Index

1. Introduction

An HTML online compiler is a web-based tool that allows users to create, edit, and preview HTML (Hypertext Markup Language) code directly within a web browser. It provides a user-friendly interface for writing and testing HTML code without the need for any specialized software or local development environments Users can easily write HTML code for web pages, including elements like headings, paragraphs, images, links, forms, and more The online compiler typically offers a code editor that allows users to make real-time changes to their HTML code.

Objective:

The objective of an HTML online compiler is to provide users with a convenient and accessible platform for creating, editing, and testing HTML code.

Purpose:

The purpose of an HTML online compiler is to provide a versatile, user-friendly platform for creating, testing, and learning HTML, streamlining web development, and promoting collaboration and accessibility in the field of web development and design.

Project Scope

Downloadable Compiled Code: Users will have the capability to download the compiled code of their programming projects in various languages, including but not limited to C++, Java, Python, and JavaScript.

2.Feasibility Study

A feasibility study is conducted to assess the practicality and viability of an HTML online compiler, which is a web-based tool for creating, editing, and previewing HTML code within a web browser. This study will evaluate the technical, economic, operational, and scheduling feasibility of developing and maintaining such a tool.

1. Technical Feasibility:

a. Development Tools:

The development of an HTML online compiler requires standard web development technologies, including HTML, CSS, and JavaScript Integration with code editors and live preview modules is technically feasible using libraries and frameworks.

b. Browser Compatibility:

The tool must be compatible with a wide range of web browsers to ensure accessibility. Testing across multiple browsers is technically feasible.

c. Real-Time Editing:

Implementing a code editor that allows real-time code editing is technically feasible, as there are well-established libraries and APIs available.

2. Economic Feasibility:

a. Development Costs:

The cost of developing an HTML online compiler is relatively low, primarily involving web development and hosting expenses Costs may include developer salaries, server hosting, and software licenses for any third-party tools used.

3. Operational Feasibility:

a. User Interface:

The tool's user interface is designed to be user-friendly, making it operational for users of varying skill levels.

b. Support and Maintenance:

Ongoing support and maintenance are feasible and necessary for bug fixes, updates, and ensuring compatibility with evolving web standards.

4. Scheduling Feasibility:

a.Development Timeline:

The development timeline depends on the scope and features of the online compiler. An initial version with essential features can be launched relatively quickly, with additional features added over time.

b. Testing and Quality Assurance:

Extensive testing and quality assurance are essential but can be integrated into the development schedule Scheduling regular updates based on user feedback is feasible and enhances the tool's usability.

3.System Analysis:

The purpose of an "Online Live Code Editor" is to provide a web-based platform that allows developers and learners to write, edit, and execute code in real-time. Here are the key purposes and functionalities of an online live code editor:

- Code Editing
- Syntax Highlighting
- Code Switching
- Live Preview
- Web Development Playground

3.1 Existing System:

It's an Existing based on that several modifications has done to that existing system various features like code highlighting syntax highlighting live result preview and more modifications has been done to that system.

3.2 Proposed System:

Online live code editor is an proposed project which includes various features for better experience ,usage, practicing web based applications and user friendly interface main functions and features ,modules are listed below

- Syntax Highlighting
- Code Switching
- Live Preview
- Error Handling

3.3 Software Requirements Specification Document

Purpose

The purpose of an "Online Live Code Editor" is to provide a web-based platform that allows developers and learners to write, edit, and execute code in real-time An HTML online compiler is a web-based tool that allows users to create, edit, and preview HTML (Hypertext Markup Language) code directly within a web browser. It provides a user-friendly interface for writing and testing HTML code without the need for any specialized software or local development environments

Write HTML Code: Users can easily write HTML code for web pages, including elements like headings, paragraphs, images, links, forms, and more Edit Code in Real Time: The online compiler typically offers a code editor that allows users to make real-time changes to their HTML code Online Live Code Editor : A Versatile Web-Based Development Tool Here are the key purposes and functionalities of an online live code editor:

- Code Editing
- Syntax Highlighting
- Code Switching
- Live Preview
- Web Development Playground

Document Conventions

Font and Text Styling:
Bookman Old Style & font Size should be 12
Line Spacing is 1.5
Headings are centered, bold & font size should be 14
margins are one-inch on all sides for printed documentation

Abbreviations:

- HTML (Hypertext Markup Language): HTML is a markup language used to structure and define the content of web pages, using elements and tags to create the layout and hierarchy of text, images, and other media.

- CSS (Cascading Style Sheets): CSS is a style sheet language that defines the presentation and design of HTML elements, allowing web developers to control the appearance, layout, and formatting of web content.

- JavaScript (JS): JavaScript is a high-level, interpreted programming language used for adding interactivity and dynamic behavior to web pages, enabling features like form validation, animations, and real-time updates.

Product Scope

The HTML code editor is an online, live platform for web developers and learners to edit, experiment with, and run HTML, CSS, and JavaScript code. It offers an interactive environment, real-time coding, syntax highlighting, event handling, and code compilation, making it useful for code testing, prototyping, and learning web development concepts.

- Interactive Learning Environment
- Efficient Code Compilation
- Educational and Productive Tool

Downloadable Compiled Code: Users will have the capability to download the compiled code of their programming projects in various languages, including but not limited to C++, Java, Python, and JavaScript.File Upload Feature: A file upload functionality will be integrated into the platform, allowing users to upload source code files and data files for their coding projects.

Product Perspective

The HTML code editor is a standalone online tool for editing HTML, CSS, and JavaScript code. It can be integrated into larger projects or educational platforms, providing real-time coding and testing, enhancing development workflow efficiency.

➔ Educational Website Integration

Product Functions

The HTML code editor is an interactive online tool that allows users to edit and preview HTML, CSS, and JavaScript code in real-time, promoting hands-on learning and rapid prototyping. It provides immediate feedback for debugging and visualization, making it adaptable to various web development scenarios. As Some of The Functionalities listed below:

- Code Editing

- Real-Time Preview
- Code Compilation
- Switching Between Languages
- Syntax Highlighting etc.…

User Classes and Characteristics

➢ The code editor accommodates a wide range of users, from novice web developers seeking hands-on learning to experienced developers in need of a rapid prototyping tool.

➢ It's a valuable resource for educational institutions and online learning platforms, enabling interactive web development instruction.

➢ Overall, this code editor caters to a diverse user base, fostering both learning and productivity in the web development field.

User classes and their characteristics are important considerations when designing and developing an HTML online compiler. Understanding the different types of users and their needs helps tailor the tool to provide a better user experience. Here are some user classes and their characteristics:

1. Novice Users:

Characteristics:

Limited or no prior experience in HTML coding Seek simplicity and user-friendly interfaces.

Needs:

Clear and easy-to-understand instructions Basic templates and examples for quick learning.

2.Intermediate Users

Characteristics:

Some experience with HTML but looking to improve their skills. Willing to experiment and learn.

Needs:

Advanced code editing features, such as syntax highlighting Access to a code library with examples and snippets.

3. Advanced Users

Characteristics:

Proficient in HTML and web development May use the tool for testing and quick prototyping.

Needs:

Code editor with advanced features (auto-complete, live preview) Support for custom libraries and external resources.

4. Educators and Trainers

Characteristics:

Teach HTML and web development Use the tool for instructional purposes.

Needs:

Classroom management features, including student progress tracking Access to teaching materials and lesson plans.

5. Web Developers and Designers:

Characteristics:

Professionals creating websites and web applications Use the tool for quick prototyping and code testing.

Needs:

Advanced code editor with code version control Integration with web development frameworks and libraries.

6. Quality Assurance and Testing Teams

Characteristics:

Use the tool for testing web applications Require real-time code changes for testing purposes.

Needs:

Integration with testing tools and access to debugging features Collaboration capabilities for reporting issues.

7. Freelancers and Small Businesses

Characteristics:

Independent web developers, freelancers, or small web agencies Use the tool for project development and client work.

Needs:

Project management features to organize and manage client Collaboration and sharing tools for working with clients.

8. Students and Coding Enthusiasts

Characteristics:

Students learning web development or coding enthusiasts Use the tool for personal projects and learning.

Needs:

Access to a supportive community or forum for asking questions and sharing knowledge Integration with educational resources and tutorials.

Understanding the characteristics and needs of these user classes is crucial for tailoring the HTML online compiler to provide a positive and productive experience for a diverse range of users. It helps in designing features, user interfaces, and support mechanisms that cater to the specific requirements of each group.

Operating Environment

- The operating environment for the provided HTML code is web-based and browser-dependent. Users can access and utilize the code editor through modern web browsers on various operating systems, including Windows, macOS, and Linux.
- It is compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. The code editor relies on client-side technologies, including HTML, CSS, and JavaScript, which execute within the user's browser. Additionally, it may require an internet connection to load external resources such as the Ace.js library and any additional scripts.

Design and Implementation Constraints

The HTML code is limited in browser compatibility and internet connectivity, necessitating security and scalability. It's primarily used for frontend development, requiring customization and accessibility. Regular maintenance and updates are necessary to keep the code editor up-to-date. The below constraints are involved

- Browser Compatibility
- Internet Connectivity
- Resource Consumption
- Code Version Control
- Customization Complexity

User Documentation

Tittle: Online live Code Editor

User guide. This document will help you make the most of this powerful tool for web development and experimentation. Whether you're a beginner or an experienced code Editor, this guide will walk you through the key features and functionalities of the code editor.

Table of Contents:

Assumptions and Dependencies

Assumptions:

- ➔ Internet Connectivity: Users are assumed to have a reliable internet connection to access external resources, including the Ace.js library and any other scripts hosted remotely.
- ➔ Modern Browsers: The code assumes that users will access the code editor using modern web browsers that support HTML5, CSS3, and JavaScript features, ensuring compatibility and functionality.
- ➔ Basic Web Development Knowledge: Users are assumed to have a basic understanding of HTML, CSS, and JavaScript, as the code editor serves as a development tool rather than a tutorial.

Dependencies:

- ➔ Ace.js Library: The code editor relies on the external Ace.js library for syntax highlighting and code editor functionality. This library must be loaded from a remote source (e.g., a CDN) for the code editor to work correctly.

- ➔ External Stylesheet: The code uses an external stylesheet (main.css) for styling purposes. This stylesheet must be available and properly linked for the code editor's visual presentation.
- ➔ JavaScript Files: The code references external JavaScript files, including "app.js." These files contain the code editor's functionality and must be accessible and correctly referenced for the code editor to operate.
- ➔ Browser Environment: The code editor is dependent on the user's web browser environment, including its capabilities and settings, which can affect performance and functionality.

Understanding these assumptions and dependencies is crucial for users and developers to effectively utilize and customize the provided HTML code in Online Live Code Editor

External Interface Requirements

User Interfaces

The provided Online Live Code Editor offers a user-friendly and interactive user interface (UI) that facilitates code editing and real-time preview. Here are the key components of the user interface:

- Header Section
- IDE (Integrated Development Environment) Section
- Description Section
- Footer Section
- Buttons and Interactive Elements
- Code Panels
- Result Panel

Hardware Interfaces

The provided Online Live Code Editor primarily operates in a web-based environment and does not have direct hardware interfaces. However, it relies on the user's hardware, particularly their computer or mobile device, to access and interact with the code editor. The hardware interfaces are primarily those of the user's device, including:

- Input Devices
- Output Devices
- System Resource

Software Interfaces

The provided Online Live Code Editor interacts with several software components and interfaces to deliver its functionality:

- Web Browsers
- Ace.js Library
- External Stylesheet
- Operating System
- CDN (Content Delivery Network)

Communications Interfaces

The provided Online Live Code Editor primarily relies on internet-based communication interfaces to function effectively. These interfaces facilitate the exchange of data and resources between the user's device and external sources. Key communications interfaces include:

- HTTP (Hypertext Transfer Protocol)
- CDN (Content Delivery Network)
- User Input and Browser Events
- Internet Connectivity

System Features

System Features of an "Online Live Code Editor" is to provide a web-based platform that allows developers and learners to write, edit, and execute code in real-time. Here are the key purposes and functionalities of an online live code editor

- Code Editing Panels
- Syntax Highlighting
- Code Switching
- Live Preview
- Web Development Playground

System Feature 1

Code Editing Panels:

HTML, CSS, JavaScript: The code editor provides dedicated panels for editing HTML, CSS, and JavaScript code. Users can switch between these panels to work on different aspects of web development in one integrated environment.

System Feature 2

Real-Time Code Preview:

The code editor offers a real-time code preview feature. Users can see the compiled result of their code changes instantly in a separate iframe, making it easier to visualize and debug their web projects.

System Feature 3

Syntax Highlighting

Utilizes the Ace.js library to provide syntax highlighting for HTML, CSS, and JavaScript code. Syntax highlighting improves code readability and helps users identify errors more efficiently

System Feature 4

Code Documentation:

Includes code comments that provide explanations and context within the HTML, CSS, and JavaScript sections, aiding users in understanding and modifying the code.

Other Nonfunctional Requirements

Performance Requirements

Performance requirements for the provided Online Live code Editor, which specify how the code editor should perform in various aspects, include:

Responsiveness:

- ➔ Requirement: The code editor must respond to user interactions within 200 milliseconds or less.

- ➔ Rationale: Ensures a seamless and interactive user experience, particularly when editing or interacting with code.

Code Compilation Time:

- ➔ Requirement: The code editor should compile and render code changes in the "Result" panel in under 500 milliseconds for typical code snippets.
- ➔ Rationale: Swift code compilation provides immediate feedback to users, aiding in debugging and enhancing productivity

Safety Requirements

Safety requirements for the provided Online Live code Editor primarily focus on ensuring user and system safety when using the online code editor. While web-based code editors are generally low-risk, it's essential to consider potential safety concerns:

Code Execution Safety:

- ➔ Requirement: The code editor must prevent the execution of malicious or harmful code within the user's browser environment.
- ➔ Rationale: Ensures that user-generated code cannot compromise the user's device or data.

Error Handling:

- ➔ Requirement: Implement robust error handling and reporting mechanisms to provide clear and non-deceptive error messages in case of code execution or compilation errors.
- ➔ Rationale: Helps users understand and troubleshoot issues without confusion.

Security Requirements

Security requirements for the provided Online Live code Editor, which focus on safeguarding the code editor and user data, include:

Secure Code Execution:

- ➔ Requirement: The code editor must employ sandboxing techniques to execute user-generated code within a controlled and isolated environment, preventing access to system resources or other sensitive data

Secure Communication:

- ➔ Requirement: Use HTTPS to encrypt data transmission between the user's browser and external servers, ensuring that data exchanged during code compilation or resource retrieval is protected from eavesdropping.

Software Quality Attributes

The below software quality attributes collectively define the expected characteristics and performance standards of the code editor, ensuring it meets user expectations and delivers a valuable and dependable web development tool.

- Usability
- Reliability
- Performance
- Compatibility
- Scalability
- Security
- Maintainability
- Documentation

Business Rules

Business rules for the provided HTML code, which govern the operation and behavior of the online code editor, include:

Code Execution Rules:

- ➔ Rule: User-generated code executed within the code editor's environment must not access or modify external resources, including files, databases, or external servers

Code Sharing Rules:

- ➔ Rule: Users may share their code or projects via generated URLs or other means, but the code editor itself should not store or maintain a public repository of shared code.

Certainly, here's the revised appendix without the star symbols:

Appendix A: Additional Resources

1. HTML Cheat Sheet: A one-page reference guide with commonly used HTML tags and attributes.

2. CSS Cheat Sheet: A one-page reference guide with common CSS properties and values.

3. JavaScript Quick Start Guide: A brief introduction to JavaScript for users who are new to the language.

W3C HTML and CSS Specifications:

Links to the official HTML and CSS specifications from the World Wide Web Consortium (W3C).

Appendix C: Troubleshooting

1. Common Error Messages: A list of common error messages users may encounter and their explanations.
2. Troubleshooting Guide: Step-by-step instructions for resolving common issues with the HTML online compiler.

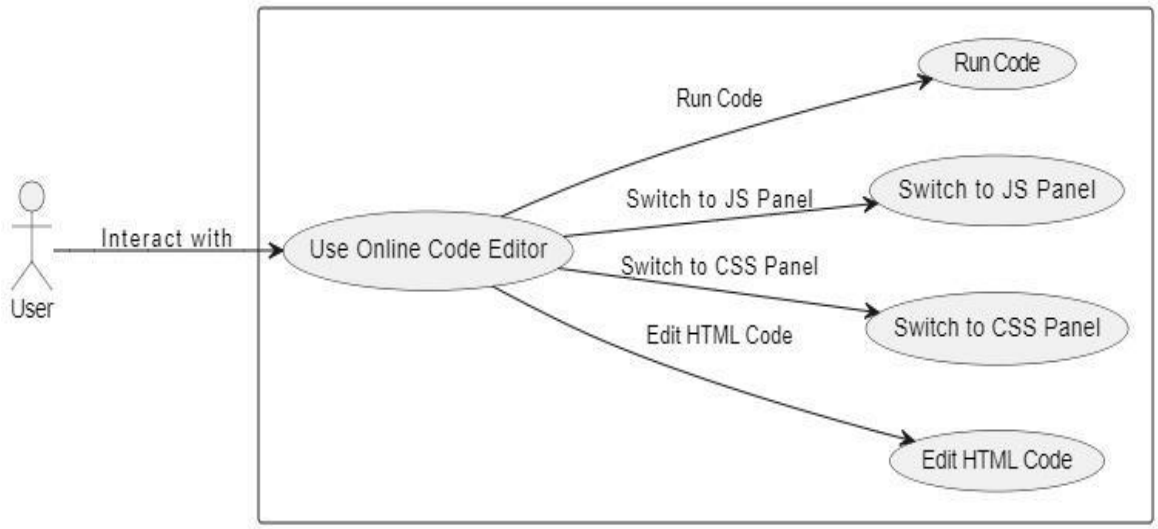
Appendix D: Technical Details

1. Server and Technology Stack: Technical details about the server and technology stack used to build the compiler.
2. API Documentation: If applicable, documentation for any APIs that developers can use to extend the functionality of the tool.

Appendix E: User Feedback Form

1. A user feedback form that allows users to provide input, report issues, or make suggestions for improvements The appendix provides users with supplementary information and resources to enhance their experience with the HTML online compiler.

3.4 Use-Case Diagram:



Use-Case Description:

Use Case: Editing and Running Code

Primary Actor: User

Goal: The user can edit and run HTML, CSS, and JavaScript code in an online code editor.

Preconditions:

The web page with the online code editor is loaded.

Main Success Scenario:

- The user accesses the online code editor.
- The user sees a title and description in the header.
- The user is presented with four buttons: "HTML," "CSS," "JS," and "Result."
- The user clicks the "HTML" button to edit HTML code. The HTML code is displayed in the HTML panel.
- The user clicks the "CSS" button to edit CSS code. The CSS code is displayed in the CSS panel.
- The user clicks the "JS" button to edit JavaScript code. The JavaScript code is displayed in the JavaScript panel.
- The user clicks the "Result" button to view the result of the code execution. The result is displayed in an iframe.
- The user can switch between code panels and view the result as needed.
- Postconditions:
- The user can edit and run HTML, CSS, and JavaScript code in the online code editor.

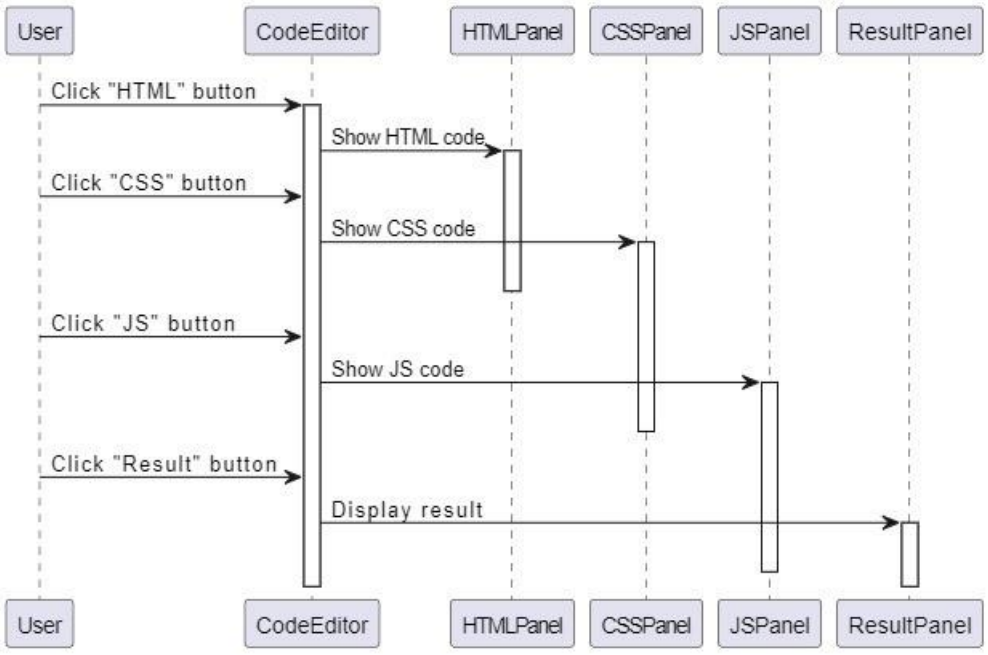
4. System Design

System design typically consists of several key components and considerations. It is a crucial phase in software development that involves designing the architecture and structure of a system or application. The following are the key aspects

- sequence diagrams
- activity diagrams
- state chart diagrams

4.1.2 Sequence Diagram:

A Sequence Diagram in UML depicts interactions between objects or components in a system over time, showing the order of messages and the flow of control. In the context of your HTML code, a Sequence Diagram could illustrate the sequence of interactions between the user and the code editor, showcasing how the user's actions trigger functions and events in the code editor and result in changes in the displayed code or output.



5. Implementation

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Online, Live Code Editor</title>

  <meta name="description" content="This is an online code editor for html css and javascript. ">

  <meta name="robots" content="index, follow">

  <meta name="author" content="Aadi Kommula">

  <meta name="keywords" content="Live code editor, online code editor, live html editor, live css editor, live javascript editor">

  <link rel="stylesheet" href="main.css" />

</head>



<body>

  <div class="header-wrap">

    <h1>Live, Online Code Editor</h1>
```

```
    <h4>Compile HTML, CSS and JavaScript on the web</h4>
  </div>
  <div id="ide-parent">
    <div id="button-wrapper">
      <button onclick="switchPanel(0)">HTML</button>
      <button onclick="switchPanel(1)">CSS</button>
      <button onclick="switchPanel(2)">js</button>
      <button onclick="runEdit(3)">Result</button>
    </div>
    <div id="ide-container">
      <div class="panel-wrapper">
        <div id="html">
&lt;!--predefined html, but editable--&gt;


&lt;div class="circle"&gt;
         click me!
&lt;/div&gt;


        </div>
```

```
      </div>
      <div class="panel-wrapper">
        <div id="css">
/*predefined CSS, but editable*/


.circle {
          display: flex;
          align-items: center;
          justify-content: center;
          width: 200px;
          height: 200px;
          border-radius: 50%;
          background: crimson;
          margin: auto;
          margin-top: 50px;
          transition-duration: 300ms;
}
        </div>
      </div>
```

```
      <div class="panel-wrapper">
        <div id="js">
//predefined JavaScript, but editable

var circle = document.querySelector(".circle");
var bool = false;

circle.addEventListener('click', function() {
          if (!bool) {
                    circle.style.background = "coral";
                    bool = true;
          } else {
                    circle.style.background = "crimson";
                    bool = false;
          );
        </div>
      </div>
      <div class="panel-wrapper">
        <iframe id="result"></iframe>
```

```
      </div>
    </div>
  </div>
  <div class="description">
    <p>This code editor uses Ace for syntax highlighting and other text
      editing features. Addition or changes to
      the code is compiled on the fly when the result tab is clicked.</p>
          </div>
  <div class="footer">
    <div class="inner-wrap">
      <p>&#169; <span></span> abc</p>
      <div class="icons-wrap">          </div>
    </div>  </div>
  <script       src="https://cdnjs.cloudflare.com/ajax/libs/ace/1.4.11/ace.js"
type="text/javascript"
   charset="utf-8"></script>  <script src="app.js"></script>
</body>


</html>
```

6. Testing

About Testing:

Testing is a process that requires more effort than any other software engineering activity. Testing in a set of activities that can be planned in advance and conducted systematically. This may lead to having many undetected errors in the system being developed. Hence performing testing by adopting systematic strategies is very much essential during development of software.

White Box Testing:

White Box Testing is defined as the testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application improving design and usability, strengthening security. White box testing is also known as Clear Box testing, Open Box testing. Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is usually performed by developers. It is one of two parts of the "Box Testing" approach to software testing.

Its counterpart, Black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing The term "White Box" was used because of the see-through box concept. The clear box or White Box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

White box testing involves the testing of the software code for the following:

- The Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output

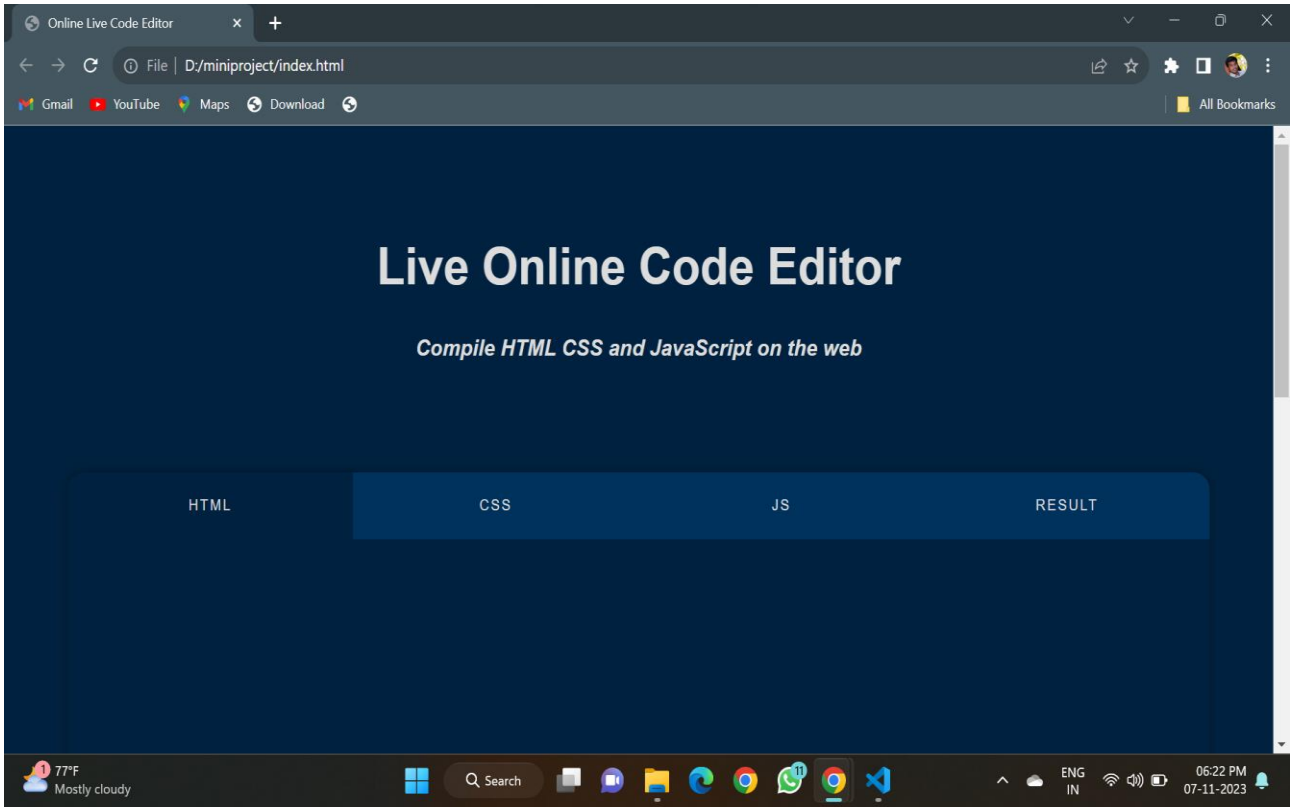
- functionality of conditional loops

Black box testing:

Black box testing or functional testing means testing the software for the outputs when specified inputs are provided. It is usually performed to see if the software meets the user's requirements. There's no focus on what goes behind running that software. In this testing use who are involved in the project they don't have knowledge about the structure of the project.

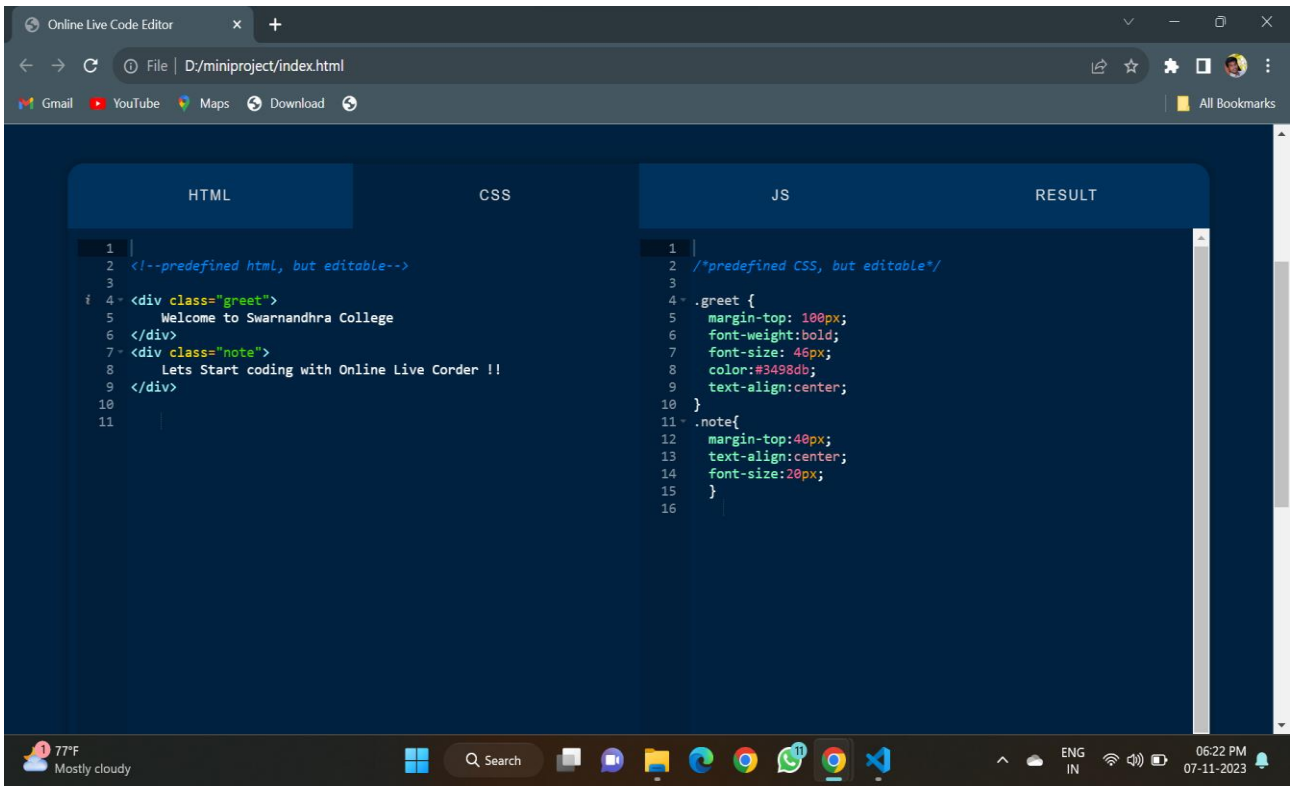
7.Results and Discussion

Here's the main page of the live code editor the layout looks like as:

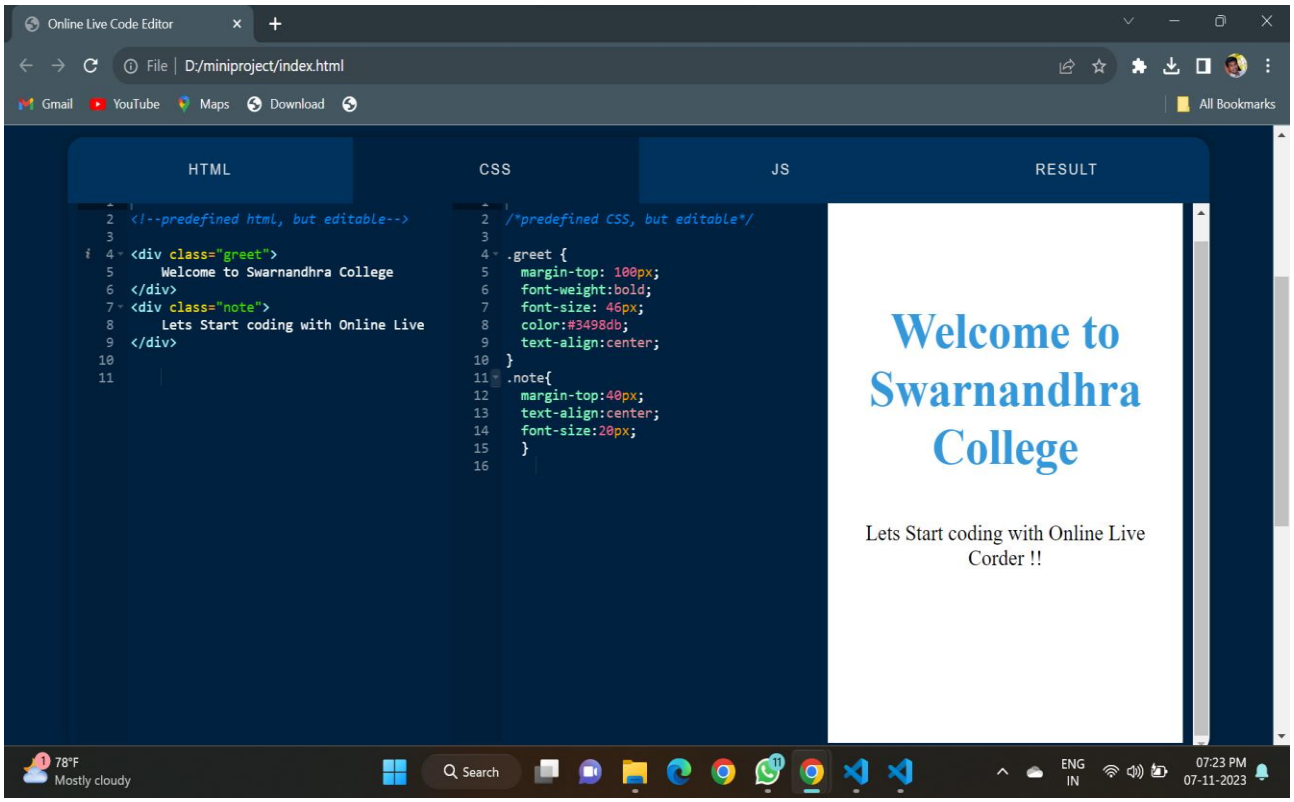


Here’s the scenario after Entering into the live code editor main pagethe below panels will be displayed as follow as:

- Html panel
- CSS Panel
- JS panel

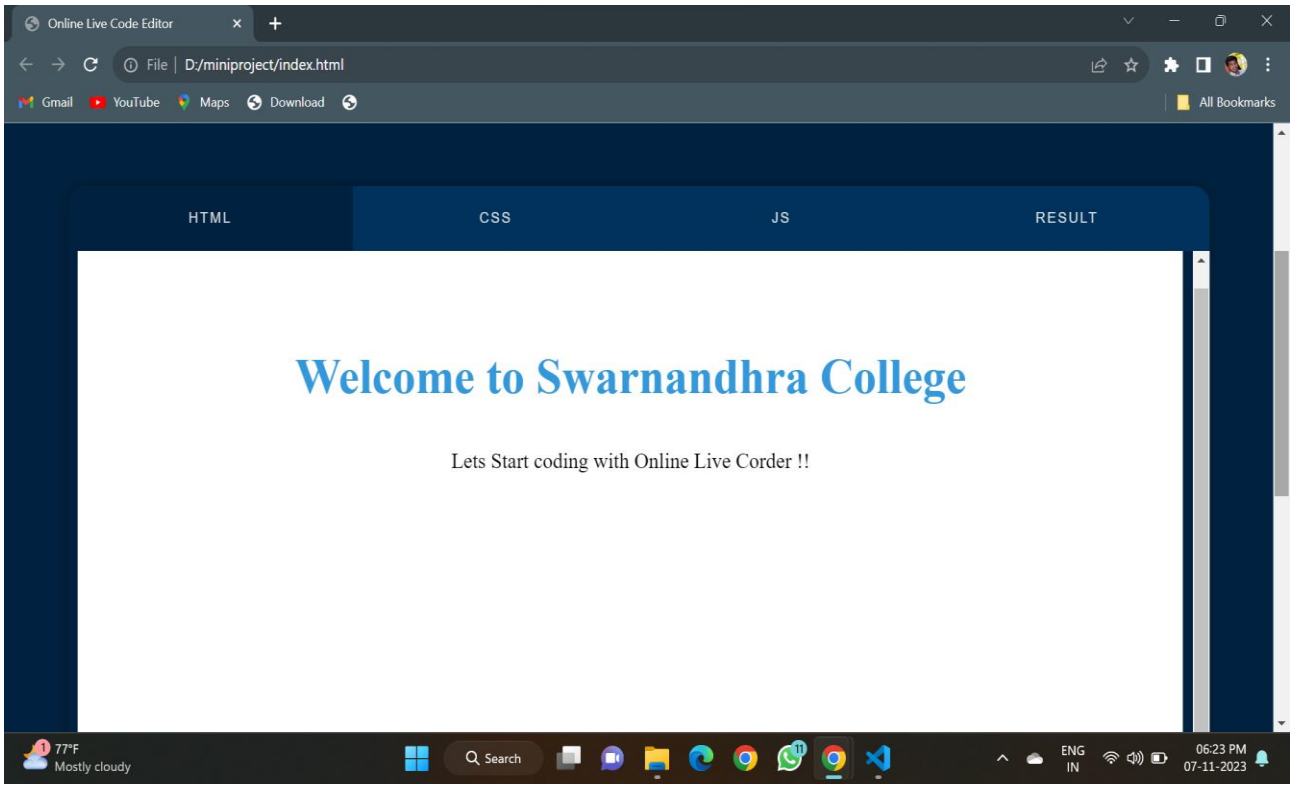


Output screen with code panels

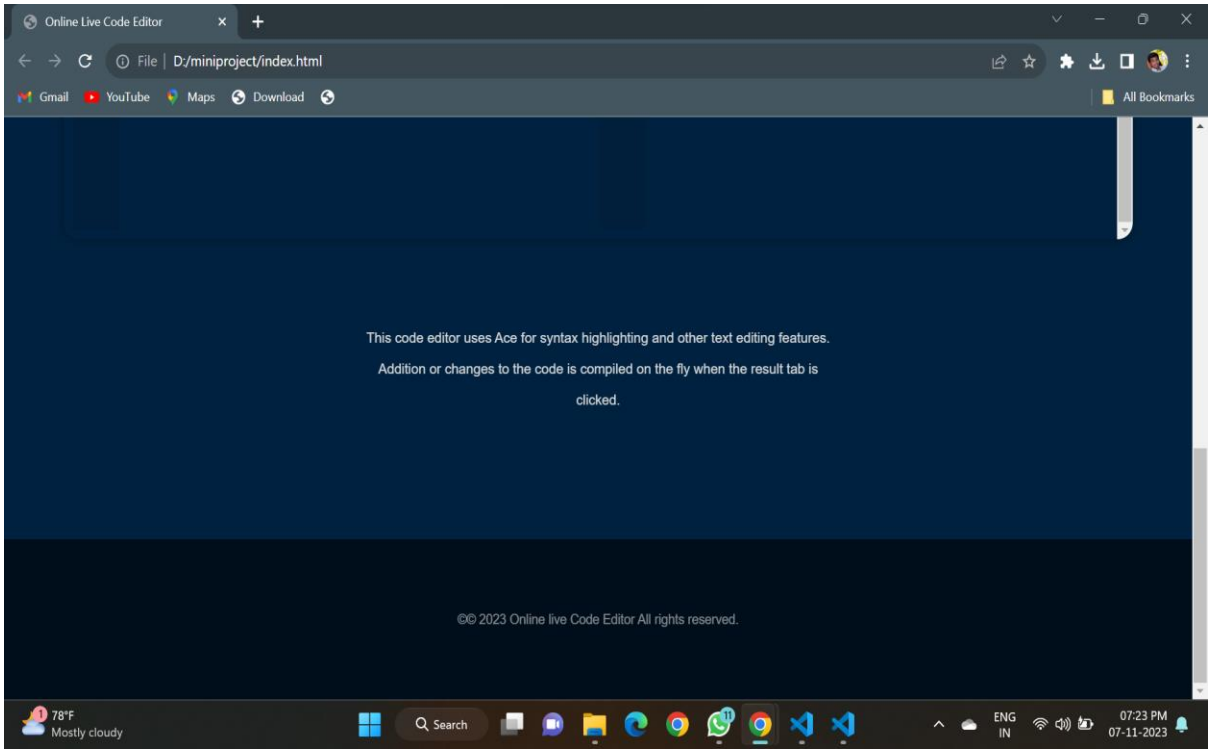


After writing the html css javascript code we can see the output by clicking the result panel the output will be displayed instantly

Total view of output screen



The footer note with a some keywords about online live code Editor Project



8.Conclusion

- After considering all the features of the website, “online live code Editor” is a much needed website for the MCA department students and all other users of various departments
- It will help other users to gain knowledge about web development from our website.
- we can practice web development pages and various style sheets like CSS and backend support java script thoroughly with this online live code editor