

Construction of Web Accessibility Tool for Chrome Extension

A Research Project Submitted in Partial Fulfillment of the Requirement for the

Degree of Bachelor of Computer Science and Engineering

**Submitted By**

Sakil Ahmed

ID: CE18033

Rubaiyat Razin Raha

ID: CE18037

Session: 2017-2018

**Supervised By**

Dr. Md. Sazzad Hossain

Professor

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(CSE)**

**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

**SANTOSH, TANGAIL-1902, BANGLADESH**

**APPROVAL**

This Project titled **“Construction of Web Accessibility Tool for Chrome Extension”**, Submitted by Sakil Ahmed, ID:CE18033 and Rubaiyat Razin Raha, ID:CE18037 to the Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Santosh Tangail-1902, Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Computer Science and Engineering and approved as to its style and contents.

**BOARD OF EXAMINERS**

1. …………………………………………………………… (Supervisor)
2. …………………………………………………………… (Examiner)
3. …………………………………………………………… (Examiner)

**DECLARATION**

We hereby declare that the research project entitled **“Construction of Web Accessibility Tool for Chrome Extension”** has been done by us under the supervision of Dr. Md. Sazzad Hossain, Professor, Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Santosh Tangail-1902, Bangladesh. We have evidence that the research project work or any part of this work has not been submitted anywhere for the award of any degree or diploma. The information has allowed for this document to be accurate and valid to the best of our knowledge.

Dr. Md. Sazzad Hossain

Professor ………………………..

Department of CSE, MBSTU Signature of Supervisor

Sakil Ahmed

ID: CE18033 ..………………………

Signature of Candidate

Rubaiyat Razin Raha

ID:18037 ………………………….

Signature of Candidate

**ACKNOWLEDGEMENT**

First of all, we are grateful to our creator, the Almighty Creator, for giving us all the ability to come so far and for his eternal blessings. God has blessed us with everything one could desire for.

We would like to express our deepest gratitude and profound regard to our honourable supervisor, Dr. Md. Sazzad Hossain, Professor, Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University (MBSTU), for his endless support and guidance. Deep Knowledge and keen interest of our supervisor within the field of “Android Development” to hold out this project. It absolutely was an excellent privilege and honor to figure and study under his direction. I’m truly blessed being under his guidance.

We would wish to thank our senior brother who helped us in the project with their ideas and suggestion.

And last not the least, We are extremely grateful to ours parents for their love, prayers, caring and sacrifices for educating and preparing us .

**ABSTRACT**

A web accessible extension is designed to be accessible to users with disabilities and follows accessibility guidelines to ensure that all users can interact with it using assistive technology. A web accessible Chrome extension is a software application that extends the functionality of the Chrome web browser and provides enhanced features and capabilities to users. It is designed to be accessed and interacted with directly from web pages rather than being confined to the browser's toolbar or menu. Web accessible Chrome extensions are software modules that extend the functionality of the Google Chrome web browser, providing additional features and enhancements by integrating with web pages and interacting with their content. The aim of this research is to investigate the potential of web accessible Chrome extensions in improving user productivity, accessibility, and customization in the context of web browsing.

**TABLE OF CONTENTS**

**APPROVAL**

**DECLARATION**

**ACKNOWLEDGEMENT**

**ABSTRACT**

**LIST OF FIGURES**

**Chapter 1 Introduction**

* 1. Introduction
  2. Motivation of the Study
  3. Problem Statement

**Chapter 2 Literature Review**

* 1. Background of Web Accessibility
  2. Web accessibility learning
  3. Web Content Accessibility Guidelines
  4. Web Accessibility Evaluation Tools
  5. Project Goal

**Chapter 3 Methodology**

* 1. Architecture of the Project
  2. Requirement of The Project
  3. Use Case Diagram
  4. Flow Diagram
  5. Technical Details

#### Visual Studio Code Editor

3.5.2 Express JavaScript Framework

3.5.3 Postman

3.5.4 Manifest File

3.5.5 Languages

3.6 Testing

**Chapter 4 Project Features and Functionalities**

4.1 Proposed Extension User Interface

4.2 Proposed Extension open on sample website

4.3 Results

4.4 Effectiveness

**Chapter 5 Conclusion**

**References**

**Chapter 1**

**----------------------------------------------------------------------**

**Introduction**

* 1. **Introduction**

Web accessibility refers to the design and development of websites, applications, and digital content in a way that ensures that people with disabilities can perceive, understand, navigate, interact, and contribute to the web effectively. In other words, web accessibility is about making the web usable for everyone, including people with disabilities, such as visual, auditory, physical, speech, cognitive, and neurological impairments. Web accessibility aims to remove barriers to access and enable people with disabilities to participate fully in the digital world, access information, communicate, learn, work, and socialize. It involves adhering to web accessibility standards, guidelines, and best practices, such as the Web Content Accessibility Guidelines (WCAG), which provide a set of recommendations for creating accessible web content.

* 1. **Motivation of the Study**

There are several motivations for studying and promoting web accessibility.

Firstly, web accessibility is a matter of social responsibility and inclusion. The web has become an essential tool for many aspects of life, including education, employment, healthcare, communication, and entertainment. Therefore, it is crucial to ensure that the web is accessible to everyone, including people with disabilities, who may face significant barriers to accessing and using digital content and services. By promoting web accessibility, we can help to create a more inclusive society that values diversity and respects the rights of all individuals.

Secondly, web accessibility is a legal need in many countries. Governments and regulatory bodies have enacted laws and regulations that mandate organizations and businesses to ensure that their websites and digital content are accessible to people with disabilities. Failure to comply with these laws can result in legal penalties, fines, and damage to the organization's reputation.

Thirdly, web accessibility can also benefit organizations and businesses in terms of increased reach, customer satisfaction, and revenue. By making their websites and digital content accessible, organizations can reach a wider audience, including people with disabilities who may have been excluded before. Moreover, accessible websites are easier to use for everyone, which can improve customer satisfaction and lead to increased sales and revenue.

Overall, the motivation for web accessibility study is driven by the desire to create a more inclusive and accessible digital world that benefits everyone, including people with disabilities, organizations, and society as a whole

* 1. **Problem Statement**

There is a growing need for web accessible tools that cater to individuals with disabilities and provide them with equal access to information, communication, and online services. Despite advancements in technology, many websites and online platforms still lack the necessary accessibility features, making it challenging for people with disabilities to navigate, interact, and benefit from the digital world.

The existing web accessibility tools available in the market often face limitations in terms of functionality, ease of use, and compatibility across different devices and platforms. Moreover, these tools may not adequately address the diverse needs of individuals with various disabilities, such as visual impairments, hearing impairments, motor disabilities, cognitive impairments conditions.

Furthermore, organizations and website developers may not have a comprehensive understanding of web accessibility guidelines and best practices, leading to the creation of inaccessible digital content. This lack of awareness and knowledge about accessibility standards can perpetuate barriers for individuals with disabilities, preventing them from fully participating in online activities and benefiting from the vast resources available on the internet.

Therefore, there is a pressing need for web accessible tools that are inclusive, robust, user-friendly, and compatible across various devices and platforms. These tools should offer comprehensive solutions to address the diverse accessibility needs of individuals with disabilities and empower them to access, navigate, and engage with digital content effectively. Additionally, there is a need for educational resources and training programs to raise awareness and educate website developers and organizations about web accessibility guidelines and best practices, fostering a more inclusive digital environment for all users.

**Chapter 2**

**----------------------------------------------------------------------**

**Literature Review**

**2.2 Web accessibility learning**

Web accessible learning refers to the practice of designing and developing educational resources and online learning environments that are inclusive and accessible to all learners, including those with disabilities. It aims to remove barriers to learning and ensure equal access to educational content and opportunities. Here are some aspects of web accessible learning that can be explored:

* Content Accessibility: One of the key considerations in web accessible learning is ensuring that educational content is accessible to individuals with different abilities. This involves providing alternative formats for content, such as text transcripts or captions for videos, accessible PDFs, and properly formatted documents that can be easily navigated using assistive technologies.
* Website and Interface Design: The design of online learning platforms and websites should follow accessibility standards and guidelines. This includes using clear and consistent navigation, providing alternative text for images, using readable fonts and appropriate color contrast, and ensuring that interactive elements can be easily accessed and operated using keyboard commands.
* Assistive Technologies: Exploring the use of assistive technologies in web accessible learning is crucial. Assistive technologies, such as screen readers, magnification tools, speech recognition software, and alternative input devices, can support learners with disabilities in accessing and interacting with online learning materials effectively.
* Multimedia Accessibility: Educational content often includes multimedia elements like videos, audio recordings, and interactive simulations. Ensuring that these multimedia resources are accessible involves providing captions, transcripts, and audio descriptions. It also includes using video players with accessibility features, such as adjustable playback speed or keyboard controls.
* Collaboration and Interaction: Web accessible learning should facilitate collaboration and interaction among learners, including those with disabilities. This can be achieved by incorporating accessible communication tools, such as chat functions, discussion boards with clear navigation, and providing multiple means of participation to accommodate different learning styles and abilities.
* User Testing and Feedback: Regular user testing and gathering feedback from learners with disabilities are essential to improve the accessibility of web-based learning resources. This can involve involving individuals with disabilities in the development and testing phases, conducting accessibility audits, and actively seeking input from learners to identify and address accessibility barriers.
* Training and Awareness: Educators, instructional designers, and content creators should receive training on web accessibility principles and techniques. Promoting awareness about the importance of accessibility in education can lead to a more inclusive learning environment, and ensure that accessibility remains a priority when developing and delivering online courses and materials.

By exploring these aspects of web accessible learning, educational institutions and content creators can make significant strides in providing

**2.3 Web Content Accessibility Guidelines**

The Web Content Accessibility Guidelines (WCAG) are the globally recognized standards for web accessibility. Developed by the World Wide Web Consortium (W3C), WCAG provides a set of guidelines and success criteria that website owners, developers, and designers should follow to make their web content accessible to individuals with disabilities. The current version of WCAG is WCAG 2.1, published in June 2018.

WCAG 2.1 is organized around four core principles, each of which has its own set of guidelines and success criteria:

* Perceivable: Web content should be presented in a way that users can perceive it using their available senses. This includes providing alternatives for non-text content (such as images and videos), ensuring that text is readable and adaptable, and making sure content is distinguishable by color-blind users.
* Operable: Web content should be operable by all users, regardless of their physical abilities. This principle covers aspects such as keyboard accessibility, providing enough time for users to interact with content, and avoiding content that could cause seizures or physical reactions.
* Understandable: Web content should be presented in a clear and understandable manner. This involves using plain language, organizing content in a logical manner, and providing assistance and guidance when necessary.
* Robust: Web content should be robust enough to be interpreted reliably by a wide range of user agents (e.g., web browsers and assistive technologies). This principle emphasizes the importance of using standard HTML, CSS, and JavaScript coding practices and avoiding reliance on technologies that may have accessibility limitations.

WCAG guidelines are categorized into three levels of conformance: A, AA, and AAA. Conformance at Level A is the minimum requirement, while Level AA conformance is considered the standard for most websites. Level AAA conformance includes the highest level of accessibility but may not be achievable for all types of content.

It's important to note that while WCAG provides comprehensive guidelines, web accessibility is an ongoing process. Regular accessibility audits, user testing, and continuous improvement are necessary to ensure websites maintain accessibility standards and accommodate the evolving needs of users with disabilities.

**2.3 Web Accessibility Evaluation Tools**

There are several web accessibility evaluation tools available that can assist in assessing the accessibility of websites and web applications. While each tool has its own unique features and capabilities, here is a comparison of some popular web accessibility evaluation tools:

WebAIM's WAVE (Web Accessibility Evaluation Tool):

WAVE is a free web accessibility evaluation tool that provides both online and browser extensions.

It highlights potential accessibility issues on a web page and offers detailed reports and suggestions for improvement.

WAVE offers a sidebar view that allows users to see the accessibility errors directly on the web page.

It supports evaluation against various accessibility guidelines, including WCAG (Web Content Accessibility Guidelines) 2.0 and 2.1.

AChecker:

AChecker is an online accessibility evaluation tool that supports the WCAG 2.0 and WCAG 2.1 guidelines.

It provides a comprehensive report indicating accessibility errors and suggests potential fixes.

AChecker supports batch processing, allowing users to evaluate multiple web pages simultaneously.

It offers an API for integration into development workflows or automated testing processes.

Axe (Deque Systems):

Axe is an open-source accessibility testing tool available as a browser extension or as part of various development frameworks.

It provides automated accessibility testing by scanning web pages for common issues and violations of WCAG guidelines.

Axe generates reports with issue details, suggestions for improvement, and relevant HTML markup.

It supports integration with testing frameworks like Selenium, Cypress, and Jest for automated accessibility testing.

Google Lighthouse:

Lighthouse is an open-source automated testing tool provided by Google, available in Chrome DevTools or as a command-line tool.

It offers a comprehensive evaluation of web page performance, accessibility, SEO, and best practices.

Lighthouse generates a detailed report with accessibility issues, including screenshots, affected elements, and guidance for resolving them.

It supports WCAG 2.1 guidelines and provides a score for each category, helping prioritize improvements.

Tota11y:

Tota11y is a JavaScript-based accessibility visualization toolkit developed by Khan Academy.

It can be integrated into web pages as a bookmarklet or added to development environments as a script.

Tota11y displays an overlay on web pages, highlighting accessibility issues and providing guidance on resolving them.

It offers various visualizations, including headings, alt text, color contrast, and more, to aid in understanding accessibility concerns.

It's important to note that while these tools can help identify common accessibility issues, manual testing and expert judgment are still necessary to ensure comprehensive accessibility. Additionally, no tool can guarantee 100% accessibility, as some issues require human interpretation and consideration of user experience.

**2.4 Project Goals**

These tools are designed to assist web developers, designers, and content creators in making their websites usable and navigable for people with diverse abilities and impairments. Here are some common goals of web accessibility tools:

* Alternative Text and Descriptions: Images, graphs, charts, and other visual elements are an integral part of web content. Web accessibility tools provide options for adding alternative text (alt text) to these elements, allowing individuals with visual impairments to understand and interpret them through assistive technologies.
* Color Contrast Enhancement: People with visual impairments, including color blindness, may struggle to perceive certain color combinations. Web accessibility tools help identify and improve color contrast, ensuring that text and graphical elements are legible for users with varying degrees of visual abilities.
* Text Resizing and Font Options: Web accessibility tools often include features that allow users to adjust the size and style of the text on a website. This accommodation caters to individuals with visual impairments, making it easier for them to read and comprehend the content.
* Providing Transcripts and Captions: Videos and audio content can present challenges for individuals with hearing impairments. Web accessibility tools support the inclusion of captions and transcripts, making multimedia content accessible to those who rely on visual or textual representation.
* Error Identification and Validation: Accessibility tools help identify potential accessibility issues during the development process by scanning websites for common errors and providing suggestions for improvements. This helps developers ensure compliance with accessibility standards and guidelines.

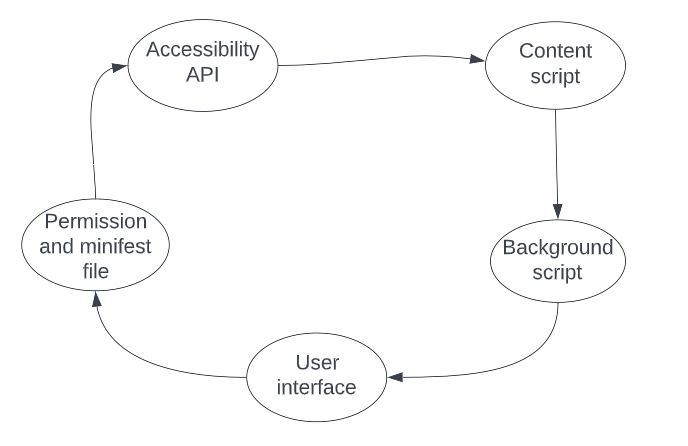
Overall, the primary goal of web accessibility tools is to promote inclusivity and equal access to digital content, empowering individuals with disabilities to engage with websites and web applications without barriers.

**Chapter 3**

---------------------------------------------------------------------------------------

**Methodology**

**3.1 Architecture of the Project:**



**Figure 1: Architecture of Proposed Extension**

Web-accessible Chrome extensions follow a specific architecture that enables them to enhance the accessibility of web content for users. This architecture involves various components and mechanisms to ensure that the extension functions effectively and seamlessly integrates with the user's browsing experience.

* **Content Scripts:** Content scripts are JavaScript files injected into web pages by the extension. They have access to the Document Object Model (DOM) of the web page and can manipulate it to enhance accessibility. Content scripts can modify the structure, style, and behavior of web content to make it more accessible, such as adding keyboard navigation support or modifying color contrast.
* **Background Scripts:** Background scripts serve as the backbone of the extension. They run independently in the background and handle tasks that require continuous monitoring or interaction with the browser. Background scripts can communicate with content scripts and coordinate their actions. They can also interact with the browser's APIs and handle events triggered by the user or the web page.
* **User Interface (UI) Components:** Web-accessible Chrome extensions often include user interface components that allow users to interact with the extension and customize its functionality. These UI components can be in the form of browser action buttons, context menu options, or pop-up windows. The UI components provide a way for users to activate the extension's features, adjust settings, or access additional information.
* **Accessibility APIs:** Chrome extensions can leverage the accessibility APIs provided by the browser to access and manipulate the accessibility properties of web content. These APIs allow the extension to retrieve information about elements on the web page, such as their role, state, and properties. By utilizing the accessibility APIs, the extension can analyze and enhance the accessibility of web content in a targeted manner.
* **Permissions and Manifest File:** To ensure security and user privacy, Chrome extensions require permissions to access specific browser features or web content. The extension's manifest file defines these permissions, along with other essential details such as the extension's name, version, icons, and supported URLs. The manifest file acts as a configuration file that informs the browser about the extension's capabilities and requirements.

By following this architecture, web-accessible Chrome extensions can seamlessly integrate with the browser environment and provide valuable accessibility enhancements to web content. They leverage various components, APIs, and storage mechanisms to modify web pages, interact with the browser, and offer a user-friendly interface for customization and control

**3.2 Requirement of the Project**

For creating this application some requirements are needed. We provide these requirements in three parts these are:

➢ Project Requirements

➢ Software Requirements

➢ Hardware Requirements

Project Requirements:

➢ Complete ERD (Entity Relationship Diagram)

➢ Complete Interaction Diagram

➢ Complete source code and run files for the front end and back end

➢Complete database re-generation script

Software Requirements:

➢ Framework: Express JavaScript.

➢ Language: JavaScript.

➢ Database: MongoDB.

➢ Web design: HTML, CSS.

➢ API: Third-Party API.

Hardware Requirements:

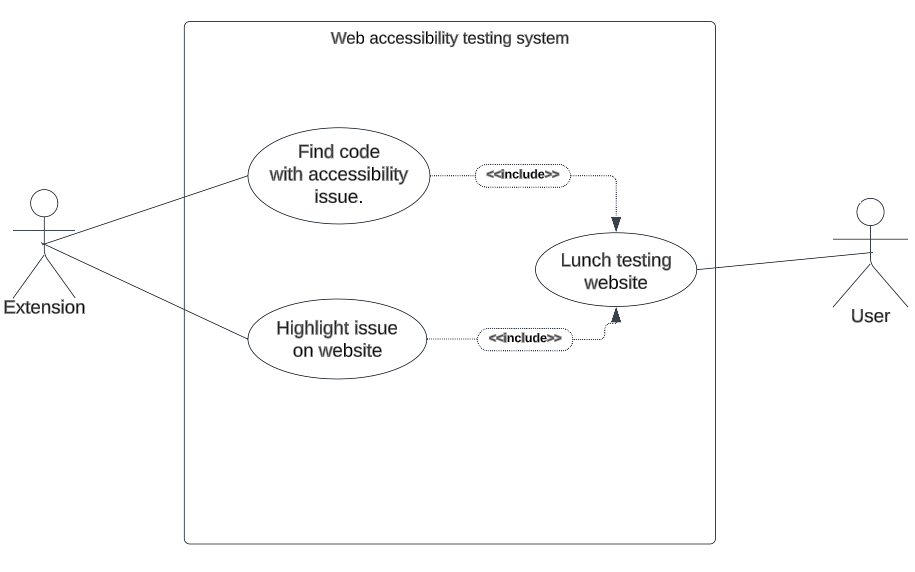
➢ CPU: Pentium 4 or upper version.

➢ RAM: 2GB or upper.

➢ HDD: As much as large so that stores a large amount of data.

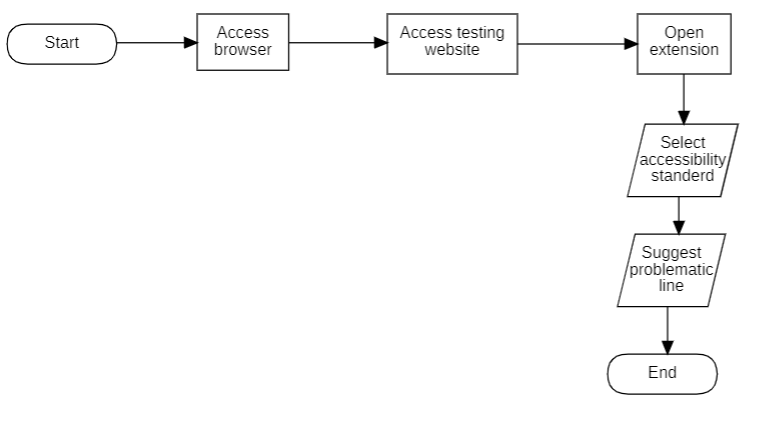
**3.3 Use Case Diagram**

The use case is used in system analysis to identify, clarify and organize system requirements. A use case diagram one kind of flow chart instinctive symbols represents the system elements which are like an actor, use case, the association includes relationship.



**Figure 2: Use Case Diagram**

**3.4 Flow Diagram**



* To initiate the process, the user would start by launching a web browser.
* The user will navigate to the website dedicated to testing the accessibility of websites.
* Subsequently, the user will activate the extension while being on the webpage of the aforementioned website.
* At that point, the user has the option to choose the desired accessibility standard.
* Subsequently, the extension scans the web page and detects any accessibility errors present.

**3.5 Technical Details**

Creating web accessible tools requires a combination of technical knowledge and the use of specific tools and technologies. Here are some commonly used technical tools for creating web accessible tools given below.

3.5.1 Visual Studio Code Editor

Visual Studio Code (VS Code) is a popular source code editor developed by Microsoft. While it is not specifically designed for web accessibility testing, you can utilize various extensions and features within VS Code to support your development process and ensure accessibility in your web projects.

3.5.2 Express JavaScript Framework

3.5.3 Postman

Postman is a widely used collaboration platform and API development tool that allows developers to design, test, and document APIs. While Postman itself is not specifically geared towards web accessibility testing, it can be leveraged in certain ways to support accessibility testing of web APIs. Here are a few ways you can utilize Postman for web accessibility testing:

* API Response Validation: Postman allows you to send API requests and inspect the responses received. As part of your accessibility testing, you can validate the API responses to ensure they conform to accessibility standards, such as returning accessible data formats or adhering to accessibility guidelines for API responses.
* Automated Testing with Pre-request Scripts: Postman allows you to execute pre-request scripts before sending an API request. You can utilize this feature to automate accessibility checks by including accessibility testing libraries or tools, such as aXe or pa11y, in your pre-request scripts. These tools can assess the accessibility of the API responses and provide feedback on any accessibility violations.
* API Documentation: Postman provides functionality to document APIs, including describing endpoints, parameters, and responses. When documenting APIs, you can consider including information about the accessibility of the API, such as any accessibility features it supports, the accessibility of the data it returns, or any accessibility considerations for clients consuming the API.
* Collaboration and Reporting: Postman offers collaboration features that allow you to share API collections, test suites, and reports with team members or stakeholders. You can leverage these features to collaborate on accessibility testing efforts, share findings, and track progress.

While Postman may not directly test the accessibility of a web application's user interface, it can be a valuable tool for verifying the accessibility of the API layer and facilitating collaboration on accessibility testing within the development team. To ensure comprehensive accessibility testing, it's important to complement API testing with other tools and techniques focused on assessing the accessibility of the user interface.

3.5.4 Manifest File

The manifest file is a crucial component when developing an extension for web browsers like Chrome. It is a JSON (JavaScript Object Notation) file that outlines important details about the extension and its functionality.

When creating an extension, the manifest file serves as a roadmap for the browser, providing information on how the extension should be loaded, displayed, and interacted with. It also defines permissions, resources, and dependencies required by the extension.

3.5.5 Languages

JavaScript, HTML, and CSS are the three core technologies used for building and designing websites and web applications. Each language serves a specific purpose and plays a crucial role in creating dynamic and interactive web experiences. Here's a brief overview of JavaScript, HTML, and CSS:

JavaScript:

JavaScript is a high-level programming language that adds interactivity and dynamic functionality to web pages. It is primarily used for client-side scripting, allowing developers to manipulate and modify the content of web pages, handle user interactions, make API requests, perform form validations, and more. JavaScript can also be used for server-side programming (with Node.js) to build web servers and perform server-side operations.

HTML:

HTML (Hypertext Markup Language) is the standard markup language used for structuring and presenting content on the web. It defines the elements and their attributes that make up the structure of a web page. HTML tags are used to represent headings, paragraphs, images, links, forms, tables, and other elements on a web page. It provides the basic framework and semantics for organizing and displaying content.

CSS:

CSS (Cascading Style Sheets) is a style sheet language that is used to describe the presentation and appearance of HTML elements. It allows developers to define colors, fonts, layouts, spacing, and other visual aspects of a web page. With CSS, you can control the design and layout of multiple web pages from a single style sheet, making it easier to maintain consistency and achieve a visually appealing and responsive design.

Together, JavaScript, HTML, and CSS form the foundation of modern web development. JavaScript provides interactivity and functionality, HTML defines the structure and content, and CSS handles the visual presentation. Web developers typically use a combination of these languages to create engaging and user-friendly web experiences.

**3.6 Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Browser | Yes | No |
| 1 | Chrome | 🗸 | 🗴 |
| 2 | Mozilla Firefox | 🗴 | 🗸 |
| 3 | Microsoft Edge | 🗴 | 🗸 |

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Input(WCAG Standard 2.0) | Pass | Fail |
| 1 | WCAG2A | 🗸 | 🗴 |
| 2 | WCAG2AA | 🗸 | 🗴 |
| 3 | WCAG2AAA | 🗸 | 🗴 |

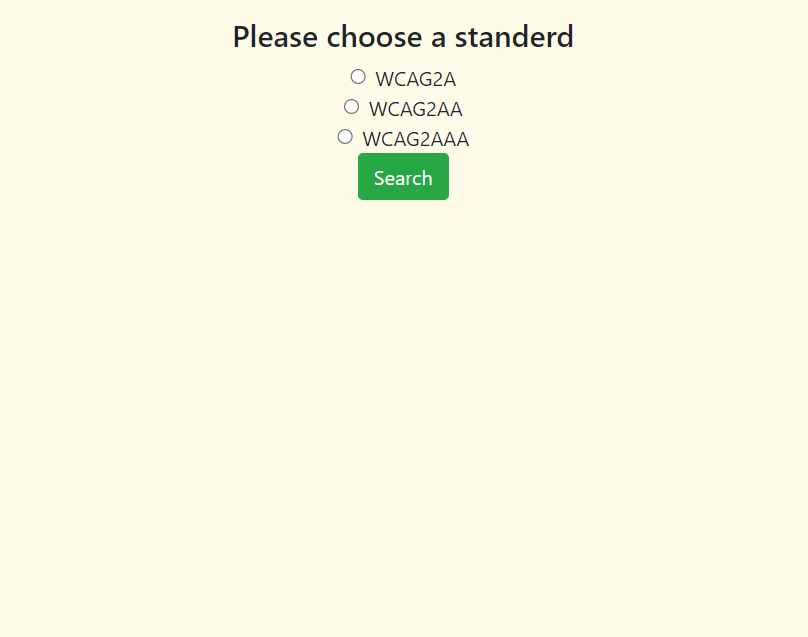
**Chapter 4**

---------------------------------------------------------------------------------------

**Project Features and Functionalities**

4.1 Proposed Extension User Interface

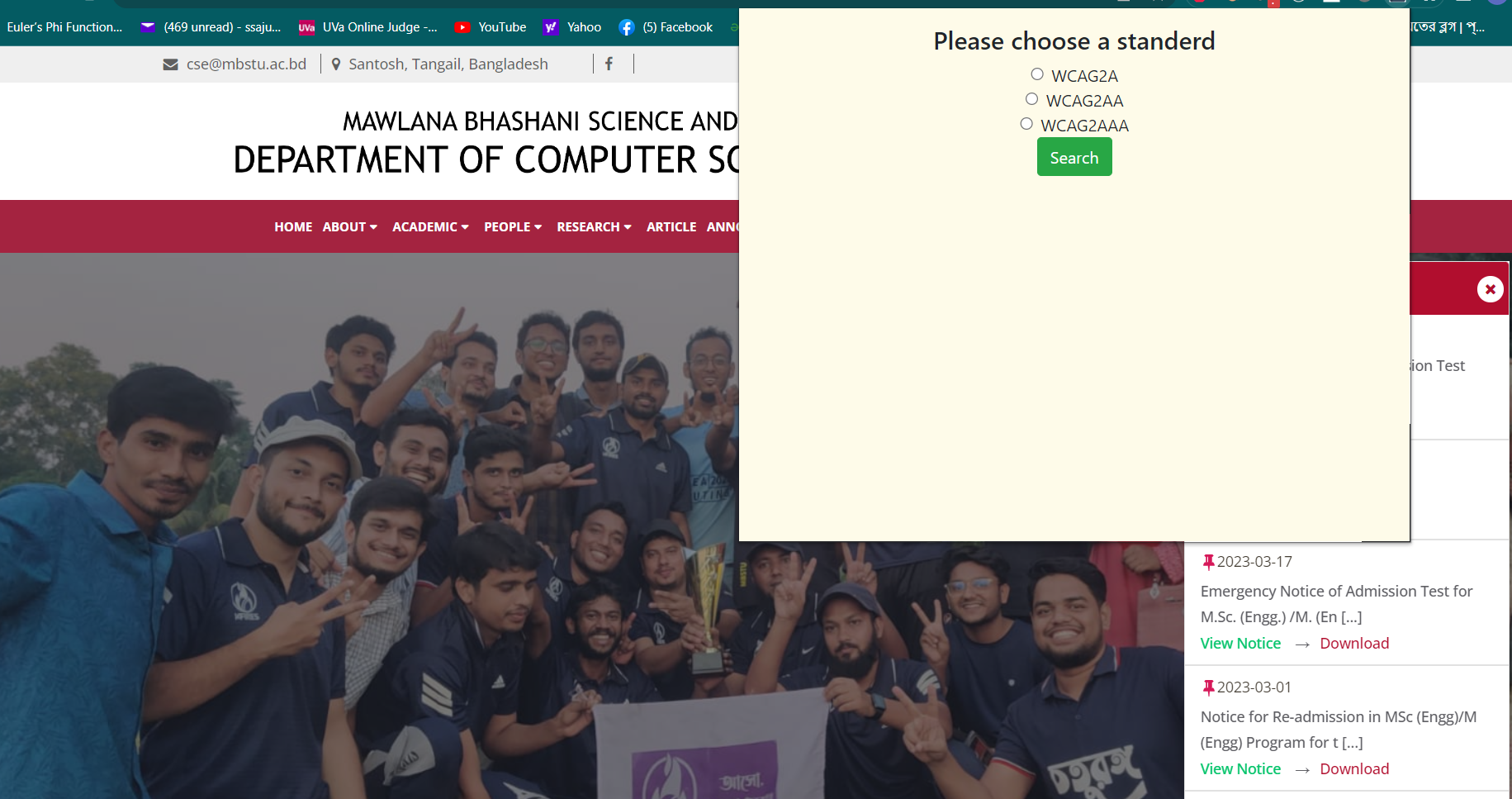
Here provided an overview of the user interface of the proposed web accessible extension that was developed and evaluated.



**Figure 4: Proposed Extension User Interface**

**4.2 Proposed Extension open on sample website**

The user will activate the extension while being on the webpage of the aforementioned website. At this point the user has the option to choose the desired accessibility standard.

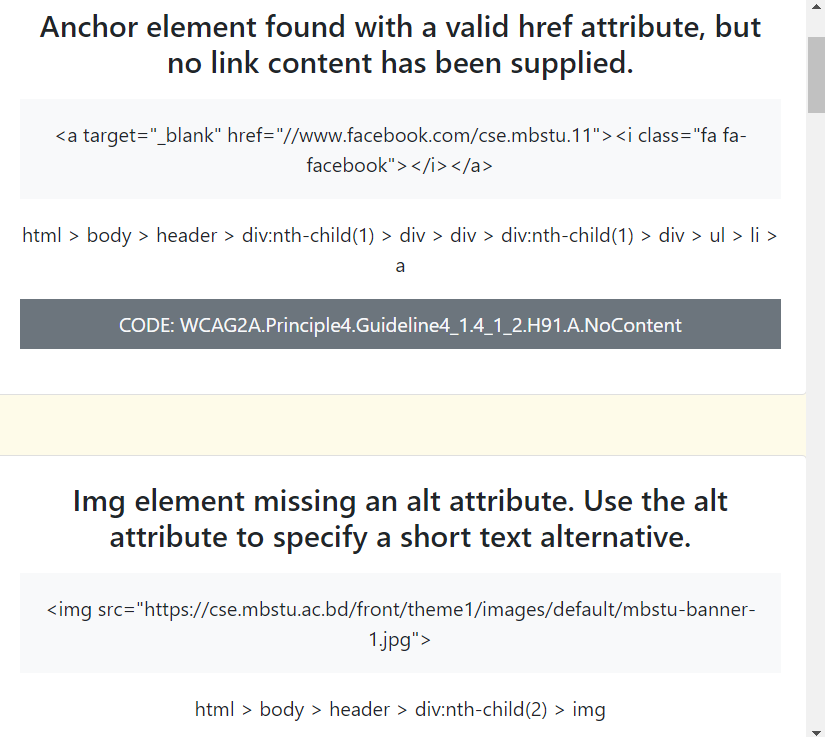


**Figure 5: Proposed Extension open on sample website**

**4.3 Results**

When the user selects an accessibility standard, the extension scans the webpage to identify any existing accessibility errors. The extension presents errors in two ways: one by indicating errors in the code, and the other by identifying the specific content element on the web page where the error occurs.

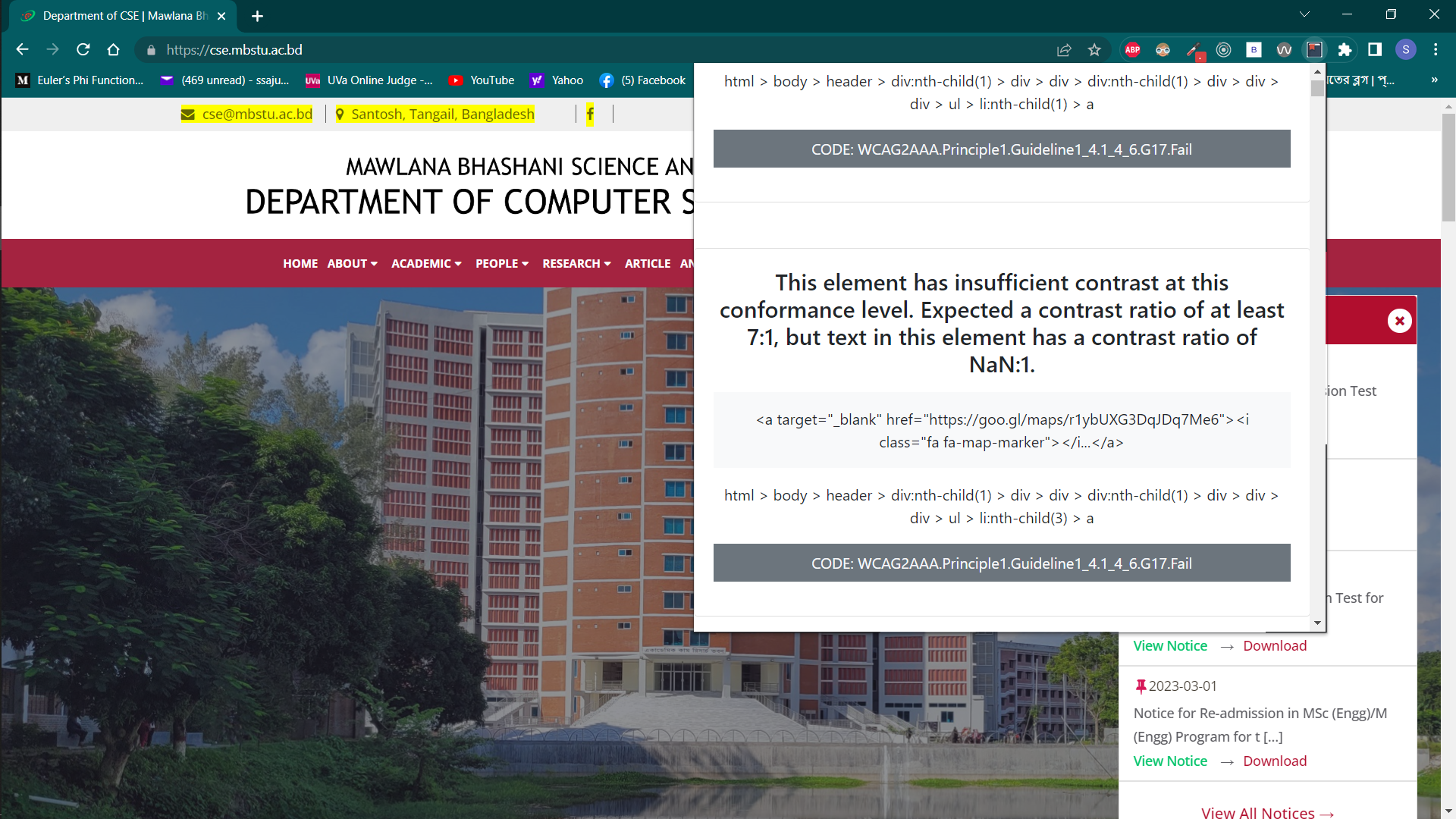
Coding Line Error:



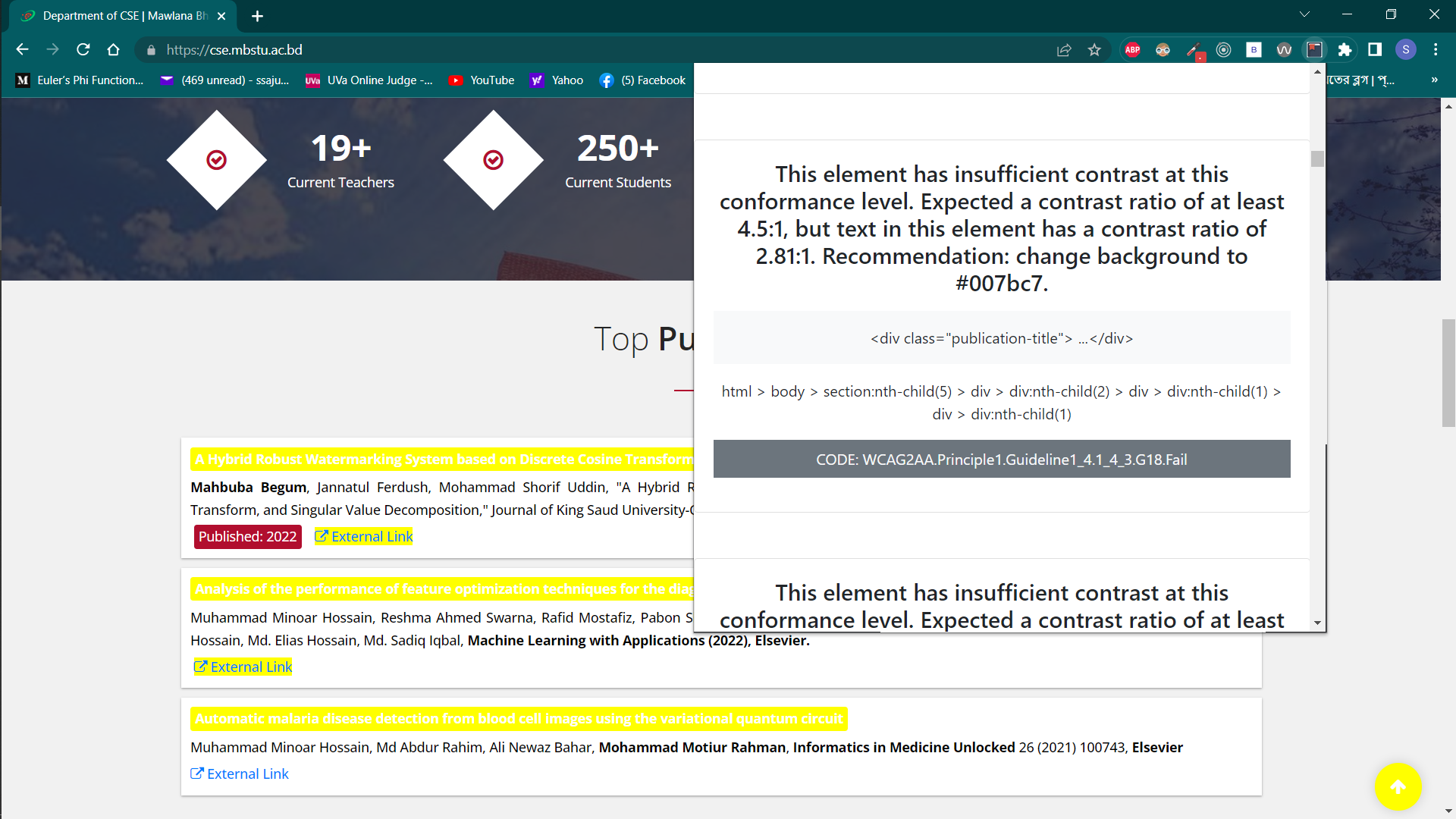
**Figure 6: Coding Line Error**

Content Element error on Web Page:

Below are some examples of the error element on a web page, indicated by the presence of a yellow mark.



**Figure 7: Content Element error on Web Page (Sample 1).**



**Figure 8: Content Element error on Web Page (Sample 2).**

**Chapter 5**

**----------------------------------------------------------------------**

**Conclusion**

In conclusion, web accessibility tools play a crucial role in ensuring that websites are accessible to individuals with disabilities. These tools provide a range of features and functionalities that help improve the overall accessibility of web content, making it easier for people with disabilities to navigate, perceive, and interact with websites. Web accessible Chrome extensions in enhancing the web browsing experience. The research provides valuable insights into the benefits and challenges associated with such extensions, paving the way for future advancements in this area. The findings contribute to the broader field of web accessibility and user-centered design, offering recommendations for the development of more effective and user-friendly web accessible Chrome extensions. This Extension is mainly designed for the Chrome browser. This Chrome Extension find out error, warning, alert in the web page. And show the error line in HTML code.