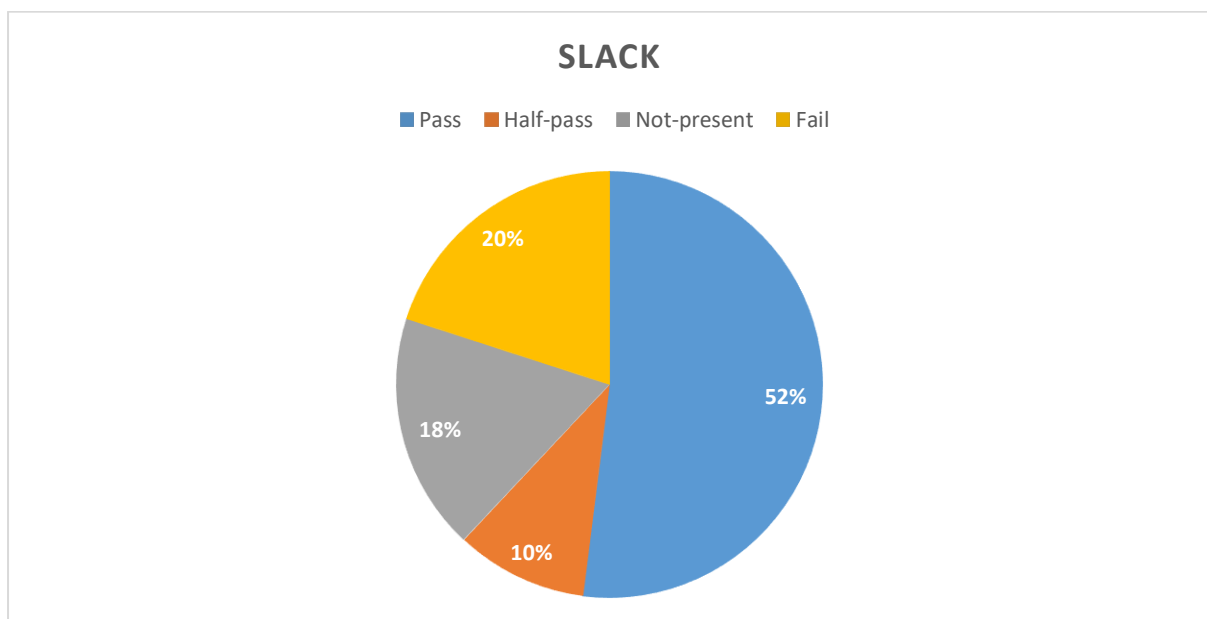


Figure 18: Slack lowest score

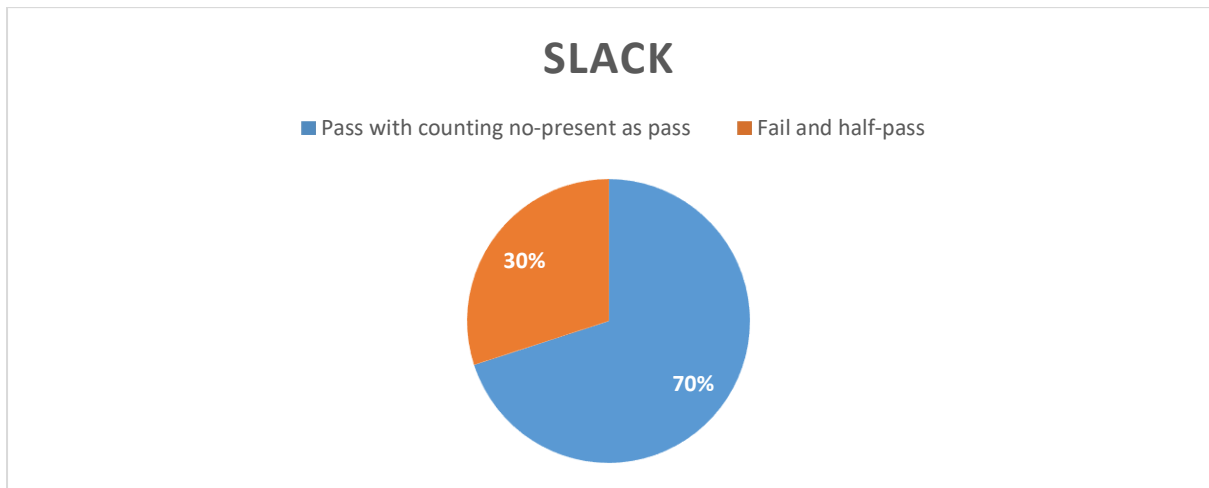


Figure 19: Slack result counting no-present as pass

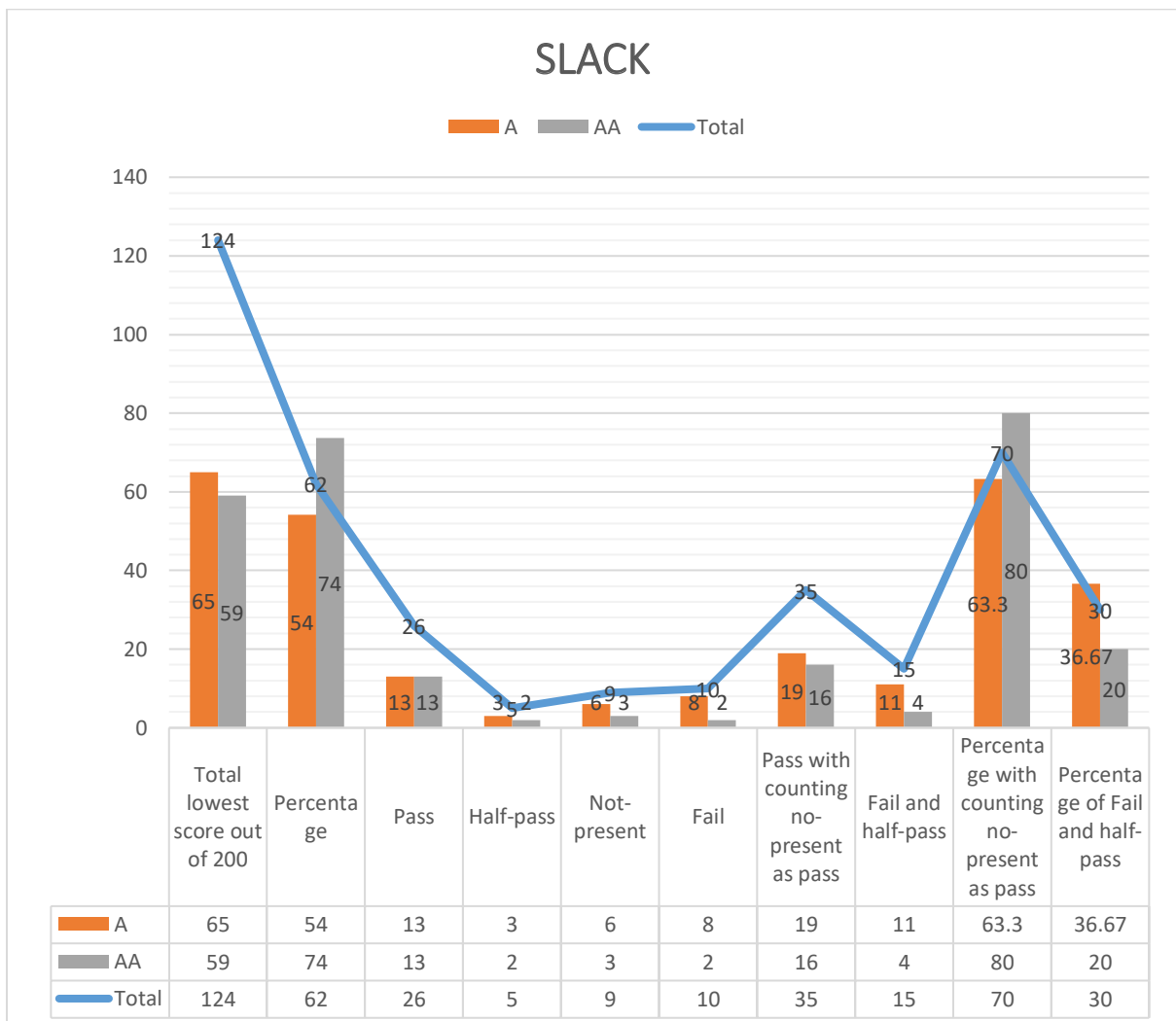


Figure 20: Slack overall result

## 4.7 TRELLO

Trello is a web-based project management and collaboration application.

### 4.7.1 Evaluation Scope

**Scope of the website:** Web pages in <https://trello.com/>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

### 4.7.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies includes: HTML5, CSS, JavaScript, JQuery, React, Scala, CloudFlare, Underscore.js, Google Font API.

### 4.7.3 Representative Sample

Web pages in <https://trello.com/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

### 4.7.4 Testing the samples

The samples are evaluated according to conformance level AA.

#### *Page 1*

Trello Home page.

Link: <https://trello.com/>

#### *Page 2*

Sign up page.

Link: <https://trello.com/signup>

#### *Page 3*

Login page.

Link: <https://trello.com/login>

#### *Page 4*

The first page after the signing-in page. Homepage for the user.

Link: <https://trello.com/skakar1/boards>

#### *Page 5*

Getting inside the created board and testing different functionality inside the board, such as creating/editing a new card, upload images to a card, add calendar functions to the board.

Link: <https://trello.com/b/PNixNfE4/devops>

#### *Overall result:*

Table 8 shows the overall result for all the representative samples.

Success criteria	Pag e 1	Pag e 2	Pag e 3	Pag e 4	Pag e 5	Lowest	Total score	Confo rmanc e level
1.1.1 Non-text content	2	1	2	2	1	2	8	A
1.2.1. Audio-only and Video-only	1	1	1	1	1	1	5	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	1	5	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	1	1	1	1	1	5	A
1.2.4. Captions (Live)	1	1	1	1	1	1	5	AA
1.2.5. Audio Description (Pre-recorded)	1	1	1	1	1	1	5	AA



1.3.1. Info and Relationships	4	4	4	0	0	0	12	A
1.3.2. Meaningful Sequence	4	4	4	4	4	4	20	A
1.3.3. Sensory Characteristics	4	4	4	4	4	4	20	A
1.3.4 Orientation	4	4	4	4	4	4	20	AA
1.3.5 Identify Input Purpose	4	4	4	4	4	4	20	AA
1.4.1 Use of Color	4	4	4	4	4	4	20	A
1.4.2 Audio Control	1	1	1	1	1	1	5	A
1.4.3 Contrast (Minimum)	2	2	2	2	3	2	11	AA
1.4.4 Resize text	4	4	4	4	4	4	20	AA
1.4.5 Images of Text	4	4	4	4	4	4	20	AA
1.4.10 Reflow	4	4	4	4	4	4	20	AA
1.4.11 Non-Text Contrast	1	1	1	1	1	1	5	AA
1.4.12 Text Spacing	4	4	4	4	4	4	20	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	20	AA
2.1.1. Keyboard	4	4	4	4	4	4	20	A
2.1.2. No Keyboard Trap	4	4	4	4	4	4	20	A
2.1.4. Character Key Shortcuts	1	1	1	4	4	4	11	A
2.2.1 Timing Adjustable	1	1	1	1	1	1	5	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	5	A

2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	5	A
2.4.1 Bypass Blocks	0	0	0	0	0	0	0	A
2.4.2 Page Titled	4	4	4	4	4	4	20	A
2.4.3 Focus Order	4	4	2	4	4	2	18	A
2.4.4 Link Purpose (In Context)	2	4	0	0	4	0	10	A
2.4.5 Multiple Ways	4	4	4	4	4	4	20	AA
2.4.6 Headings and Labels	0	0	0	0	0	0	0	AA
2.4.7 Focus Visible	0	2	2	2	4	0	10	AA
2.5.1 Pointer Gestures	4	4	4	4	4	4	20	A
2.5.2 Pointer Cancellation	4	4	4	4	4	4	20	A
2.5.3 Label in Name	4	4	4	4	4	4	20	A
2.5.4 Motion Actuation	1	1	1	1	1	1	5	A
3.1.1 Language of Page	0	0	0	4	4	0	8	A
3.1.2 Language of Parts	1	1	1	1	1	1	5	AA
3.2.1 On Focus	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	AA
3.2.4 Consistent Identification	4	4	4	4	4	4	20	AA

3.3.1 Error Identification	4	4	4	4	4	4	20	A
3.3.2 Labels or Instructions	0	0	0	0	0	0	0	A
3.3.3 Error Suggestion	4	4	4	4	4	4	20	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	4	4	20	AA
4.1.1 Parsing	2	2	2	0	0	0	6	A
4.1.2 Name, Role, Value	4	4	4	2	2	2	16	A
4.1.3 Status Messages	4	4	4	1	1	4	14	AA

Table 8: Trello results

## 4.7.5 Evaluation Findings

As per findings, Trello has a lowest score of 124 (66A, 58 AA) out of 200, which is 62% (55% A, 73% AA), and total score of 664 out of 1000, which is 66.4%, as shown in Figure 23. The total number of pass criteria is 26 (13 A, 13 AA), 4 (3 A, 1 AA) half-pass, 12 (8 A, 4 AA) not-present, 8 (6 A, 2 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 21, pass percentage is 52% (43,3% A, 65% AA), 8% (10% A, 5% AA) half-pass, 24% (26.6% A, 20% AA) not-present, and 16% (20% A, 10% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of the pass is 38 (21 A, 17 AA) and fail as 12 (9 A, 3 AA). In percentage, as shown in Figure 22, pass percentage is 76% (70% A, 85% AA) and fail as 24% (30% A, 15% AA).

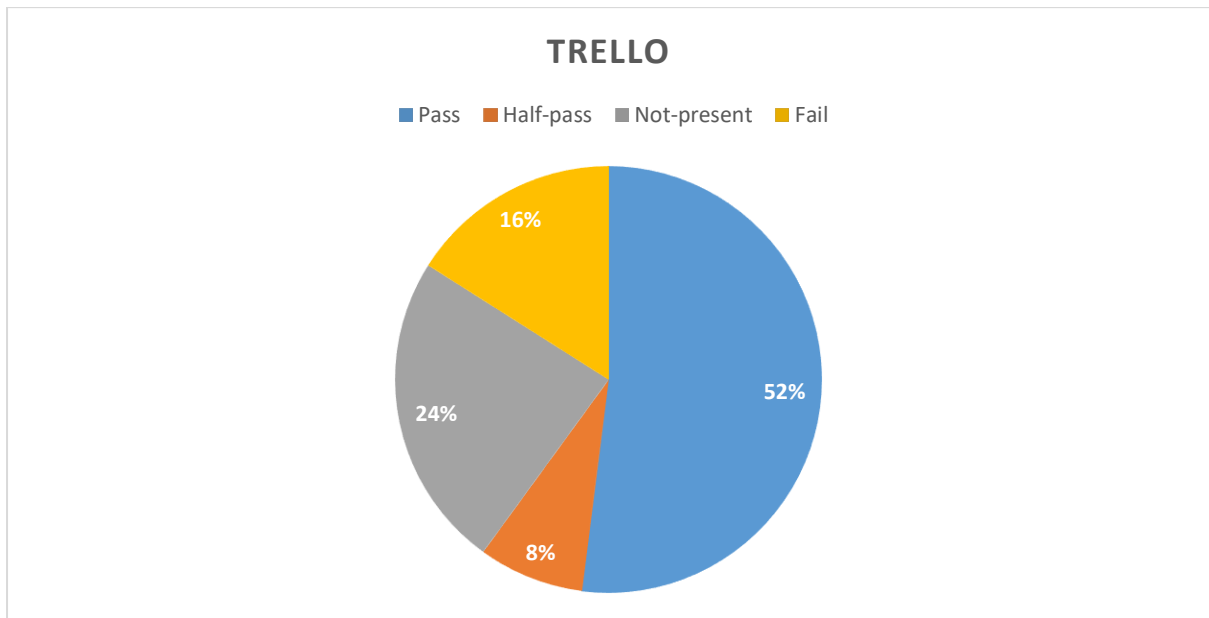


Figure 21: Trello lowest score

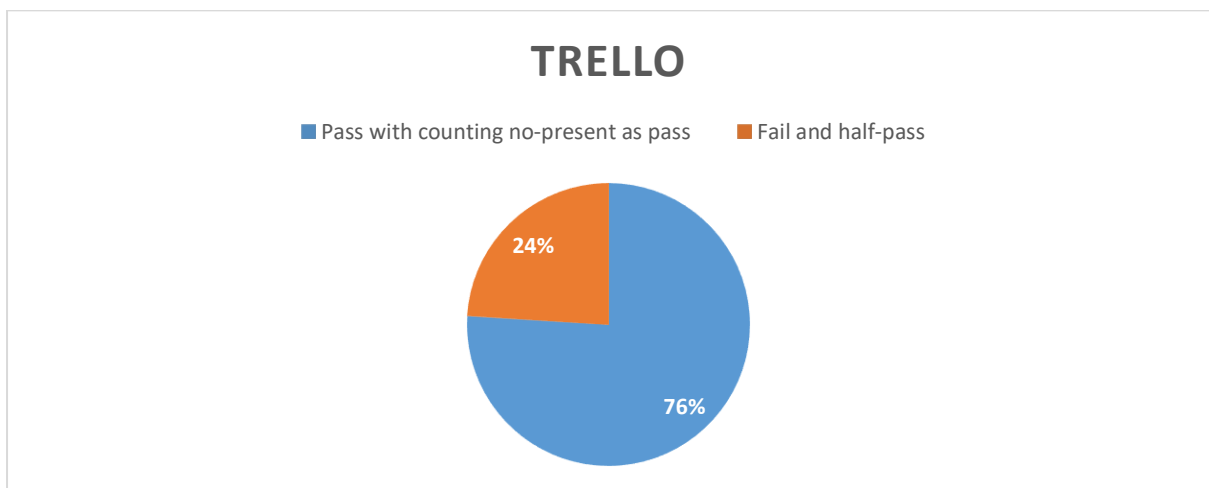


Figure 22: Trello result counting no-present as pass

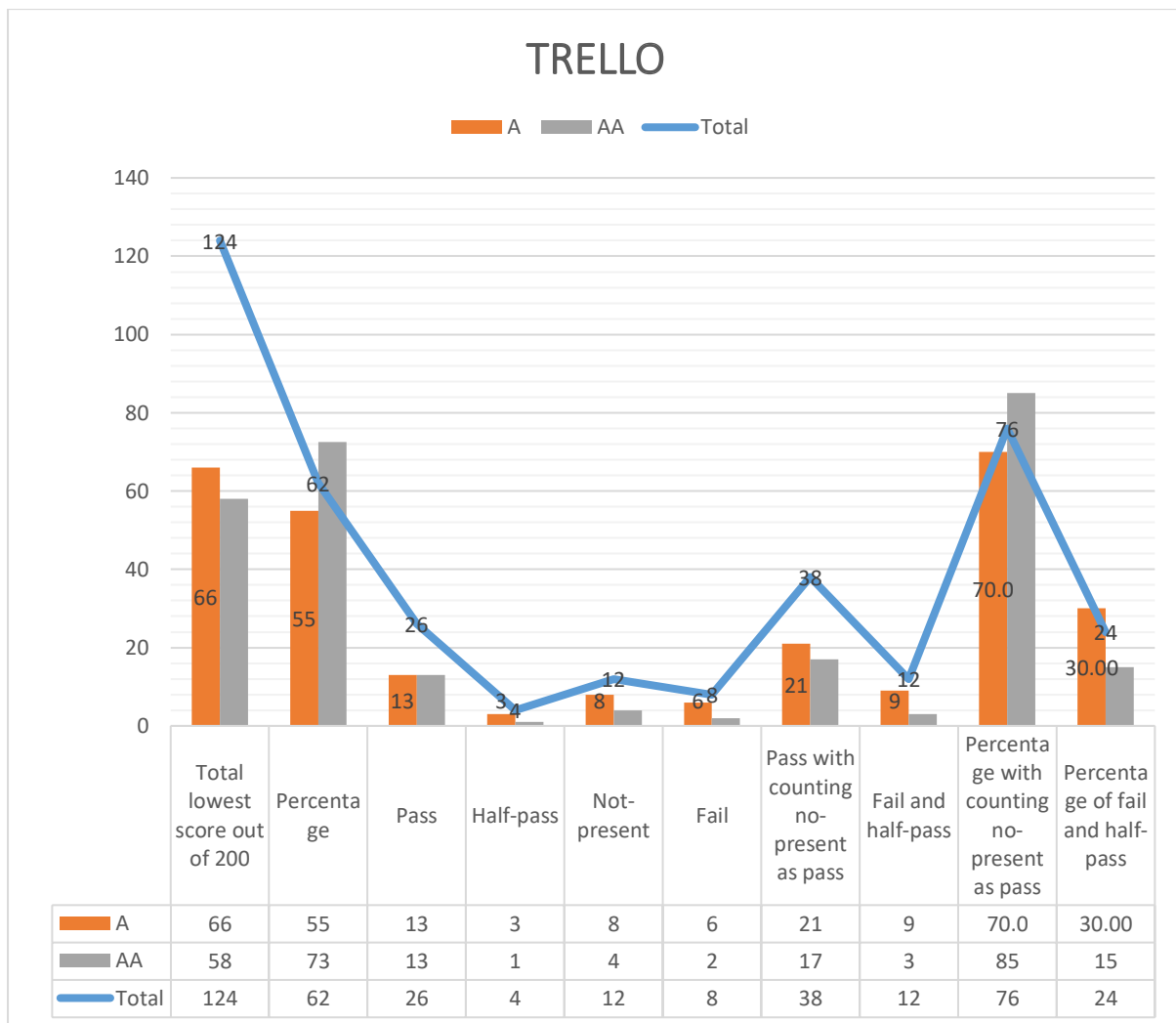


Figure 23: Trello overall result

## 4.8 HEROKU

Heroku is a cloud platform as a service (PaaS) supporting several programming languages.

### 4.8.1 Evaluation Scope

**Scope of the website:** Web pages in <https://www.heroku.com>

**Conformance target:** AA

**Accessibility support baseline:** Firefox 69.0.3 (64-bit)

## 4.8.2 Web technologies relied upon

Wappalyzer was used to find out web technologies relied upon on the website.

Technologies includes: HTML5, CSS, JavaScript, JQuery, List.js, Moments.js.

## 4.8.3 Representative Sample

Web pages in <https://www.heroku.com/> from home to signing in pages and afterward.

List of all web pages: Samples are represented in the below section.

## 4.8.4 Testing the samples

The samples are evaluated according to conformance level AA.

### *Page 1*

Heroku Home page.

Link: <https://www.heroku.com/>

### *Page 2*

Sign up page.

Link: <https://signup.heroku.com/>

### *Page 3*

Login page.

Link: <https://id.heroku.com/login>

### *Page 4*

Heroku after login page. Dashboard and create a new app.

Links: <https://dashboard.heroku.com/> and <https://dashboard.heroku.com/new-app>

### *Page 5*

Inside the created app. and testing different functions.

Link: <https://dashboard.heroku.com/apps/wcag-test>

### *Overall result:*

Table 9 shows the overall result for all the representative samples.

Success criteria	Pag e 1	Pag e 2	Pag e 3	Pag e 4	Pag e 5	Lowest	Total score	Confor mance level
1.1.1 Non-text content	3	4	4	4	4	3	13	A
1.2.1. Audio-only and Video-only	0	1	1	1	1	0	5	A
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	1	5	A
1.2.3. Audio Description or Media Alternative (Pre-recorded)	0	1	1	1	1	0	5	A
1.2.4. Captions (Live)	1	1	1	1	1	1	5	AA
1.2.5. Audio Description (Pre-recorded)	0	1	1	1	1	0	5	AA
1.3.1. Info and Relationships	1	4	3	1	1	3	16	A
1.3.2. Meaningful Sequence	4	4	4	0	4	0	8	A
1.3.3. Sensory Characteristics	4	4	4	4	4	4	20	A
1.3.4 Orientation	4	4	4	4	4	4	20	AA
1.3.5 Identify Input Purpose	3	4	4	4	4	3	20	AA
1.4.1 Use of Color	4	4	4	4	4	4	20	A
1.4.2 Audio Control	4	1	1	1	1	4	5	A

1.4.3 Contrast (Minimum)	2	3	3	3	3	2	15	AA
1.4.4 Resize text	4	4	4	4	4	4	20	AA
1.4.5 Images of Text	4	4	4	4	4	4	20	AA
1.4.10 Reflow	4	4	4	4	4	4	20	AA
1.4.11 Non-Text Contrast	2	3	4	3	3	2	14	AA
1.4.12 Text Spacing	4	4	4	4	4	4	20	AA
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	20	AA
2.1.1. Keyboard	4	4	4	4	4	4	20	A
2.1.2. No Keyboard Trap	4	4	4	4	4	4	20	A
2.1.4. Character Key Shortcuts	1	1	1	1	1	1	5	A
2.2.1 Timing Adjustable	1	1	1	1	1	1	5	A
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	5	A
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	5	A
2.4.1 Bypass Blocks	0	0	0	0	4	0	0	A
2.4.2 Page Titled	4	4	4	4	4	4	20	A
2.4.3 Focus Order	4	4	4	4	4	4	20	A
2.4.4 Link Purpose (In Context)	3	4	3	3	3	3	15	A
2.4.5 Multiple Ways	4	4	4	4	4	4	20	AA



2.4.6 Headings and Labels	1	3	3	2	2	2	14	AA
2.4.7 Focus Visible	2	4	4	2	2	2	18	AA
2.5.1 Pointer Gestures	4	4	4	4	4	4	20	A
2.5.2 Pointer Cancellation	4	4	4	4	4	4	20	A
2.5.3 Label in Name	4	4	4	4	4	4	20	A
2.5.4 Motion Actuation	1	1	1	1	1	1	5	A
3.1.1 Language of Page	4	0	4	4	4	0	20	A
3.1.2 Language of Parts	1	1	1	1	1	1	5	AA
3.2.1 On Focus	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	A
3.2.2 On Input	4	4	4	4	4	4	20	AA
3.2.4 Consistent Identification	4	4	4	4	4	4	20	AA
3.3.1 Error Identification	4	4	4	4	4	4	20	A
3.3.2 Labels or Instructions	0	4	4	0	0	0	0	A
3.3.3 Error Suggestion	4	4	4	4	4	4	20	AA
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	4	4	4	20	AA
4.1.1 Parsing	2	3	2	3	3	2	11	A
4.1.2 Name, Role, Value	2	4	3	3	3	2	18	A
4.1.3 Status Messages	1	4	4	4	1	4	11	AA

Table 9: Heroku results

### 4.8.5 Evaluation Findings

As per findings, Heroku has a lowest score of 132 (71 A, 61 AA) out of 200, which is 66% (59% A, 76% AA), and total score of 713 out of 1000, which is 71.3%, as shown in Figure 26. The total number of pass criteria is 25 (13 A, 12 AA), 10 (5 A, 5 AA) half-pass, 8 (6 A, 2 AA) not-present, 7 (6 A, 1 AA) fail out of all 50 success criteria. In percentage, as shown in Figure 24, pass percentage is 50% (43,3% A, 60% AA), 20% (16.7% A, 25% AA) half-pass, 16% (20% A, 10% AA) not-present, and 14% (20% A, 5% AA) fail. However, if we count not-present criteria as pass and half-pass as failing, then the total number of the pass is 33 (19 A, 14 AA) and fail as 17 (11 A, 6 AA). In percentage, as shown in Figure 25, pass percentage is 66% (63.3% A, 70% AA) and fail as 34% (36.6% A, 30% AA).

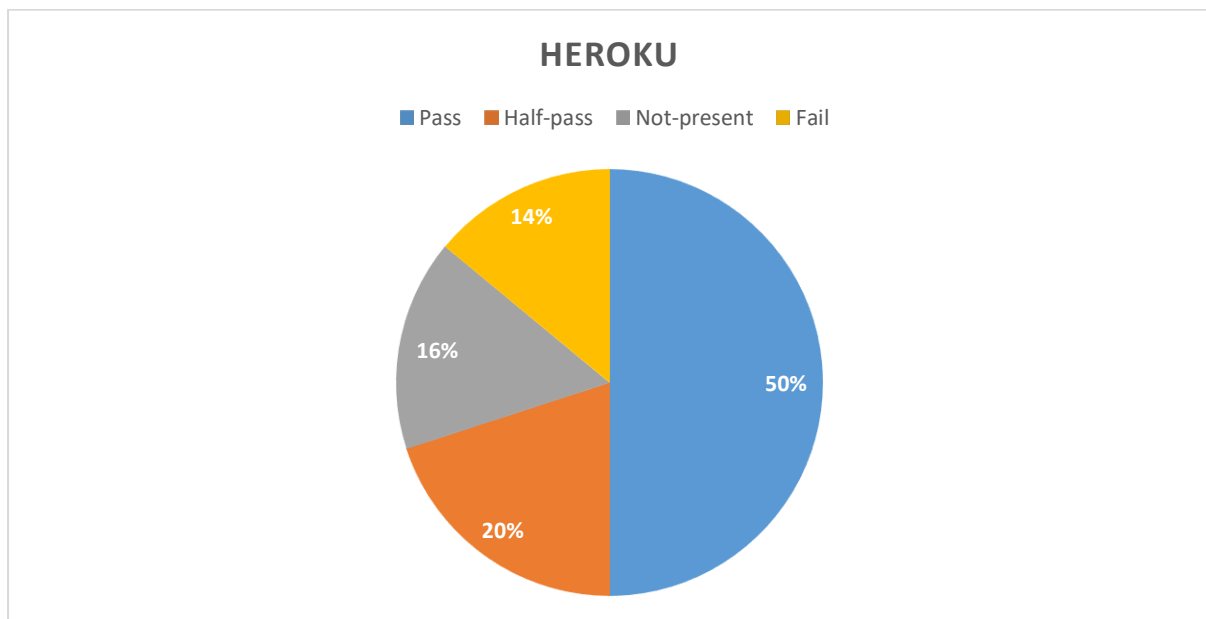


Figure 24: Heroku lowest score

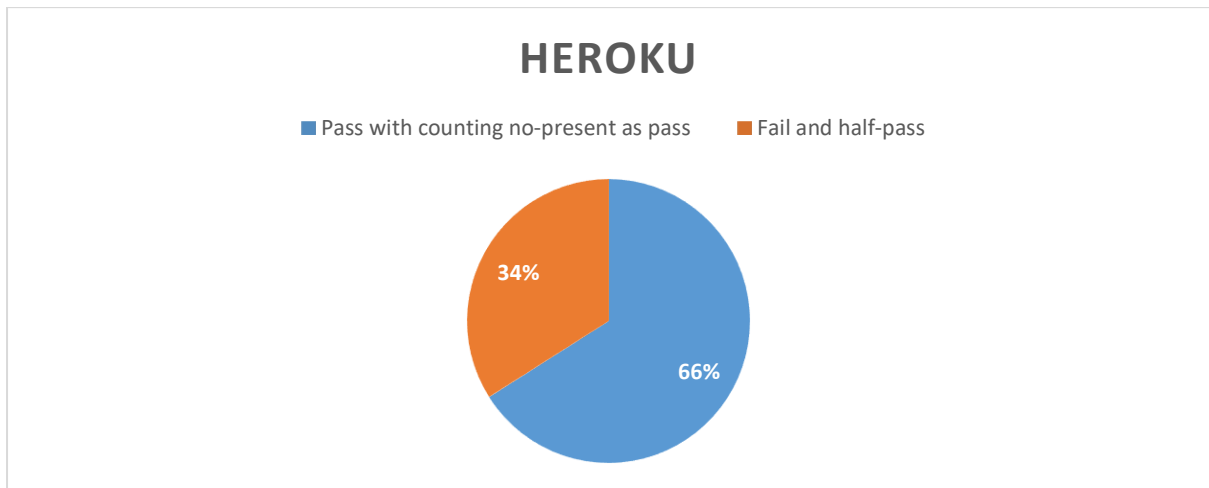


Figure 25: Heroku result counting no-present as pass

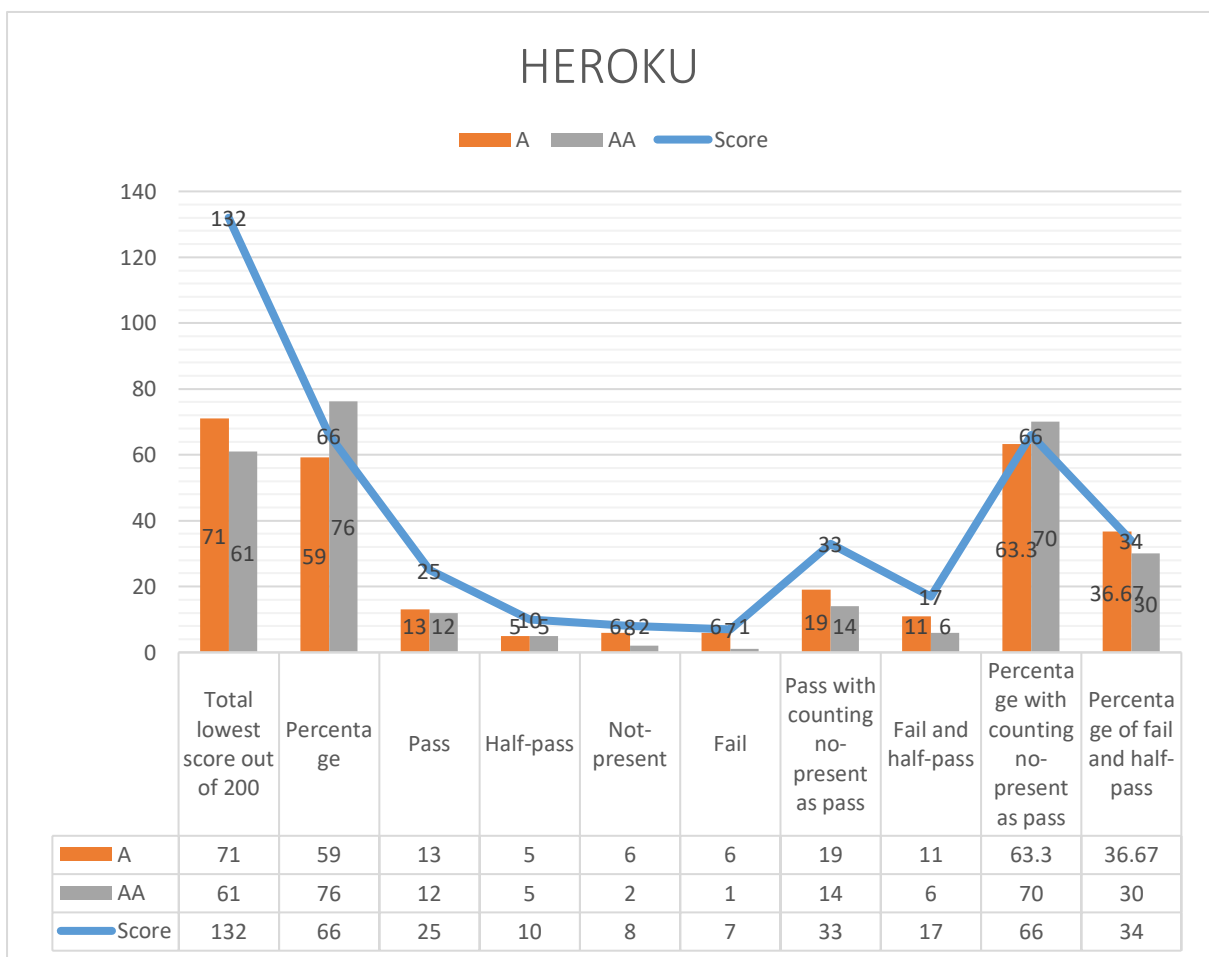


Figure 26: Heroku overall result

## 5 DISCUSSION

---

As per the overall result for all DevOps tools shown in Table 10, the average lowest score is 130.4 out of 200, which is 65.2%. The average pass criteria are 25, 7 half-pass, 11 not-present, and 6 fail out of all 50 success criteria. In percentage, as shown in Figure 28, pass percentage is 51%, 15% half-pass, 22% not-present, and 12% fail. However, if we count not-present criteria as pass and half-pass as fail, then the average of the pass score is 36.6 and fail as 13.4. In percentage, 73% pass and 27% fail, as shown in Figure 29. As per the result shown in Figure 27, there is a variation of 6.5% in the score for all DevOps tools. Travis CI has the highest lowest score percentage of 68.5, and Slack and Trello with the lowest 62. The average total score percentage is 70.4 for all DevOps. Trello has the lowest total score by 66.4%, and the highest is by GitHub 74.7%. There is a 5.2% difference while calculating the average percentage by lowest score and total score for all DevOps.

The pass percentage for all DevOps tools is almost the same, between 50-52%. The half-pass percentage varies from 8-20%, with Trello having the lowest and Heroku the highest half-pass percentage. The not-present percentage varies from 16-28%, with Gitlab and Heroku having the lowest and SonarQube the highest not-present percentage. Fail percentage varies from 4-20%, with SonarQube having the lowest and Slack the highest fail percentage. The pass percentage with counting no-present as pass varies from 66-78%, with Heroku having the lowest and Travis CI the highest. The fail percentage with counting no-present as pass and half-pass as fail varies from 22-34%, with SonarQube and Travis CI having the lowest and Heroku the highest. The high variation in percentage for half-pass, not-present, and fail values depends on the DevOps tools.

Tools	GitHub	GitLab	Bitbucket	SonarQube	Travis CI	Slack	Trello	Heroku	Average
Lowest score out of 200	126	130	136	134	137	<b>124</b>	<b>124</b>	132	130.38
Percentage	63	65	68	67	68.5	<b>62</b>	<b>62</b>	66	65.188

Pass	<b>25</b>	<b>26</b>	<b>26</b>	<b>25</b>	<b>26</b>	<b>26</b>	<b>26</b>	<b>25</b>	25.625
Pass %	<b>50</b>	<b>52</b>	<b>52</b>	<b>50</b>	<b>52</b>	<b>52</b>	<b>52</b>	<b>50</b>	51.25
Half-pass	6	8	9	9	8	5	<b>4</b>	<b>10</b>	7.375
Half-pass %	12	16	18	18	16	10	<b>8</b>	<b>20</b>	14.75
Not-present	12	<b>8</b>	12	<b>14</b>	13	9	12	<b>8</b>	11
Not-present %	24	<b>16</b>	24	<b>28</b>	26	18	24	<b>16</b>	22
Fail	7	8	3	<b>2</b>	3	<b>10</b>	8	7	6
Fail %	14	16	6	<b>4</b>	6	<b>20</b>	16	14	12
Pass with counting no-present as pass	37	34	38	39	<b>39</b>	35	38	<b>33</b>	36.625
Pass with counting no-present as pass %	74	68	76	78	<b>78</b>	70	76	<b>66</b>	73.25
Fail and half-pass	13	16	12	<b>11</b>	<b>11</b>	15	12	<b>17</b>	13.375
Fail and half-pass %	26	32	24	<b>22</b>	<b>22</b>	30	24	<b>34</b>	26.75
Total score percentage	<b>74.7</b>	70.7	71.3	69.3	70.8	69	<b>66.4</b>	71.3	70.438

Table 10: Overall DevOps tools results

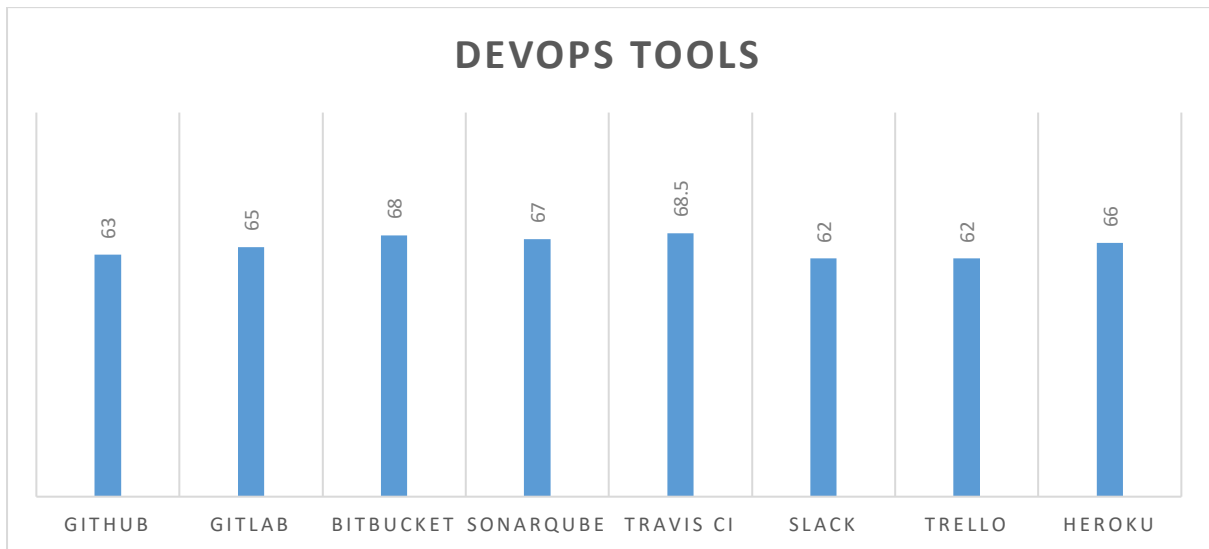


Figure 27: Overall DevOps tools lowest score percentage

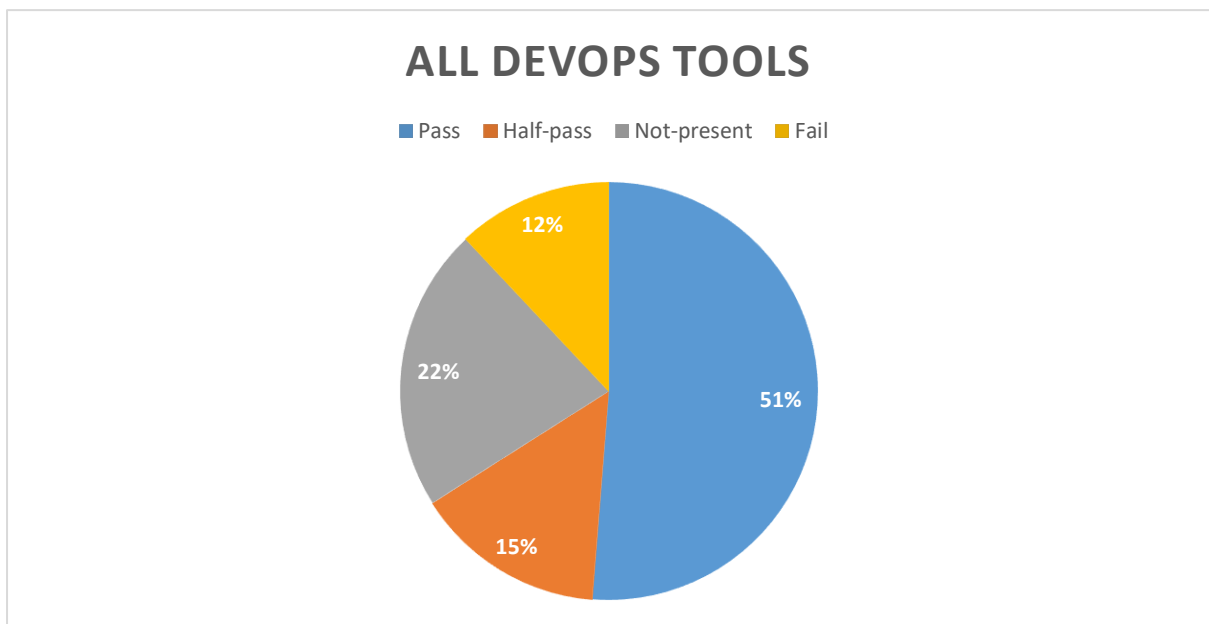


Figure 28: Overall DevOps tools lowest score

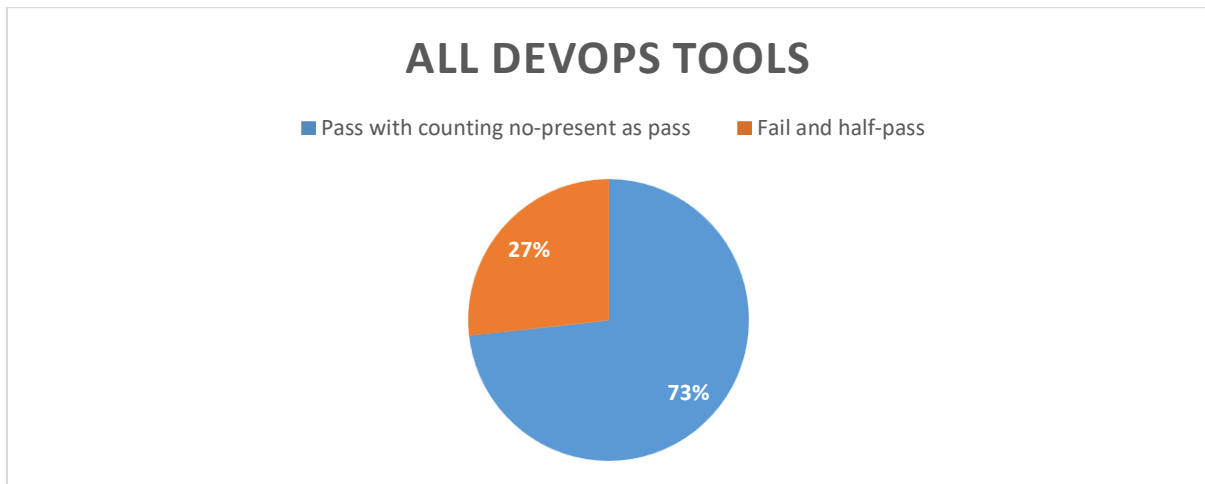


Figure 29: Overall DevOps tools counting no-present as pass

Another alternative for calculating the result for all DevOps tools is “aggregate lowest score.”. Aggregate lowest score means getting the lowest score of each success criterion for each DevOps tool and then getting the lowest score for all DevOps tools, as shown in Table 11. The total lowest score for all DevOps is 93 (46 A, 47 AA) out of 200, which is 46.5% (38 A%, 59 AA%). The total number of pass criteria is 19 (10 A, 9 AA), half-pass 4 (0 A, 4 AA), not-present 8 (6 A, 2 AA), and fail 19 (14 A, 5 AA) out of all 50 success criteria. In percentage, as shown in Figure 30, 38% (33.3% A, 45% AA) pass, 8% (0% A, 20% AA) half-pass, 16% (20% A, 10% AA) not-present, and 38% (46.6% A, 25% AA) fail. However, if we count not-present criteria as pass and half-pass as fail, then the total number of the pass is 27 (16 A, 11 AA) and fail as 23 (14 A, 9 AA). In percentage, as shown in Figure 31, 54% (53.3% A, 55% AA) pass and 46% (46.6% A, 45% AA) fail. The total score by all DevOps tools is 1043 out of 1600, which is 65.1%. In addition, if we calculate not-present as pass, then the total score by all DevOps tools is 1307 out of 1600, which is 81.6%. There is an 8% variance comparing the lowest score and lowest score, counting not-present as pass, and 16.5% variation comparing total score and total score not-present as pass.

As shown in Figure 32, comparing conformance level AA and level A, the lowest percentage for conformance level AA is higher than conformance level A, 59% to 38%. The pass percentage for level AA is higher than level A, 45% AA to 33%, and the fail percentage of level AA is lower than level A, 25% to 46%. However, if we count not-present as pass and half-pass as fail, then the pass and fail percentage between conformance level AA and level

A becomes somewhat similar. The pass percentage of level AA is 55% to 53.3 % of level A, and the fail percentage for level AA is 45% to 46.6% of level A.

Success criteria	Git Hu b	GitL ab	BitB ucke t	Sona rQub e	Trav is Cl	Sla ck	Tre llo	Her oku	Lo we st res ult	Sum mar y	Confo rman ce level	Tot al sco re
1.1.1 Non-text content	3	0	3	4	4	3	2	3	0	Fail	A	22
1.2.1. Audio-only and Video-only	1	0	1	1	1	0	1	0	0	Fail	A	5
1.2.2. Captions (Pre-recorded audio)	1	1	1	1	1	1	1	1	1	Not-present	A	8
1.2.3. Audio Description or Media Alternative (Pre-recorded)	1	0	1	1	1	0	1	0	0	Fail	A	5
1.2.4. Captions (Live)	1	1	1	1	1	1	1	1	1	Not-present	AA	8
1.2.5. Audio Description (Pre-recorded)	1	0	1	1	1	0	1	0	0	Fail	AA	5
1.3.1. Info and Relationships	0	0	2	2	2	4	0	3	0	Fail	A	13
1.3.2. Meaningful Sequence	0	4	0	4	4	0	4	0	0	Fail	A	16



1.3.3. Sensory Characteristics	4	4	4	4	4	4	4	4	4	Pass	A	32
1.3.4 Orientation	4	4	4	4	4	4	4	4	4	Pass	AA	32
1.3.5 Identify Input Purpose	4	3	4	4	4	4	4	3	3	Half-pass	AA	30
1.4.1 Use of Color	4	4	4	4	4	4	4	4	4	Pass	A	32
1.4.2 Audio Control	1	4	1	1	1	4	1	4	1	Not-present	A	17
1.4.3 Contrast (Minimum)	2	2	2	2	2	2	2	2	2	Half-pass	AA	16
1.4.4 Resize text	4	4	4	4	4	4	4	4	4	Pass	AA	32
1.4.5 Images of Text	4	4	4	1	4	4	4	4	4	Pass	AA	29
1.4.10 Reflow	0	4	4	2	4	4	4	4	0	Fail	AA	26
1.4.11 Non-Text Contrast	2	2	2	1	2	2	1	2	2	Half-pass	AA	14
1.4.12 Text Spacing	4	4	4	2	4	4	4	4	2	Half-pass	AA	30
1.4.13 Content on Hover or Focus	4	4	4	4	4	4	4	4	4	Pass	AA	32
2.1.1. Keyboard	0	4	4	4	4	4	4	4	0	Fail	A	28
2.1.2. No Keyboard Trap	0	4	4	4	4	0	4	4	0	Fail	A	24

2.1.4. Character Key Shortcuts	1	1	1	4	1	1	4	1	4	Pass	A	14
2.2.1 Timing Adjustable	1	1	1	1	1	1	1	1	1	Not-present	A	8
2.2.2 Pause, Stop, Hide	1	1	1	1	1	1	1	1	1	Not-present	A	8
2.3.1 Three Flashes or Below Threshold	1	1	1	1	1	1	1	1	1	Not-present	A	8
2.4.1 Bypass Blocks	4	0	0	0	0	0	0	0	0	Fail	A	4
2.4.2 Page Titled	4	4	4	4	4	4	4	4	4	Pass	A	32
2.4.3 Focus Order	0	4	4	4	4	0	2	4	0	Fail	A	22
2.4.4 Link Purpose (In Context)	2	2	2	3	3	2	0	3	0	Fail	A	17
2.4.5 Multiple Ways	4	4	4	4	4	4	4	4	4	Pass	AA	32
2.4.6 Headings and Labels	3	3	2	2	3	0	0	2	0	Fail	AA	15
2.4.7 Focus Visible	4	2	2	2	3	4	0	2	0	Fail	AA	19
2.5.1 Pointer Gestures	4	4	4	4	4	4	4	4	4	Pass	A	32
2.5.2 Pointer Cancellation	4	4	4	4	4	4	4	4	4	Pass	A	32
2.5.3 Label in Name	4	4	4	4	4	4	4	4	4	Pass	A	32

2.5.4 Motion Actuation	1	1	1	1	1	1	1	1	1	Not-present	A	8
3.1.1 Language of Page	4	0	4	4	0	4	0	0	0	Fail	A	16
3.1.2 Language of Parts	1	1	1	1	1	1	1	1	1	Not-present	AA	8
3.2.1 On Focus	4	4	4	4	4	4	4	4	4	Pass	A	32
3.2.2 On Input	4	4	4	4	4	4	4	4	4	Pass	A	32
3.2.2 On Input	4	4	4	4	4	4	4	4	4	Pass	AA	32
3.2.4 Consistent Identification	4	4	4	4	4	4	4	4	4	Pass	AA	32
3.3.1 Error Identification	4	4	4	4	4	4	4	4	4	Pass	A	32
3.3.2 Labels or Instructions	4	0	0	0	0	0	0	0	0	Fail	A	4
3.3.3 Error Suggestion	4	4	4	4	4	4	4	4	4	Pass	AA	32
3.3.4 Error Prevention (Legal, Financial, Data)	4	4	4	1	4	1	4	4	4	Pass	AA	26
4.1.1 Parsing	2	2	2	2	2	2	0	2	0	Fail	A	14
4.1.2 Name, Role, Value	4	2	3	3	3	0	2	2	0	Fail	A	19
4.1.3 Status Messages	0	4	4	4	1	4	4	4	0	Fail	AA	25

Table 11: All criteria for DevOps tools

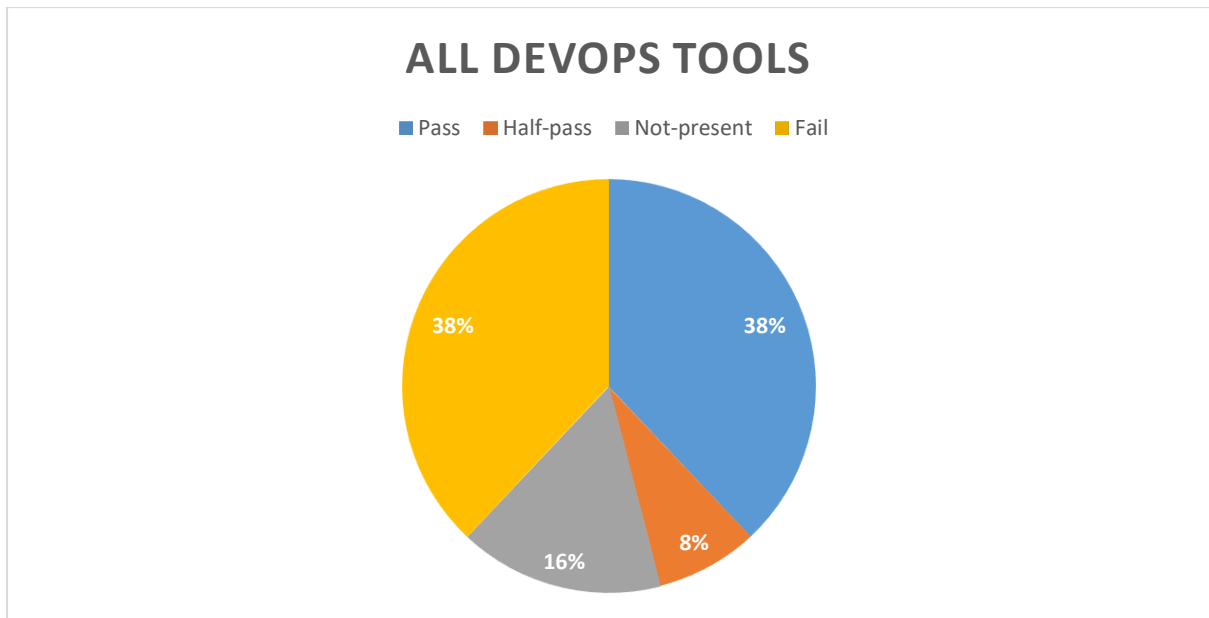


Figure 30: DevOps tools aggregate lowest score

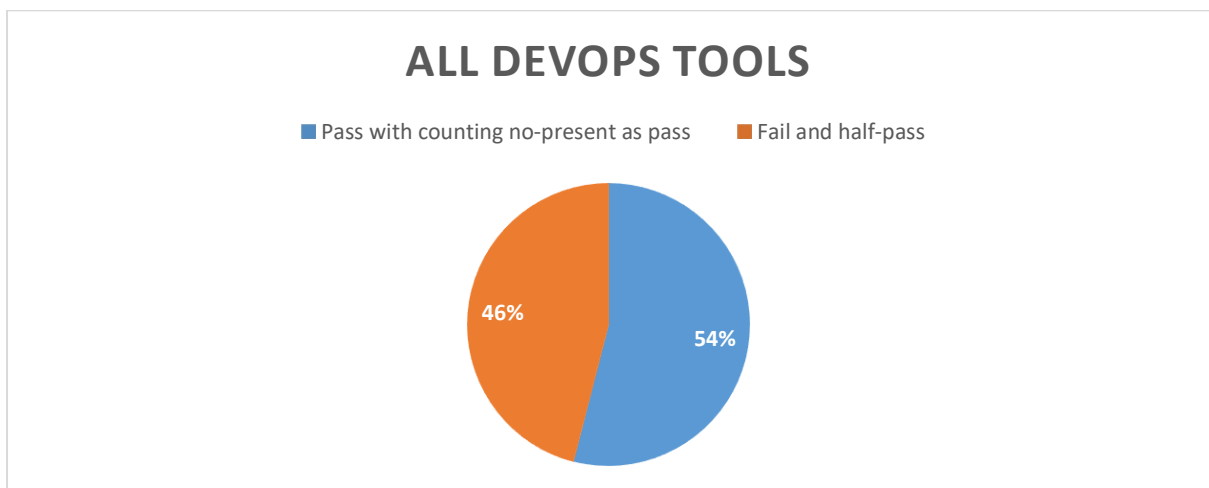


Figure 31: DevOps tools result counting no-present as pass for aggregate lowest score

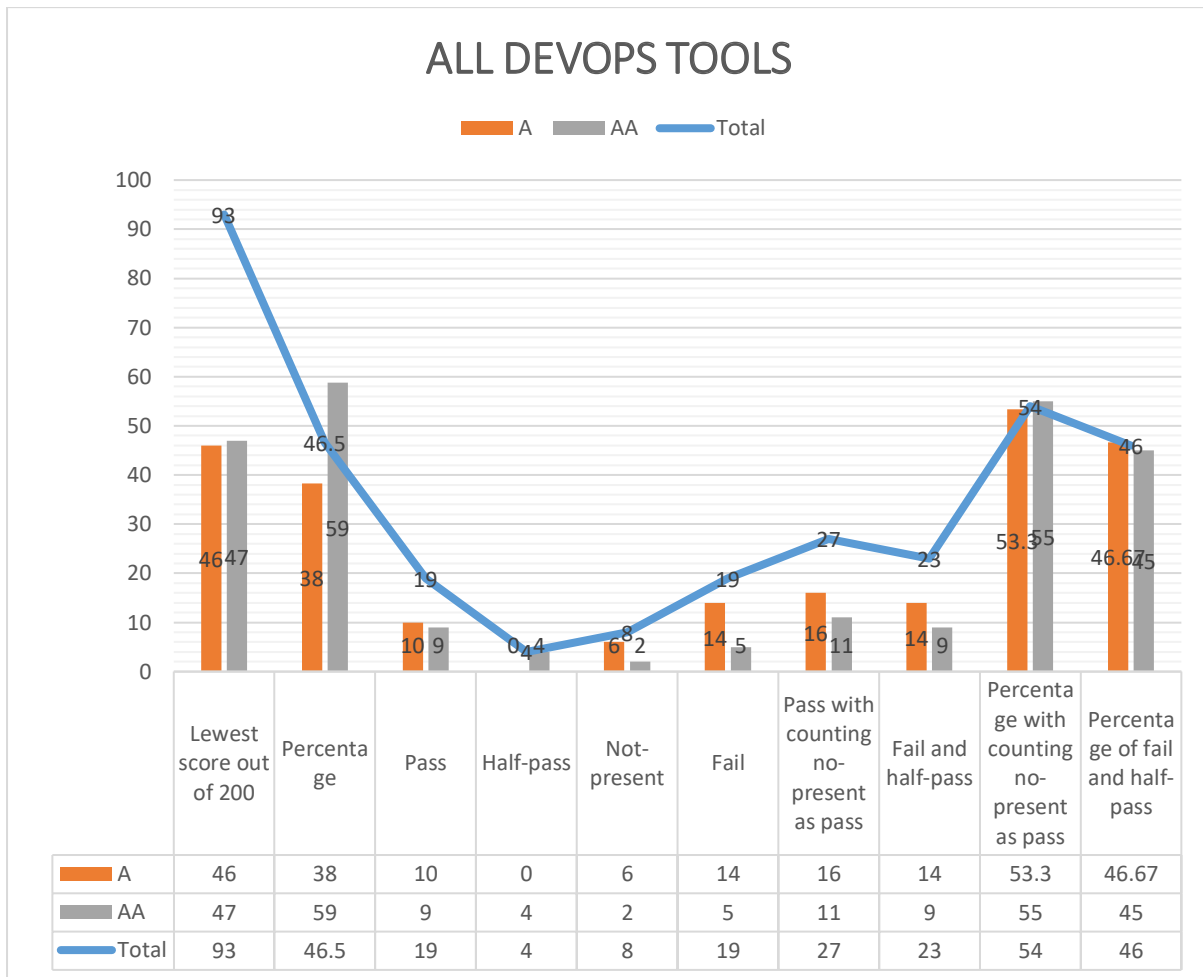


Figure 32: DevOps tools Aggregate lowest score results

In Table 12, we can see the DevOps tools result for all the success criteria based on choosing the lowest score (summary), in Table 13 based on total score, and in Table 14, total score not counting not-present as pass. Based on these tables mentioned earlier, we clearly see which criteria *the selected tools* are most likely to pass. The criteria with pass result, highest total score, and highest total score counting not-present as pass have some shared success criteria as shown in Table 12, Table 13, and Table 14. The criteria fail, half-fail, not-present varies. The variation depends on the result calculations. Choosing the lowest score (summary) means that we are strict with giving the pass score since one criterion fails for any DevOps tools, then the result is fail. The total score shows results based on numbers; the higher the number, the most likely to pass. However, this can be misleading if a success criterion is not-present for all DevOps, then will suggest the score 8, which is low. For example, success criterion 1.2.4. Captions (Live) is not-present for any DevOps tools. To omit the confusion if a success criterion is not-present for all DevOps, total score counting not-

present as pass is introduced. Total score counting not-present as pass more or less shows a more accurate result when ordered from lowest to highest, as shown in Table 14.

No.	Success criteria	Summary
1	1.1.1 Non-text content	<b>Fail</b>
2	1.2.1. Audio-only and Video-only	<b>Fail</b>
3	1.2.3. Audio Description or Media Alternative (Pre-recorded)	<b>Fail</b>
4	1.2.5. Audio Description (Pre-recorded)	<b>Fail</b>
5	1.3.1. Info and Relationships	<b>Fail</b>
6	1.3.2. Meaningful Sequence	<b>Fail</b>
7	1.4.10 Reflow	<b>Fail</b>
8	2.1.1. Keyboard	<b>Fail</b>
9	2.1.2. No Keyboard Trap	<b>Fail</b>
10	2.4.1 Bypass Blocks	<b>Fail</b>
11	2.4.3 Focus Order	<b>Fail</b>

12	2.4.4 Link Purpose (In Context)	<b>Fail</b>
13	2.4.6 Headings and Labels	<b>Fail</b>
14	2.4.7 Focus Visible	<b>Fail</b>
15	3.1.1 Language of Page	<b>Fail</b>
16	3.3.2 Labels or Instructions	<b>Fail</b>
17	4.1.1 Parsing	<b>Fail</b>
18	4.1.2 Name, Role, Value	<b>Fail</b>
19	4.1.3 Status Messages	<b>Fail</b>
20	1.3.5 Identify Input Purpose	<b>Half-pass</b>
21	1.4.3 Contrast (Minimum)	<b>Half-pass</b>
22	1.4.11 Non-Text Contrast	<b>Half-pass</b>
23	1.4.12 Text Spacing	<b>Half-pass</b>
24	1.2.2. Captions (Pre-recorded audio)	<b>Not- present</b>

25	1.2.4. Captions (Live)	<b>Not-present</b>
26	1.4.2 Audio Control	<b>Not-present</b>
27	2.2.1 Timing Adjustable	<b>Not-present</b>
28	2.2.2 Pause, Stop, Hide	<b>Not-present</b>
29	2.3.1 Three Flashes or Below Threshold	<b>Not-present</b>
30	2.5.4 Motion Actuation	<b>Not-present</b>
31	3.1.2 Language of Parts	<b>Not-present</b>
32	1.3.3. Sensory Characteristics	<b>Pass</b>
33	1.3.4 Orientation	<b>Pass</b>
34	1.4.1 Use of Color	<b>Pass</b>
35	1.4.4 Resize text	<b>Pass</b>
36	1.4.5 Images of Text	<b>Pass</b>
37	1.4.13 Content on Hover or Focus	<b>Pass</b>



38	2.1.4. Character Key Shortcuts	<b>Pass</b>
39	2.4.2 Page Titled	<b>Pass</b>
40	2.4.5 Multiple Ways	<b>Pass</b>
41	2.5.1 Pointer Gestures	<b>Pass</b>
42	2.5.2 Pointer Cancellation	<b>Pass</b>
43	2.5.3 Label in Name	<b>Pass</b>
44	3.2.1 On Focus	<b>Pass</b>
45	3.2.2 On Input	<b>Pass</b>
46	3.2.2 On Input	<b>Pass</b>
47	3.2.4 Consistent Identification	<b>Pass</b>
48	3.3.1 Error Identification	<b>Pass</b>
49	3.3.3 Error Suggestion	<b>Pass</b>
50	3.3.4 Error Prevention (Legal, Financial, Data)	<b>Pass</b>

Table 12: Result of all Criteria based on the lowest score

No.	Success criteria	Total score
1	2.4.1 Bypass Blocks	4
2	3.3.2 Labels or Instructions	4
3	1.2.1. Audio-only and Video-only	5
4	1.2.3. Audio Description or Media Alternative (Pre-recorded)	5
5	1.2.5. Audio Description (Pre-recorded)	5
6	1.2.2. Captions (Pre-recorded audio)	8
7	1.2.4. Captions (Live)	8
8	2.2.1 Timing Adjustable	8
9	2.2.2 Pause, Stop, Hide	8

10	2.3.1 Three Flashes or Below Threshold	8
11	2.5.4 Motion Actuation	8
12	3.1.2 Language of Parts	8
13	1.3.1. Info and Relationships	13
14	1.4.11 Non-Text Contrast	14
15	2.1.4. Character Key Shortcuts	14
16	4.1.1 Parsing	14
17	2.4.6 Headings and Labels	15
18	1.3.2. Meaningful Sequence	16
19	1.4.3 Contrast (Minimum)	16
20	3.1.1 Language of Page	16
21	1.4.2 Audio Control	17

22	2.4.4 Link Purpose (In Context)	17
23	2.4.7 Focus Visible	19
24	4.1.2 Name, Role, Value	19
25	1.1.1 Non-text content	22
26	2.4.3 Focus Order	22
27	2.1.2. No Keyboard Trap	24
28	4.1.3 Status Messages	25
29	1.4.10 Reflow	26
30	3.3.4 Error Prevention (Legal, Financial, Data)	26
31	2.1.1. Keyboard	28
32	1.4.5 Images of Text	29
33	1.3.5 Identify Input Purpose	30
34	1.4.12 Text Spacing	30
35	1.3.3. Sensory Characteristics	32

36	1.3.4 Orientation	32
37	1.4.1 Use of Color	32
38	1.4.13 Content on Hover or Focus	32
39	1.4.4 Resize text	32
40	2.4.2 Page Titled	32
41	2.4.5 Multiple Ways	32
42	2.5.1 Pointer Gestures	32
43	2.5.2 Pointer Cancellation	32
44	2.5.3 Label in Name	32
45	3.2.1 On Focus	32
46	3.2.2 On Input	32
47	3.2.2 On Input	32
48	3.2.4 Consistent Identification	32
49	3.3.1 Error Identification	32
50	3.3.3 Error Suggestion	32

Table 13: Result of all Criteria based on total score

No.	Success criteria	Total score counting not-present as pass
1	2.4.1 Bypass Blocks	4
2	3.3.2 Labels or Instructions	4
3	1.3.1. Info and Relationships	13
4	4.1.1 Parsing	14
5	2.4.6 Headings and Labels	15
6	1.3.2. Meaningful Sequence	16
7	3.1.1 Language of Page	16
8	1.4.3 Contrast (Minimum)	16
9	2.4.4 Link Purpose (In Context)	17
10	2.4.7 Focus Visible	19
11	4.1.2 Name, Role, Value	19

12	1.2.1. Audio-only and Video-only	20
13	1.2.3. Audio Description or Media Alternative (Pre-recorded)	20
14	1.2.5. Audio Description (Pre- recorded)	20
15	1.4.11 Non-Text Contrast	20
16	1.1.1 Non-text content	22
17	2.4.3 Focus Order	22
18	2.1.2. No Keyboard Trap	24
19	1.4.10 Reflow	26
20	2.1.1. Keyboard	28
21	4.1.3 Status Messages	28
22	1.4.12 Text Spacing	30
23	1.3.5 Identify Input Purpose	30

24	1.2.2. Captions (Pre-recorded audio)	32
25	1.2.4. Captions (Live)	32
26	1.3.3. Sensory Characteristics	32
27	1.3.4 Orientation	32
28	1.4.1 Use of Color	32
29	1.4.2 Audio Control	32
30	1.4.4 Resize text	32
31	1.4.5 Images of Text	32
32	1.4.13 Content on Hover or Focus	32
33	2.1.4. Character Key Shortcuts	32
34	2.2.1 Timing Adjustable	32
35	2.2.2 Pause, Stop, Hide	32



36	2.3.1 Three Flashes or Below Threshold	32
37	2.4.2 Page Titled	32
38	2.4.5 Multiple Ways	32
39	2.5.1 Pointer Gestures	32
40	2.5.2 Pointer Cancellation	32
41	2.5.3 Label in Name	32
42	2.5.4 Motion Actuation	32
43	3.1.2 Language of Parts	32
44	3.2.1 On Focus	32
45	3.2.2 On Input	32
46	3.2.2 On Input	32
47	3.2.4 Consistent Identification	32
48	3.3.1 Error Identification	32

49	3.3.3 Error Suggestion	32
50	3.3.4 Error Prevention (Legal, Financial, Data)	32

Table 14: Result of all Criteria based on total score counting not-present as pass

All the abovementioned data shows the over-reliance on number scoring, which may not be advisable [62] since the evaluation results shown above can be manipulated differently. Depending on what the user is looking after, as per the results shown in Figure 33 and Table 15. There is a high variation between the results for overall average (A), aggregate lowest score (B), and aggregate lowest score counting not-present as pass (C) as previously shown and discussed in the above segments. Failure percentage (fail and half-pass) rose from 26.7% to 46%, with a variation of 19%. Variation between pass percentage between A and B is high by 13%. However, pass variations Between A and C are almost the same. The high variation proves the point that over-reliance on number scoring is not suggested.

<b>Result</b>	<b>Overall average (A)</b>	<b>Aggregate lowest score (B)</b>	<b>Aggregate lowest score counting not-present as pass (C)</b>	<b>Variation Between A and B</b>	<b>Variation Between A and C</b>	<b>Variation Between B and C</b>
Lowest score %	65.1875	46.5	58.5	18.6875	6.6875	12
Pass %	51.25	38	54	13.25	2.75	16
Half-pass %	14.75	8	8	6.75	6.75	0
Not-present %	22	16	0	6	22	16
Fail %	12	38	38	26	26	0
Pass with counting no-present as pass %	73.25	54	54	19.25	19.25	0
Fail and half-pass %	26.75	46	46	19.25	19.25	0
Total score %	70.4375	64.9375	81.6875	5.5	11.25	16.75

Table 15: Comparison of Overall, Aggregate lowest score and Aggregate lowest score counting not-present as pass results

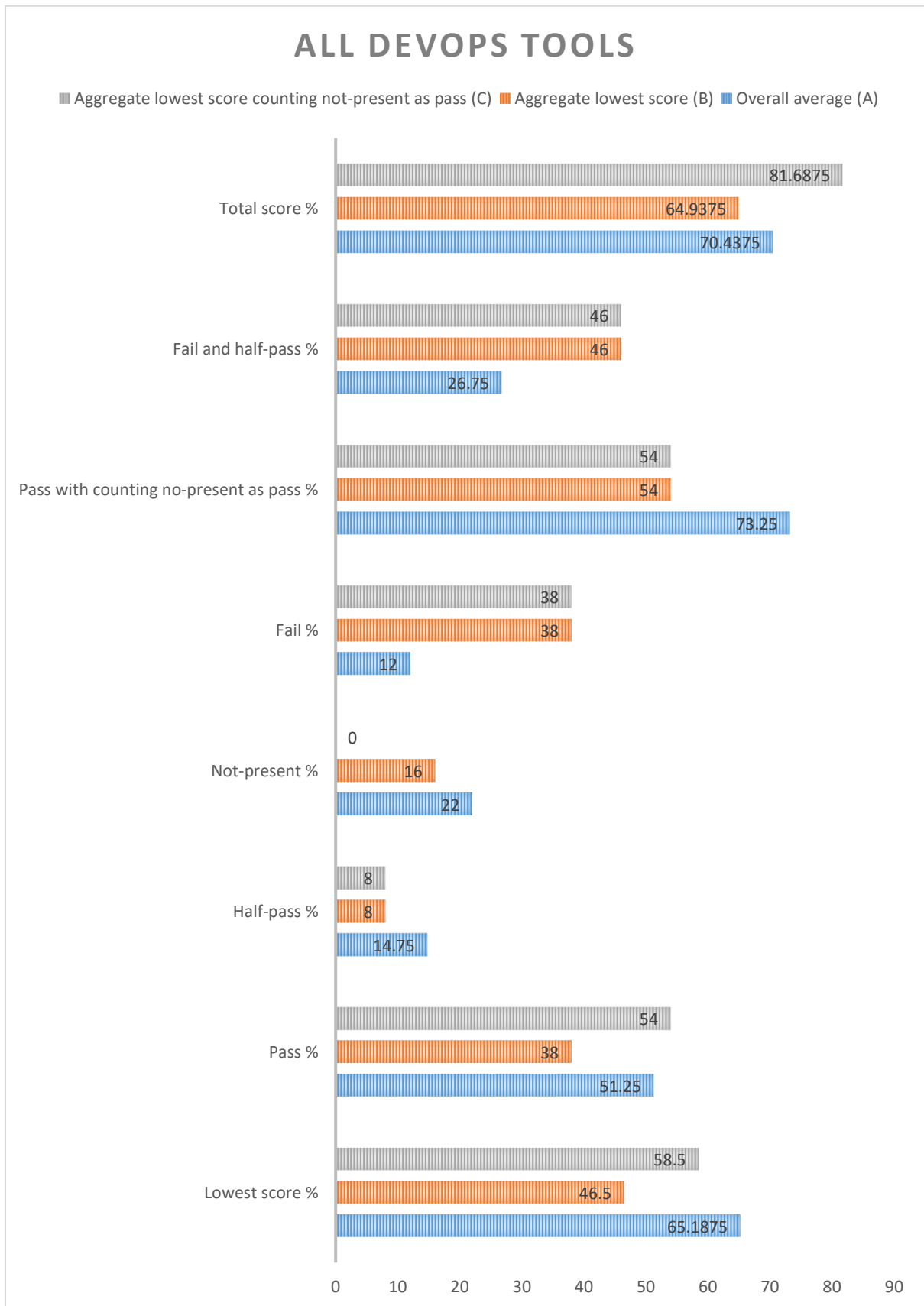


Figure 33: Comparison of Overall, Aggregate lowest score and Aggregate lowest score counting not-present as pass results

## 5.1 LIMITATIONS AND FUTURE WORK

Due to the scope and shortage of time of the master thesis, hereinafter are the suggestions for finding out accessibility for DevOps tools:

- Representative Sample should have more web pages than chosen for this master thesis.
- Include more DevOps tools for testing.
- Learn and test each DevOps tool to gain more understating and produce accurate results.

## 6 CONCLUSION

---

Automated and simulation software tools do not consistently deliver accurate outcomes since not all usability issues are found automatically. Besides, as per Brajnik [62], a tool can generate positive failures up to 33% and adverse failures up to 35%, respectively. It is vital to eradicate such positive and negative errors using a variety of resources. Implementing the WCAG on websites benefits both disabled and non-disabled user groups. Therefore, the higher the WCAG conformance level, the better outcome for both groups [42].

The score results obtained by DIFI in 2014 and 2018 are 51% and 60%, respectively. The results obtained from the master thesis are 65% for the lowest score, and the total score is 70%. Comparing DIFI's 2018 results and the master thesis, there is a 5% variation with the lowest score and 10% for the total score. The results mentioned in the discussion section show that there is 27% failure (12% fail and 15% half-fail). Therefore, to answer the research question "to what extent are DevOps tools accessible according to WCAG guidelines?", the selected online DevOps tools need to be made more accessible for developers with disabilities; thus, they do not face any issues using these tools. Switching from one DevOps tool to another will not solve the accessibility issues. Certain limitations must be dealt with when implementing software like DevOps. These tools are resistant to changes.

Transformation strategies have to be intelligently devised when implementing changes in the software like these, or accessibility must be a key point from the start of development instead of considering it an intrusion that may lead to compromises.

## 7 APPENDIX A

---

### 7.1 WCAG 2.1

Table 16 shows the summary of WCAG 2.0 with the first three layers of guidance (*principles, guidelines, and success criteria*) and conformance level. In the table, the new success criteria from WCAG 2.1 are included and marked as red.

Principles	Guidelines	Success Criteria	Conformance level
<b>1.</b> <i>Perceivable</i>	1.1 Text Alternatives	1.1.1 Non-text content	A
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.1 Audio-only and Video-only	A
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.2 Captions (Pre-recorded)	A
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.3 Audio Description or Media Alternative (Pre-recorded)	A
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.4 Captions (Live)	AA
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.5 Audio Description (Pre-recorded)	AA
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.6 Sign Language (Pre-recorded)	AAA
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.7 Extend Audio Description (Pre-recorded)	AAA

<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.8 Media Alternative (Pre-recorded)	AAA
<b>1.</b> <i>Perceivable</i>	1.2 Time-based Media	1.2.9 Audio-only (Live)	AAA
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.1 Info and Relationships	A
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.2 Meaningful Sequence	A
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.3 Sensory Characteristics	A
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.4 Orientation	AA
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.5 Identify Input Purpose	AA
<b>1.</b> <i>Perceivable</i>	1.3 Adaptable	1.3.6 Identify Purpose	AAA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.1 Use of Color	A
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.2 Audio Control	A
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.3 Contrast (Minimum)	AA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.4 Resize text	AA

<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.5 Images of Text	AA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.6 Contrast (Enhanced)	AAA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.7 Low or No Background Audio	AAA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.8 Visual Presentation	AAA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.9 Images of Text (No Exception)	AAA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.10 Reflow	AA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.11 Non-Text Contrast	AA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.12 Text Spacing	AA
<b>1.</b> <i>Perceivable</i>	1.4 Distinguishable	1.4.13 Content on Hover or Focus	AA
<b>2. Operable</b>	2.1 Keyboard Accessible	2.1.1 Keyboard	A
<b>2. Operable</b>	2.1 Keyboard Accessible	2.1.2 No Keyboard Trap	A
<b>2. Operable</b>	2.1 Keyboard Accessible	2.1.3 Keyboard (No Exception)	AAA

<b>2. Operable</b>	2.1 Keyboard Accessible	2.1.4 Character Key Shortcuts	A
<b>2. Operable</b>	2.2 Enough Time	2.2.1 Timing Adjustable	A
<b>2. Operable</b>	2.2 Enough Time	2.2.2 Pause, Stop, Hide	A
<b>2. Operable</b>	2.2 Enough Time	2.2.3 No Timing	AAA
<b>2. Operable</b>	2.2 Enough Time	2.2.4 Interruptions	AAA
<b>2. Operable</b>	2.2 Enough Time	2.2.5 Re-authenticating	AAA
<b>2. Operable</b>	2.2 Enough Time	2.2.6 Timeouts	AAA
<b>2. Operable</b>	2.3 Seizures and Physical Reactions	2.3.1 Three Flashes or Below Threshold	A
<b>2. Operable</b>	2.3 Seizures and Physical Reactions	2.3.2 Three Flashes	AAA
<b>2. Operable</b>	2.3 Seizures and Physical Reactions	2.3.3 Animation from Interactions	AAA
<b>2. Operable</b>	2.4 Navigable	2.4.1 Bypass Blocks	A
<b>2. Operable</b>	2.4 Navigable	2.4.2 Page Titled	A
<b>2. Operable</b>	2.4 Navigable	2.4.3 Focus Order	A
<b>2. Operable</b>	2.4 Navigable	2.4.4 Link Purpose (In Context)	A
<b>2. Operable</b>	2.4 Navigable	2.4.5 Multiple Ways	AA
<b>2. Operable</b>	2.4 Navigable	2.4.6 Headings and Labels	AA
<b>2. Operable</b>	2.4 Navigable	2.4.7 Focus Visible	AA



<b>2. Operable</b>	2.4 Navigable	2.4.8 Location	AAA
<b>2. Operable</b>	2.4 Navigable	2.4.9 Link Purpose (Link Only)	AAA
<b>2. Operable</b>	2.4 Navigable	2.4.10 Section Headings	AAA
<b>2. Operable</b>	2.5 Input Modalities	2.5.1 Pointer Gestures	A
<b>2. Operable</b>	2.5 Input Modalities	2.5.2 Pointer Cancellation	A
<b>2. Operable</b>	2.5 Input Modalities	2.5.3 Label in Name	A
<b>2. Operable</b>	2.5 Input Modalities	2.5.4 Motion Actuation	A
<b>2. Operable</b>	2.5 Input Modalities	2.5.5 Target Size	AAA
<b>2. Operable</b>	2.5 Input Modalities	2.5.6 Concurrent Input Mechanisms	AAA
<b>3. Understand</b>	3.1. Readable	3.1.1 Language of Page	A
<b>3. Understand</b>	3.1. Readable	3.1.2 Language of Parts	AA
<b>3. Understand</b>	3.1. Readable	3.1.3 Unusual Words	AAA
<b>3. Understand</b>	3.1. Readable	3.1.4 Abbreviations	AAA
<b>3. Understand</b>	3.1. Readable	3.1.5 Reading Level	AAA
<b>3. Understand</b>	3.1. Readable	3.1.6 Pronunciation	AAA
<b>3. Understand</b>	3.2. Predictable	3.2.1 On Focus	A

<b>3.</b> <i>Understand</i>	3.2. Predictable	3.2.2 On Input	A
<b>3.</b> <i>Understand</i>	3.2. Predictable	3.2.3 Consistent Navigation	AA
<b>3.</b> <i>Understand</i>	3.2. Predictable	3.2.4 Consistent Identification	AA
<b>3.</b> <i>Understand</i>	3.2. Predictable	3.2.5 Change on Request	AAA
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.1 Error Identification	A
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.2 Labels or Instructions	A
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.3 Error Suggestion	AA
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.4 Error Prevention (Legal, Financial, Data)	AA
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.5 Help	AAA
<b>3.</b> <i>Understand</i>	3.3 Input Assistance	3.3.6 Error Prevention (All)	AAA
<b>4. Robust</b>	4.1 Compatible	4.1.1 Parsing	A
<b>4. Robust</b>	4.1 Compatible	4.1.2 Name, Role, Value	A
<b>4. Robust</b>	4.1 Compatible	4.1.3 Status Messages	AA

Table 16: Summary of WCAG 2.0 AND WCAG 2.1

## 8 APPENDIX B

---

### 8.1 THE 2014 DIFI SURVEY ON THE UNIVERSAL DESIGN OF WEBSITES IN THE PRIVATE AND PUBLIC SECTOR

DIFI conducted surveys on 300 Norwegian websites regarding WCAG 2.0 guidelines as part of the road to universal design. According to the DIFI report, the survey was conducted on private and public sectors between September 2014 and January 2015 [55]. The survey aimed to determine the level of compliance of the selected websites and alleviate digital exclusivity. The selection of the websites is based on the importance of the services they provide. That secure and guarantee inclusivity and websites with a high number of users. The 304 selected websites from both sectors are divided into six groups that correspond to the background of their services:

- Public and local administration, Human health. (126 organizations)
- Banking and finance. (48 organizations)
- Information and communication. (36 organizations)
- Travel and accommodation. (40 organizations)
- Retail trade. (34 organizations)
- Agriculture. Fishing. Mining and quarrying. Manufacturing. Electricity, gas. Water supply. (20 organizations)

DIFI introduced 21 indicators to measure compliance or non-compliance for 15 success criteria out of the 35 mandatory success criteria in WCAG 2.0. The highest possible total of points that a website can obtain is 63. The following scores were given: 3 points for full compliance, 0 points for a high level of non-compliance, and 1-2 points were given to identify the degree of non-compliance. Each success criteria have several indicators. The focus points evaluated in the survey were: text alternatives for images, coding, navigability, keyboard navigation, forms. Tests were carried out on Windows 7 using Internet Explorer 11. The test tools used to test the mentioned focus points were Internet Explorer for code inspection, WhatFont for font measurement, W3C HTML validator, and Web Accessibility Toolbar (WAT) for contrast, text alternatives, headings, and tables.

Text alternatives for images have 1 indicator. It is considered a risk area.

- *Text-alternatives for images*: total percentage of for 7% compliance and 97% non-compliance

Coding has 7 indicators. It is considered the main risk area.

- *Headings are coded with correct markup*: total percentage of for 20% compliance and 80% non-compliance
- *Headings are coded with the correct nesting level*: total percentage of for 44% compliance and 56% non-compliance
- *Data tables are coded with correct markup*: total percentage of for 91% compliance and 9% non-compliance
- *Data tables have correctly defined header cells*: total percentage of for 2% compliance and 98% non-compliance
- *Form controls have correct markup*: total percentage of for 33% compliance and 67% non-compliance
- *Search fields have correct markup*: total percentage of for 12% compliance and 88% non-compliance
- *The code does not contain syntax errors*: total percentage of for 46% compliance and 54% non-compliance

Navigability has 6 indicators. Overall, it is not considered a risk area, but it has specific points that pose a high risk of non-compliance.

- *Search fields have instructions or labels*: total percentage of for 43% compliance and 57% non-compliance
- *Purpose and function of links are understandable*: total percentage of for 72% compliance and for 28% non-compliance
- *Links are distinguishable from headings and body text*: total percentage of for 16% compliance and 84% non-compliance
- *There is sufficient contrast between text and its background*: total percentage of for 16% compliance and 84% non-compliance
- *Text can be resized without loss of content or functionality*: total percentage of for 81% compliance and 19% non-compliance

- *There are multiple ways of navigating the website*: total percentage of for 63% compliance and 27% non-compliance

Keyboard navigation has 6 indicators. It is not considered a risk area, but it has challenges that give it the potential to be a risk area.

- *No keyboard traps*: total percentage of for 96% compliance and 4% non-compliance
- *There is a link to skip to the main content of a web page*: total percentage of 9% compliance and 81% non-compliance
- *The skip link has a visible focus indicator*: total percentage of for 56% compliance and 44% non-compliance
- *Links and form controls have a visible focus indicator*: total percentage of for 23% compliance and 77% non-compliance
- *Selecting form controls through keyboard has predictable effects*: total percentage of for 86% compliance and 14% non-compliance

Lastly, forms have 5 indicators. It does not pose a risk area; however, it can become a risk area.

- *Form controls have correct markup*: total percentage of for 33% compliance and 67% non-compliance
- *All content can be operated through a keyboard*: total percentage of for 74% compliance and for 26% non-compliance
- *Descriptive labels in forms*: total percentage of for 92% compliance and 8% non-compliance
- *Links and form controls have a visible focus indicator*: total percentage of for 23% compliance and 77% non-compliance
- *Selecting form controls through keyboard has predictable effects*: total percentage of for 86% compliance and 14% non-compliance

## 9 REFERENCES

---

- [1] M. J. Mueller, R. Jolly, and E. Eggert, “Web Accessibility Laws & Policies,” *World Wide Web Consortium (W3C)*, 2018. <https://www.w3.org/WAI/policies/>.
- [2] B. Vollenwyder, G. H. Iten, F. Brühlmann, K. Opwis, and E. D. Mekler, “Salient beliefs influencing the intention to consider Web Accessibility,” *Comput. Hum. Behav.*, vol. 92, pp. 352–360, Mar. 2019, doi: 10.1016/j.chb.2018.11.016.
- [3] United Nations, *Convention on the Rights of Persons with Disabilities and Optional Protocol*. 2006.
- [4] B. Bruce, *Sams teach yourself Macromedia Dreamweaver 4 in 24 hours*. Indianapolis, Ind: Sams, 2001.
- [5] A. Aizpurua, S. Harper, and M. Vigo, “Exploring the relationship between web accessibility and user experience,” *Int. J. Hum.-Comput. Stud.*, vol. 91, pp. 13–23, Jul. 2016, doi: 10.1016/j.ijhcs.2016.03.008.
- [6] S. Song *et al.*, “Web Accessibility Evaluation in a Crowdsourcing-Based System with Expertise-Based Decision Strategy,” in *Proceedings of the Internet of Accessible Things on - W4A '18*, Lyon, France, 2018, pp. 1–4, doi: 10.1145/3192714.3192827.
- [7] R. Lopes, D. Gomes, and L. Carriço, “Web not for all: a large scale study of web accessibility,” in *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) - W4A '10*, Raleigh, North Carolina, 2010, p. 1, doi: 10.1145/1805986.1806001.
- [8] European Commission, “Web Accessibility | Shaping Europe’s digital future,” *European Commission*, 2019. <https://ec.europa.eu/digital-single-market/en/web-accessibility>.
- [9] World Health Organization, “Disability and Health,” *Disability and Health*, 2018. <http://www.who.int/en/news-room/fact-sheets/detail/disability-and-health>.
- [10] United Nations, “Department of Economic and Social Affairs Disability,” *Factsheet on Persons with Disabilities*, 2019. <https://www.un.org/development/desa/disabilities/resources/factsheet-on-persons-with-disabilities.html>.
- [11] EOWG, “Introduction to Web Accessibility,” *Introduction to Web Accessibility*, 2005. <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.

- [12] W3C, "Accessibility, Usability, and Inclusion," *Accessibility, Usability, and Inclusion*, 2010. <https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/>.
- [13] Stack Overflow, "Developer Survey Results 2017," *Stack Overflow*, 2017. <https://insights.stackoverflow.com/survey/2017>.
- [14] community Stack Overflow, "Developer Survey Results 2018," *Stack Overflow*, 2018. <https://insights.stackoverflow.com/survey/2018/>.
- [15] community Stack Overflow, "Developer Survey Results 2019," *Stack Overflow*, 2019. <https://insights.stackoverflow.com/survey/2019>.
- [16] E. Manser, "Inclusive Design: IBM Employees with Disabilities Volunteer Time to Create Accessible Technology," *IBM*, 2018. <https://www.ibm.com/blogs/age-and-ability/2018/03/07/ibm-employees-disabilities-volunteer-time-create-accessible-technology/>.
- [17] M. Kearney, D. Gash, A. Boxhall, and R. Dodson, "Accessibility | Web Fundamentals | Google Developers," *Accessibility | Web Fundamentals | Google Developers*, 2019. <https://developers.google.com/web/fundamentals/accessibility>.
- [18] "Google Accessibility support," *Google Accessibility support*, 2020. <https://developers.google.com/web/fundamentals/accessibility>.
- [19] Á. Menezes and R. Prikladnicki, "Diversity in software engineering," in *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '18*, Gothenburg, Sweden, 2018, pp. 45–48, doi: 10.1145/3195836.3195857.
- [20] S. Kurt, "Moving toward a universally accessible web: Web accessibility and education," *Assist. Technol.*, vol. 31, no. 4, pp. 199–208, Aug. 2019, doi: 10.1080/10400435.2017.1414086.
- [21] P. Ralph, S. Baltes, G. Adisaputri, and R. Torkar, "Pandemic Programming How COVID-19 affects software developers and how their organizations can help," 2020.
- [22] M. Kulkarni and M. L. Lengnick-Hall, "Obstacles to Success in the Workplace for People With Disabilities: A Review and Research Agenda," *Hum. Resour. Dev. Rev.*, vol. 13, no. 2, pp. 158–180, Jun. 2014, doi: 10.1177/1534484313485229.
- [23] M. Sánchez-Gordón and R. Colomo-Palacios, "Characterizing DevOps Culture: A Systematic Literature Review," in *Software Process Improvement and Capability Determination*, Cham, 2018, pp. 3–15.

- [24] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "DevOps," *IEEE*, 2016, [Online]. Available: <https://ieeexplore.ieee.org/document/7458761/>.
- [25] M. Senapathi, J. Buchan, and H. Osman, "DevOps Capabilities, Practices, and Challenges: Insights from a Case Study," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 - EASE'18*, Christchurch, New Zealand, 2018, pp. 57–67, doi: 10.1145/3210459.3210465.
- [26] W3C/WAI, "Website Accessibility Conformance Evaluation Methodology (WCAG-EM) 1.0," 2014. <https://www.w3.org/TR/WCAG-EM/>.
- [27] Elsevier, "Guidelines - Accessibility Checklist," *Guidelines - Accessibility Checklist*, 2019. [http://romeo.elsevier.com/accessibility\\_checklist/](http://romeo.elsevier.com/accessibility_checklist/).
- [28] E. Eggert, "Web Accessibility Evaluation Tools List," *Web Accessibility Evaluation Tools List*, 2016. <https://www.w3.org/WAI/ER/tools/>.
- [29] iso25000, "ISO 25000," *ISO 25000 Software Product Quality*, 2019. <https://iso25000.com/index.php/en/>.
- [30] W3C, "Diverse Abilities and Barriers," *World Wide Web Consortium (W3C)*, 2017. <https://www.w3.org/WAI/people-use-web/abilities-barriers/>.
- [31] W3C, "Older Users and Web Accessibility: Meeting the Needs of Ageing Web Users," 2018. <https://www.w3.org/WAI/older-users/>.
- [32] W3C, "Essential Components of Web Accessibility," *Essential Components of Web Accessibility*, 2018. <https://www.w3.org/WAI/fundamentals/components/>.
- [33] EOWG, "Tools and Techniques," *W3C*, 2017. <https://www.w3.org/WAI/people-use-web/tools-techniques/>.
- [34] W3C, "About W3C," *About W3C*, 2019. <https://www.w3.org/Consortium/>.
- [35] W3C/WAI, "W3C MISSION," *W3C MISSION*, 2019. <https://www.w3.org/Consortium/mission#principles>.
- [36] World Wide Web Consortium (W3C), "Web Content Accessibility Guidelines (WCAG), version 2.0," *World Wide Web Consortium (W3C)*, 2011. <https://www.w3.org/TR/WCAG20/>.
- [37] W3C/WAI, "Accessibility Principles," 2019. <https://www.w3.org/WAI/fundamentals/accessibility-principles/>.
- [38] M. Vigo, J. Brown, and V. Conway, "Benchmarking web accessibility evaluation tools: measuring the harm of sole reliance on automated tests," p. 10, 2013.



- [39] W3C/WAI, "Selecting Web Accessibility Evaluation Tools," *Selecting Web Accessibility Evaluation Tools*, 2017. <https://www.w3.org/WAI/test-evaluate/tools/selecting/>.
- [40] G. Brajnik, "Web accessibility testing: when the method is the culprit," 2006.
- [41] W3C/WAI, "WCAG-EM Overview: Website Accessibility Conformance Evaluation Methodology," *WCAG-EM Overview: Website Accessibility Conformance Evaluation Methodology*, 2016. <https://www.w3.org/WAI/test-evaluate/tools/selecting/>.
- [42] S. Schmutz, A. Sonderegger, and J. Sauer, "Implementing Recommendations From Web Accessibility Guidelines: Would They Also Provide Benefits to Nondisabled Users," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 58, no. 4, pp. 611–629, Jun. 2016, doi: 10.1177/0018720816640962.
- [43] S. Keohane, "DevOps: Three Ways to Improve DevOps to Ensure Accessibility," *IBM*, 2015. <https://www.ibm.com/blogs/age-and-ability/2015/12/02/three-ways-to-improve-devops-to-ensure-accessibility/>.
- [44] G. B. Ghantous and A. Gill, "DevOps: Concepts, Practices, Tools, Benefits and Challenges," *PACIS 2017 Proceedingd*, no. 96, p. 13, 2017.
- [45] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "From Agile to DevOps: Smart Skills and Collaborations," *Inf. Syst. Front.*, Mar. 2019, doi: 10.1007/s10796-019-09905-1.
- [46] R. Jabbari, N. Ali, K. Petersen, and B. Tanveer, "What is DevOps?: A Systematic Mapping Study on Definitions and Practices," 2016, pp. 1–11, doi: 10.1145/2962695.2962707.
- [47] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and Its Practices," *IEEE Softw.*, vol. 33, no. 3, pp. 32–34, May 2016, doi: 10.1109/MS.2016.81.
- [48] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017, doi: 10.1109/ACCESS.2017.2685629.
- [49] Xebialabs, "XebiaLabd: Enterprise DevOps and Application Release Automation," *XebiaLabs: Enterprise DevOps and Application Release Automation*, 2019. <https://xebialabs.com/>.
- [50] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?," *Empir. Softw. Eng.*, vol. 22, no. 6, pp. 3149–3185, Dec. 2017, doi: 10.1007/s10664-017-9514-4.

- [51] DIFI, "Digital barriers on Norwegian websites, status report," *DIFI*, 2019, [Online]. Available: <https://uu.DIFI.no/om-oss/english/status-universal-design-ict/digital-barriers-norwegian-websites-2018>.
- [52] Norwegian on the Ministry of Children, Equality and Social Inclusion, "UN Convention on the Rights of Persons with Disabilities—Norway's Initial Report," Initial Report, 2009. [Online]. Available: <https://www.regjeringen.no/contentassets/26633b70910a44049dc065af217cb201/crpd-initial-report-norway-english-01072015.pdf>.
- [53] Ministry of Local Government and Modernisation, "Regulations for universal design of information and communication technology solutions," 2013. <https://lovdata.no/dokument/SF/forskrift/2013-06-21-732?q=ICT>.
- [54] DIFI, "Indicators for Universal Design of Web solutions," *Indicators for Universal Design of Web solutions*, 2014. <https://uu.DIFI.no/om-oss/english/indicators-universal-design-web-solutions>.
- [55] DIFI, "Status report on the universal design of websites in the private and public sector – 2014," DIFI, Norway, Status Report 7, 2015.
- [56] DIFI, "Digital barriers on Norwegian websites, status report," *DIFI*, 2018, [Online]. Available: <https://www.uutilsynet.no/english/digital-barriers-norwegian-websites-2018/122>.
- [57] W3C/WAI, "WCAG-EM Report Tool," *WCAG-EM Report Tool*, 2016. <https://www.w3.org/WAI/eval/report-tool/#/>.
- [58] W3C/WAI, "Evaluation and Report Language (EARL) 1.0 Schema," 2017.
- [59] WebAIM, "WebAIM's WCAG 2 Checklist," *WebAIM's WCAG 2 Checklist*, 2019. <https://webaim.org/standards/wcag/checklist>.
- [60] World Wide Web Consortium (W3C), "Web Content Accessibility Guidelines (WCAG), version 2.1," *World Wide Web Consortium (W3C)*, 2018. <https://www.w3.org/TR/WCAG21/>.
- [61] W3C/WAI, "How to Meet WCAG (Quick Reference)," *WCAG-EM Report Tool*, 2019. [https://www.w3.org/WAI/WCAG21/quickref/?currentsidebar=%23col\\_customize&level=aaa](https://www.w3.org/WAI/WCAG21/quickref/?currentsidebar=%23col_customize&level=aaa).

- [62] G. Brajnik, Y. Yesilada, and S. Harper, “Is accessibility conformance an elusive property? A study of validity and reliability of WCAG 2.0,” *ACM Trans. Access. Comput.*, vol. 4, no. 2, pp. 1–28, Mar. 2012, doi: 10.1145/2141943.2141946.