SOFTWARE ENGINEERING PROJECT

UNDER THE GUIDANCE OF:

DR. DINESH KUMAR VERMA

Jan, 2024 - May, 2024

Submitted in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY: TEAM N0.-1

KARTIK SHARMA - 211B160

MANAN JAIN – 211B173

PANSHUL KHURCHWAL – 211B202

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY

A-B ROAD, RAGHOGARH, DT. GUNA – 473226, M.P., INDIA

SYNOPSIS OF THE B.Tech. (CSE) III Year PROJECT for SE and APL-III

Tentative Title of Project: Online Code Editor with Live Preview

Team Details:

Team No.: 1

Name: Manan Jain **Er. No.:** 211B173
Email: 211b173@juetguna.in
Ph. Number: 8871113041 **Signatures**

Name: kartik sharma **Er. No.: 211B160**
Email:211b160@juetguna.in
Ph. Number: 6261677800 **Signatures**

Name:Panshul Khurchwal **Er. No.:211B202**
Email: 211b202@juetguna.in
Ph. Number: 8813982207 **Signatures**

Name: **Er. No.:**
Email:
Ph. Number: **Signatures**

Problem Formulation:
Key words- Code Editor, Text Editor, Live Preview.

Objectives/Aim- The online Code Editor will be used to provide code editing with live preview on a web page aiding the users to handle and edit code.

Description (not more than 100 words)-
This Online Code Editor project aims to deliver a comprehensive and intuitive solution for effective task management, catering to the needs of both individual users and collaborative teams. By providing a feature-rich and user-friendly environment, the application seeks to streamline the process of creating, organizing your code, ultimately enhancing users' ability to code seamlessly with direct error solutions upfront.

Technical Details:
Details of Methodology/ Approach of Development-

Implement the user interface using HTML, CSS, and JavaScript.
Utilize a frontend framework or library for enhanced functionality and responsiveness. Ensure a user-friendly design with intuitive navigation and interactive elements.
Integrate the frontend and backend components to ensure seamless communication. Test the integration to identify and address any compatibility or functionality issues.

Tools/ Languages to be used –

Frontend: HTML, CSS, JavaScript
Backend: [To be determined based on project requirements]

Synopsis Status

Faculty Remark-

Approved / Approved with Changes / Not Approved

Signature - **Date:-**

Lab 2: Writing a Project Narrative

Objective

- To obtain a preliminary user-view and a process view of the system.

Project Title: Online Code Editor with Live Preview

Project Summary and Overall Approach:

The Online Code Editor with Live Preview is a web-based platform aimed at providing users with a collaborative and real-time coding environment. This system allows users to write, edit, and execute code instantly, with a live preview of the output. The main goal is to create a user-friendly and efficient tool for coding enthusiasts, developers, and students, promoting experimentation and learning without the need for a local development setup.

User-View:

1. **Homepage:**
    - Users are greeted with an intuitive homepage that offers options to log in, sign up, and access a new code editor.
    - A search bar facilitates the discovery of existing code snippets.
2. **Code Editor Interface:**
    - The core interface allows users to write, edit, and preview code in real-time.
    - Features include syntax highlighting, autocompletion, and live preview panels for immediate visualization of code output.
3. **User Authentication:**
    - Registered users have a dedicated area for managing their accounts.
    - Options include login, sign-up, access to saved code snippets, and personalized profile settings.
4. **Collaboration Features:**
    - Users can initiate collaborative coding sessions with real-time indicators for multiple participants.
    - Shareable links facilitate collaboration, supported by a chat system for communication.
    - Version history and code comparison tools enhance collaborative project management.

5. **Dashboard:**
    - A personalized dashboard provides an overview of recent projects, code snippets, and collaboration notifications.
6. **Documentation and Support:**
    - Resources such as documentation, support forums, and FAQs are available for users to learn about and get assistance with the application.

Process View:

1. **User Registration:**
    - Users register by providing necessary details and activating their accounts through a verification email.
2. **Creating a New Code Snippet:**
    - Users navigate to the code editor, choose a language, and write or paste code.
    - Real-time syntax highlighting, autocompletion, and live preview aid code creation.
3. **Collaborative Coding Session:**
    - Users create collaborative projects, share links, and collaborate in real-time with communication features.
4. **Saving and Accessing Code Snippets:**
    - Users save code snippets, access them from the dashboard, and edit or preview as needed.
5. **Version Control:**
    - The system automatically tracks code changes in collaborative projects.
    - Users can revert to previous versions and utilize a visual diff tool for code comparison.
6. **User Preferences and Settings:**
    - Users manage their profile settings, customize editor preferences, and update account information.

Responsibilities of Team Members:

1. **Member 1:**
    - Gathers information from stakeholders (developers, students, coding enthusiasts).
    - Analyzes requirements and prepares a well-defined requirement set.

2. **Member 2:**
    - Prepares modeling diagrams based on information provided by Member 1.
3. **Member 3:**
    - Performs the construction activity, including coding and implementation.
4. **Member 4:**
    - Performs verification, validation, and deployment activities to ensure the functionality and stability of the system.

Common Project Narrative:

The Online Code Editor with Live Preview project aims to provide a collaborative coding platform with a user-friendly interface and real-time features. The team, comprising members responsible for gathering requirements, modeling, construction, and validation, collaborates to deliver a robust and user-centric system. The narrative highlights the seamless user experience, emphasizing collaboration, version control, and customization features.

LAB 3

Objective : To Identify a software process model for the project.

Project Title : Online Code Editor with Live Preview .

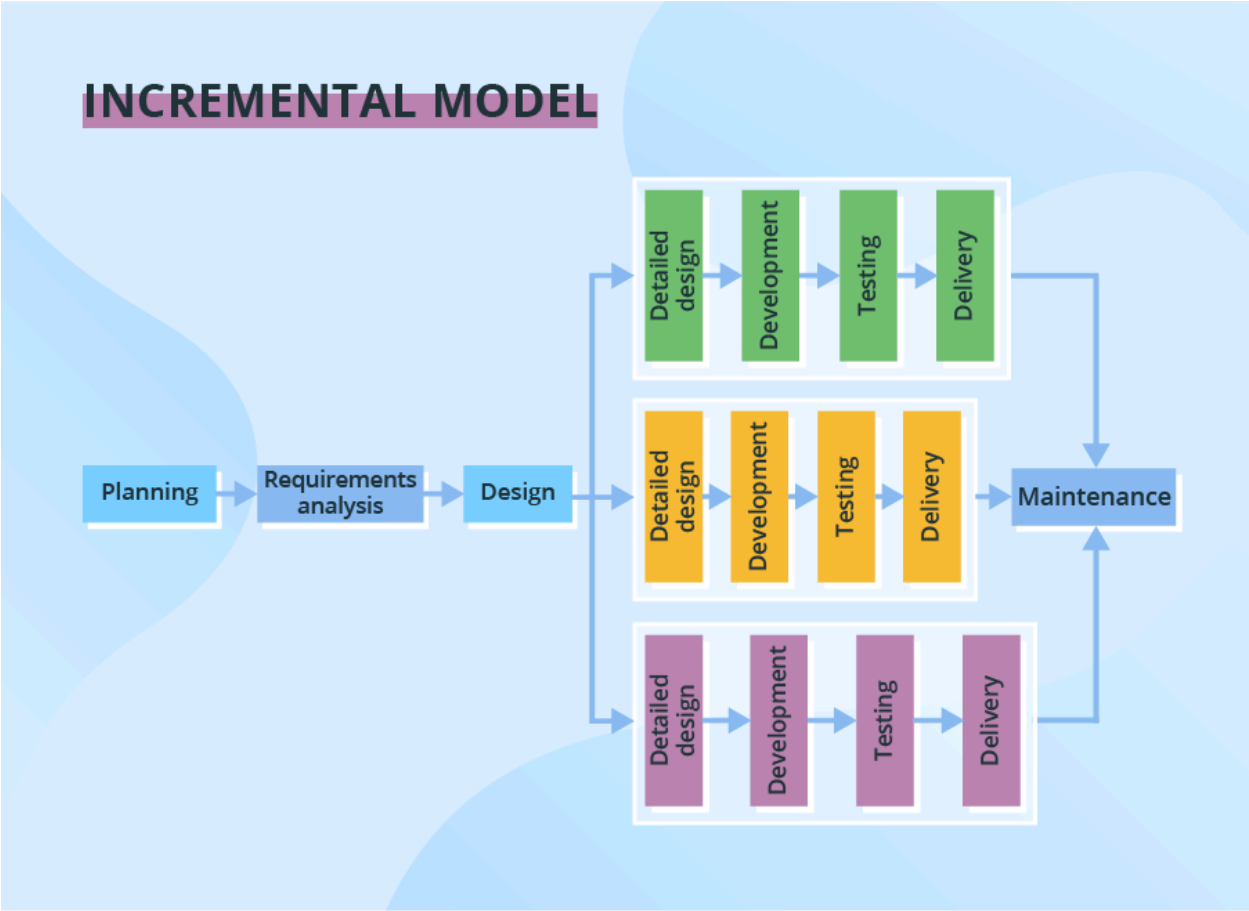
Proposed Development Model : Incremental Model

Overview :

The Incremental Model is a software development model that emphasizes building and delivering the system in small, manageable parts called **increments**. It is an iterative approach to development, where each increment represents a portion of the complete system's functionality. The development process involves progressively adding new features or enhancing existing ones in each iteration.

Features :

- Early Delivery of Core Features
- Adaptability to Changing Requirements
- Continuous Improvement
- Phased Deployment



Possible Modules :

- User Authentication and Profile Management Module
- Code Editor Interface Module
- Project Management and Version Control Module
- Collaboration and Sharing Module
- Dashboard and Overview Module
- User Feedback and Reporting Module
- Notification and Communication Module

Online Code Editor with Live Preview

1. Introduction

1.1 Purpose

The purpose of the project is to provide a collaborative and real-time coding environment for users. The project aims to offer a platform where developers, coding enthusiasts, and students can write, edit, and execute code seamlessly.

1.2 Scope

The Code Editor will include features such as code editing, user registration, live preview, theme Customization, project Management and version Control, dashboard Overview.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification
UI: User Interface
API: Application Programming Interface

2. Overall Description

2.1 Product Perspective

The Online Code Editor with Live Preview project positions itself as a collaborative coding platform, seamlessly integrating with the web development ecosystem. Leveraging the CodeMirror library, it offers an intuitive and responsive user experience, fostering learning, experimentation, and real-time collaboration. The project aims to contribute to the web developer community by providing a dynamic and supportive coding environment.

2.2 Product Features

- Code Editing
- Live Preview
- Collaborative Coding
- Theme Customization
- Project Management
- Dashboard Overview

2.3 User Classes and Characteristics

Developers : Actively engage in coding, collaborative projects, and use advanced code editing features

Students : Utilize the platform for educational purposes, experimenting with code, and collaborating on assignments.

Code Reviewers : Utilize collaboration and version control features for code review sessions.

Educators and Instructors : Leverage the platform for teaching coding concepts, conducting collaborative coding sessions, and assessing student work .

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Intuitive and Responsive: Ensuring a user-friendly experience.
- Accessible Across Devices: Fully responsive across various devices and screen sizes.
- Menu-Based Navigation: Users can navigate through the website using an intuitive menu-based navigation system.

3.1.2 Hardware Interfaces

The system shall be hosted on dedicated servers with the following minimum specifications:

Processor: Intel i5 or Amd Ryzen 5 Equivalent
RAM: 16 GB DDR4
Storage: 512 GB SSD
Network Interface: Gigabit Ethernet

The servers must be capable of supporting the required software stack, including the web server, database server, and application runtime environment.

The system shall be accessible from client devices, including desktop computers, laptops, tablets, and smart phones.

3.1.3 Software Interfaces

The system utilizes JavaScript CodeMirror library . The application shall be accessible via standard web browsers such as Google Chrome, Mozilla Firefox, and Safari.

3.2 Functional Requirements

3.2.1 Code Editor Management:

Creating New Code Snippets:

Users shall create new code snippets, specifying language, title, and collaborators .

Editing Code Snippets:

Users shall be able to edit existing code snippets, modifying code content, title, and collaborators.

Deleting Code Snippets:

Users shall have the ability to delete their code snippets.

3.2.2 Collaboration Management :

Collaborative Coding:

Users can collaborate on the same code snippet.

Version Control:

The system shall automatically track code changes in collaborative projects.

3.3 Non-Functional Requirements

3.3.1 Performance:

Responsiveness:

The code editor shall respond to user interactions (e.g., typing, editing) in real-time.

Concurrent User Load:

The system shall handle a concurrent user load of up to 500 users without significant performance degradation.

3.3.2 Security:

Code Security:

Implement secure code storage and access controls to protect the integrity of users' code snippets.

HTTPS Encryption:

Utilize HTTPS encryption to ensure secure communication between the client and the server.

LAB 5

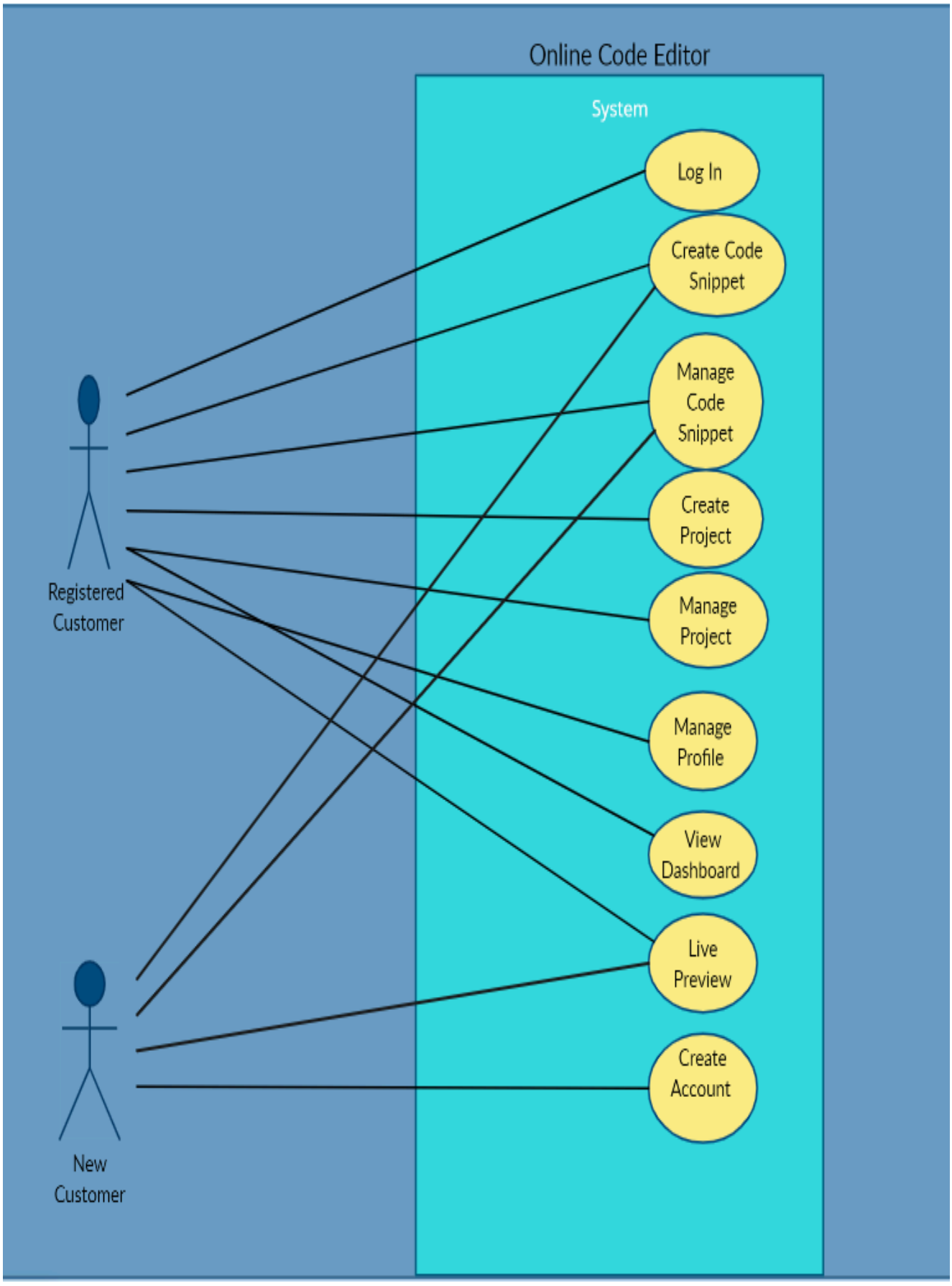
Objective : To Make Use Case Diagram .

Project Title : Online Code Editor with Live Preview .

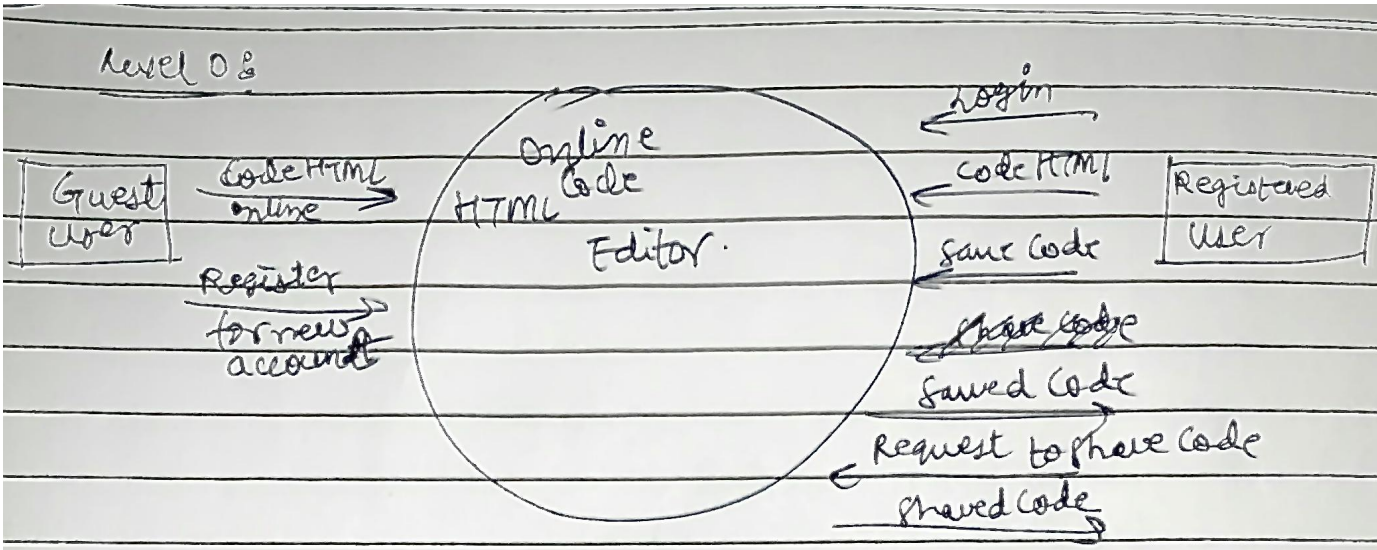
Actors : Registered User , Unregistered User (New User)

Use Case :

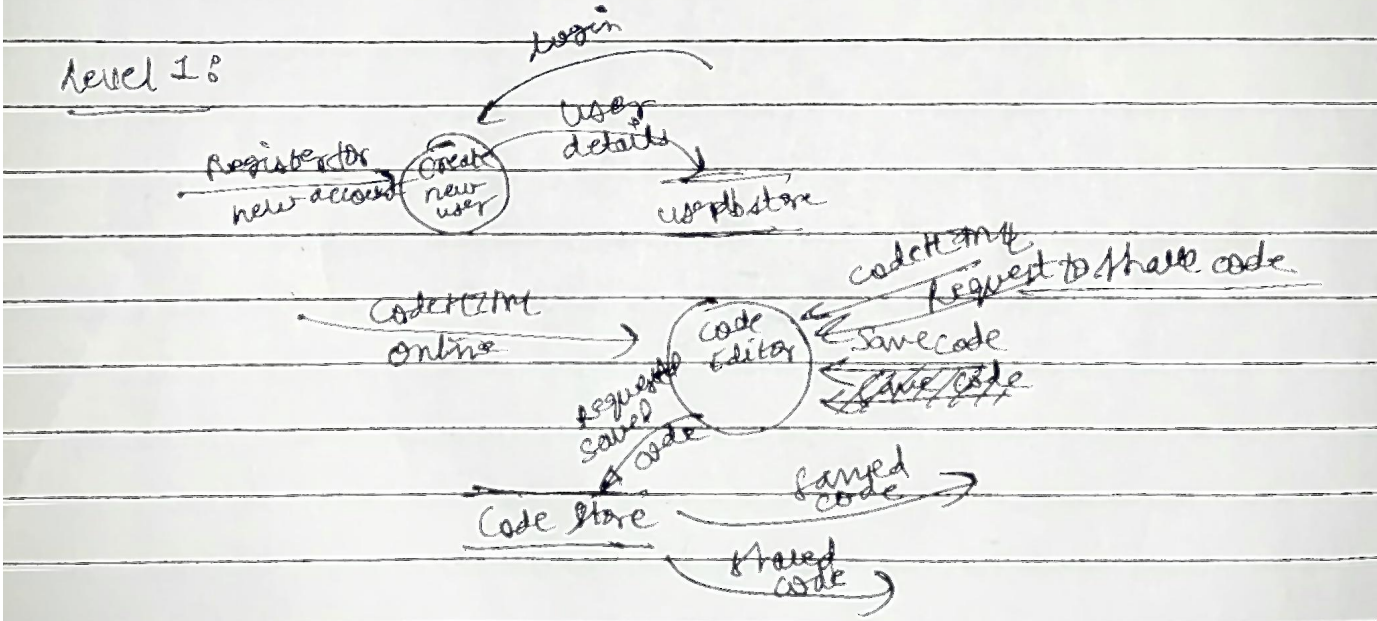
- Registered User :
    - Log In
    - Create New Code Snippet
    - Edit Code Snippet
    - Delete Code Snippet
    - View Project Dashboard
    - Add New Projects
    - Edit Project Details
    - Delete Projects
    - Manage Profile
    - Live Preview
- New User
    - Create Account
    - Live Preview
    - Create Code Snippet
    - Manage Code Snippet


Online Code Editor
System
Log In
Create Code Snippet
Manage Code Snippet
Create Project
Manage Project
Manage Profile
View Dashboard
Live Preview
Create Account
Registered Customer
New Customer

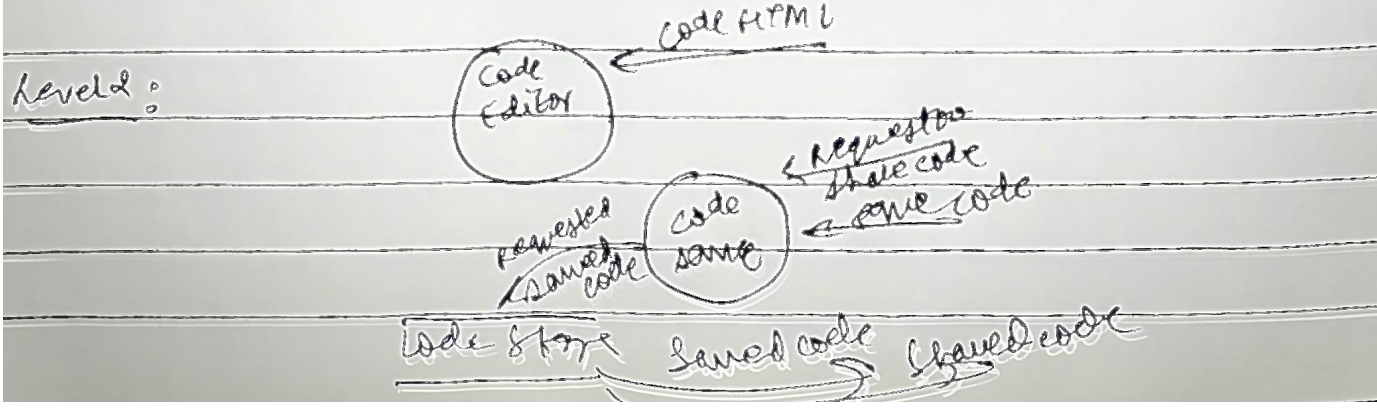
Level 0 :



Level 1 :

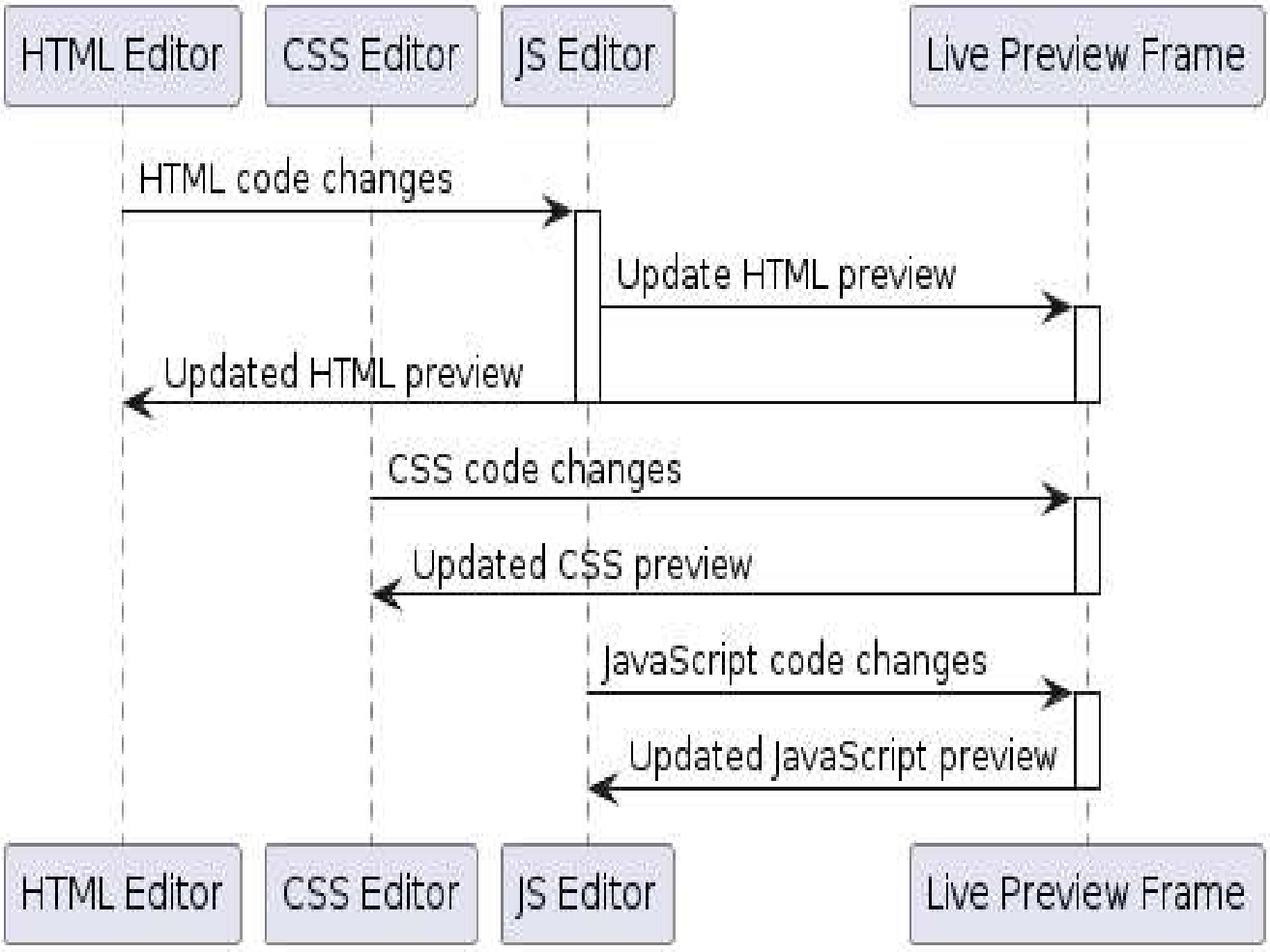


Level 2 :



LAB 7

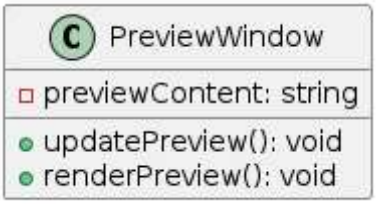
Objective : To Make a Sequence Diagram.

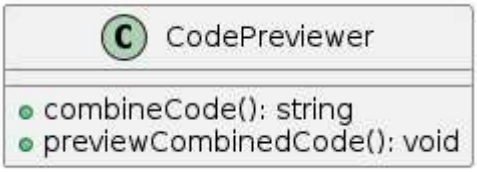


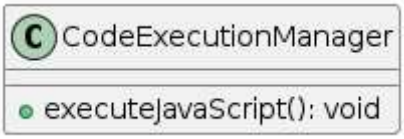
LAB 8

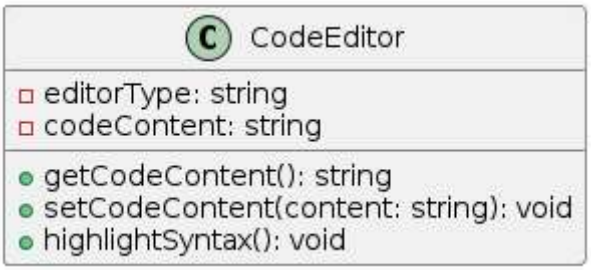
Objective :

To identify the classes, their attributes and methods.





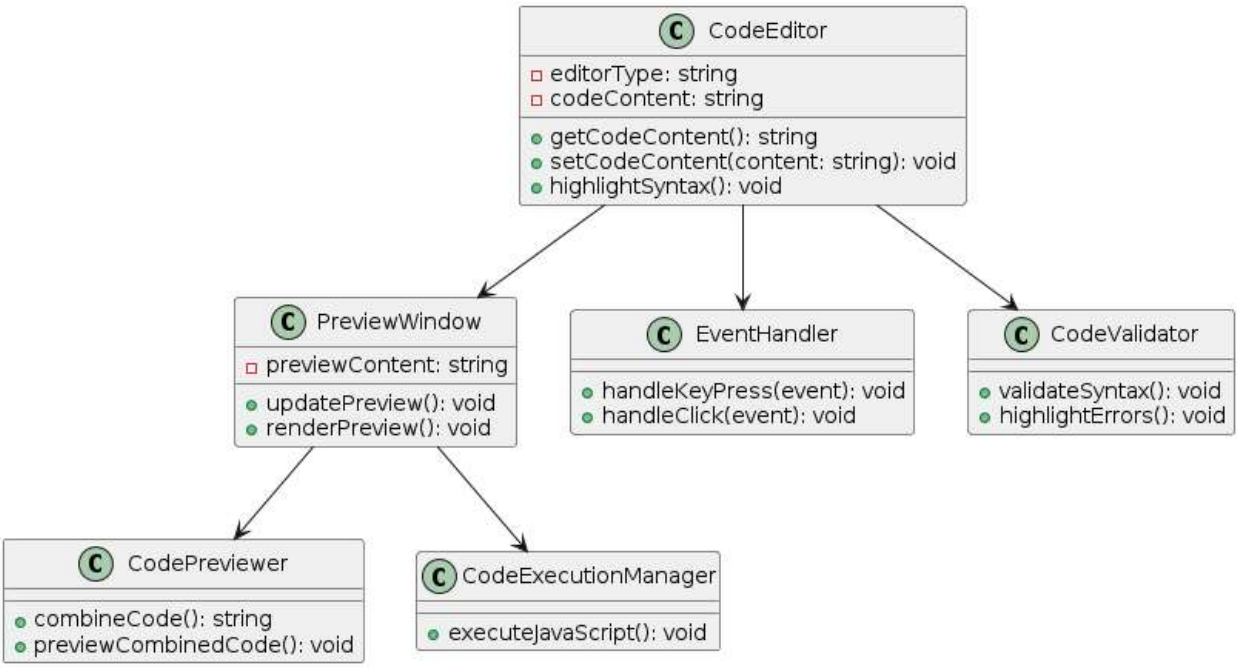




LAB 9

Objective :

To Make the Class Diagram .



```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/codemirror.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/theme/panda-syntax.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/addon/hint/show-hint.min.css">
    <link rel="stylesheet" href="style.css">
    <title>Live Code Editor </title>
</head>

<body>

    <!-- Login Form -->
    <div id="loginForm">
        <input type="email" id="email" placeholder="Email">
        <input type="password" id="password" placeholder="Password">
        <button id="loginButton">Login</button>
    </div>

    <!-- Code Editors (Initially hidden) -->
    <div id="codeEditors" style="display: none;">
        <div class="code-box">
            <div class="editor" id="html"></div>
            <div class="editor" id="css"></div>
            <div class="editor" id="js"></div>
        </div>
        <div class="preview">
            <iframe id="live-preview"></iframe>
        </div>
    </div>

    <script src="https://www.gstatic.com/firebasejs/9.0.2/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.0.2/firebase-auth.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.0.2/firebase-firestore.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/codemirror.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/mode/htmlmixed/htmlmixed.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/mode/css/css.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/mode/javascript/javascript.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/addon/edit/closetag.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/addon/edit/closebrackets.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/addon/hint/show-hint.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/6.65.7/mode/xml/xml.min.js"></script>
    <script type="module">
        // Import the functions you need from the SDKs you need
        import { initializeApp } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-app.js";
        import { getAnalytics } from "https://www.gstatic.com/firebasejs/10.11.1/firebase-analytics.js";

        import { getAuth, createUserWithEmailAndPassword } from "firebase/auth";
        // TODO: Add SDKs for Firebase products that you want to use
        // https://firebase.google.com/docs/web/setup#available-libraries

        // Your web app's Firebase configuration
        // For Firebase JS SDK v7.20.0 and later, measurementId is optional
        const firebaseConfig = {
          apiKey: "AIzaSyBvOoI1f3kg1YyR35UQMEGrTElJt4ZS9Mo",
          authDomain: "livecodeeditor-808ed.firebaseapp.com",
          projectId: "livecodeeditor-808ed",
          storageBucket: "livecodeeditor-808ed.appspot.com",
          messagingSenderId: "818373088803",
          appId: "1:818373088803:web:d477fabf98bd9924834396",
          measurementId: "G-1NG57GRQ2M"
        };

        // Initialize Firebase
        const app = initializeApp(firebaseConfig);
        const analytics = getAnalytics(app);
      </script>
    <script>



        // Function to handle login
        function login() {
            const email = document.getElementById("email").value;
            const password = document.getElementById("password").value;

            firebase.auth().signInWithEmailAndPassword(email, password)
                .then((userCredential) => {
                    // Signed in
                    const user = userCredential.user;
                    console.log(user);
                    // Hide login form and show code editors
                    document.getElementById("loginForm").style.display = "none";
                    document.getElementById("codeEditors").style.display = "block";
                    // Initialize code editors and set up live preview studio
                    setupLivePreviewStudio();
                })
                .catch((error) => {
                    const errorCode = error.code;
                    const errorMessage = error.message;
                    alert(errorMessage);
                });
        }

```

```
        // Event listener for login button
        document.getElementById("loginButton").addEventListener("click", login);

        // Catching commonly used elements to minimize dom queries
        const livePreviewFrame = document.getElementById('live-preview');
        const htmlEditor = document.getElementById('html');
        const cssEditor = document.getElementById('css');
        const jsEditor = document.getElementById('js');

        // Function to set up the live preview iframe and include necessary scripts
        function initializeLivePreview() {
            livePreviewFrame.contentWindow.document.body.innerHTML = '';
            const styleElement = document.createElement('style');
            styleElement.setAttribute('id', 'live-preview-style');
            livePreviewFrame.contentWindow.document.head.appendChild(styleElement);

            const pagedJsScript = document.createElement('script');
            pagedJsScript.src = 'https://unpkg.com/pagedjs/dist/paged.legacy.polyfill.js';
            livePreviewFrame.contentWindow.document.head.appendChild(pagedJsScript);
        }

        // Function to update the live preview iframe with the html code from editor
        function updateLiveHTMLPreview(codeEditors) {
            livePreviewFrame.contentWindow.document.body.innerHTML = codeEditors.html.getValue();
        }

        // Function to update the live preview iframe with the css code from editor
        function updateLiveCSSPreview(codeEditors) {
            const styleElement = livePreviewFrame.contentWindow.document.getElementById('live-preview-style');
            styleElement.innerHTML = codeEditors.css.getValue();
        }

        // Function to update the live preview iframe with the js code from editor
        function updateLiveJSPreview(codeEditors) {
            const scriptElement = document.createElement('script');
            scriptElement.innerHTML = codeEditors.js.getValue();
            livePreviewFrame.contentWindow.document.body.appendChild(scriptElement);
        }

        // Function to initialize CodeMirror editors for html, css and javascript
        function initializeCodeEditors() {
            function getDefaultOptions(object) {
                const defaultOptions = {
                    lineNumbers: true,
                    autoCloseTags: true,
                    autoCloseBrackets: true,
                    theme: 'panda-syntax'
                };
                if (object) {
                    const keys = Object.keys(object);
                    for (const key of keys) {
                        defaultOptions[key] = object[key];
                    }
                }
                return defaultOptions;
            }

            const codeEditors = {
                html: CodeMirror(htmlEditor, getDefaultOptions({
                    mode: 'text/html',
                    value: '',
                })),
                css: CodeMirror(cssEditor, getDefaultOptions({
                    mode: 'css',
                    value: '',
                    extraKeys: { 'Ctrl-Space': 'autocomplete' },
                    hintOptions: {
                        completeSingle: false,
                        closeOnUnfocus: false
                    }
                })),
                js: CodeMirror(jsEditor, getDefaultOptions({
                    mode: 'javascript',
                    value: ''
                })),
            };
            return codeEditors;
        }

        // Function to set up the live preview studio with CodeMirror editors and event listeners
        function setupLivePreviewStudio() {
            const codeEditors = initializeCodeEditors();

            // Event listener for changes in HTML editor
            CodeMirror.on(codeEditors.html, 'change', () => {
                updateLiveHTMLPreview(codeEditors);
            });

            // Event listener for changes in CSS editor
            CodeMirror.on(codeEditors.css, 'change', () => {
                updateLiveCSSPreview(codeEditors);
            });

            // Event listener for changes in HTML editor
            CodeMirror.on(codeEditors.js, 'change', () => {
                updateLiveJSPreview(codeEditors);
            });
        }

        // Event listener to set up the live preview studio after the dom is fully loaded
        document.addEventListener('DOMContentLoaded', () => {
            initializeLivePreview();
        });
    </script>
</body>

</html>
```

```css
@import url('https://fonts.googleapis.com/css2?family=Source+Sans+3:wght@300;400;500;600;700&display=swap');

body{
    margin: 0;
}

.container{
    font-family: "Source Sans 3", sans-serif;
}

.header{
    background: #000;
    color: #fff;
    height: 60px;
    border-bottom: 5px solid #333;
}

.header .title{
    padding: 0.5rem;
}

.header .title .main-title{
    font-size: 18px;
    font-weight: 600;
}

.header .title .sub-title{
    font-size: 14px;
}

.header .title .sub-title span{
    color: #999;
}

.editor{
    width: 33.33%;
    height: 100%;
    float: left;
    box-sizing: border-box;
    position: relative;
}

.editor #html{
    border-right: 1px solid #777;
}

.editor #css{
    border-left: 20px solid #333;
    border-right: 1px solid #777;
}

.editor #js{
    border-left: 20px solid #333;
}

.code-box{
    position: relative;
    height: calc(50vh - 4.6rem);
    border-top: 1px solid #000;
    border-bottom: 10px solid #333;
    overflow: hidden;
}

.preview{
    position: relative;
    height: calc(90vh - 30px);
    background: #fff;
    border-top: 10px ridge #333;
}

.preview iframe{
    width: 100%;
    height: 100%;
}

::-webkit-scrollbar{
    width: 5px;
    height: 5px;
}

::-webkit-scrollbar-track{
    background: transparent;
}

::-webkit-scrollbar-thumb{
    background: #555;
    border-radius: 20px;
}
```

```js
// Catching commonly used elements to minimize dom queries
const livePreviewFrame = document.getElementById('live-preview');
const htmlEditor = document.getElementById('html');
const cssEditor = document.getElementById('css');
const jsEditor = document.getElementById('js');

// Function to set up the live preview iframe and include necessary scripts
function initializeLivePreview() {
    livePreviewFrame.contentWindow.document.body.innerHTML = '';
    const styleElement = document.createElement('style');
    styleElement.setAttribute('id', 'live-preview-style');
    livePreviewFrame.contentWindow.document.head.appendChild(styleElement);

    const pagedJsScript = document.createElement('script');
    pagedJsScript.src = 'https://unpkg.com/pagedjs/dist/paged.legacy.polyfill.js';
    livePreviewFrame.contentWindow.document.head.appendChild(pagedJsScript);
}

// Function to update the live preview iframe with the html code from editor
function updateLiveHTMLPreview(codeEditors) {
    livePreviewFrame.contentWindow.document.body.innerHTML = codeEditors.html.getValue();
}

// Function to update the live preview iframe with the css code from editor
function updateLiveCSSPreview(codeEditors) {
    const styleElement = livePreviewFrame.contentWindow.document.getElementById('live-preview-style');
    styleElement.innerHTML = codeEditors.css.getValue();
}

// Function to update the live preview iframe with the js code from editor
function updateLiveJSPreview(codeEditors) {
    const scriptElement = document.createElement('script');
    scriptElement.innerHTML = codeEditors.js.getValue();
    livePreviewFrame.contentWindow.document.body.appendChild(scriptElement);
}

// Function to initialize CodeMirror editors for html, css and javascript
function initializeCodeEditors() {
    function getDefaultOptions(object) {
        const defaultOptions = {
            lineNumbers: true,
            autoCloseTags: true,
            autoCloseBrackets: true,
            theme: 'panda-syntax'
        };
        if (object) {
            const keys = Object.keys(object);
            for (const key of keys) {
                defaultOptions[key] = object[key];
            }
        }
        return defaultOptions;
    }

    const codeEditors = {
        html: CodeMirror(htmlEditor, getDefaultOptions({
            mode: 'text/html',
            value: '',
        })),
        css: CodeMirror(cssEditor, getDefaultOptions({
            mode: 'css',
            value: '',
            extraKeys: { 'Ctrl-Space': 'autocomplete' },
            hintOptions: {
                completeSingle: false,
                closeOnUnfocus: false
            }
        })),
        js: CodeMirror(jsEditor, getDefaultOptions({
            mode: 'javascript',
            value: ''
        })),
    };
    return codeEditors;
}

// Function to set up the live preview studio with CodeMirror editors and event listeners
function setupLivePreviewStudio() {
    const codeEditors = initializeCodeEditors();

    // Event listener for changes in HTML editor
    CodeMirror.on(codeEditors.html, 'change', () => {
        updateLiveHTMLPreview(codeEditors);
    });

    // Event listener for changes in CSS editor
    CodeMirror.on(codeEditors.css, 'change', () => {
        updateLiveCSSPreview(codeEditors);
    });

    // Event listener for changes in HTML editor
    CodeMirror.on(codeEditors.js, 'change', () => {
        updateLiveJSPreview(codeEditors);
    });
}

// Event listener to set up the live preview studio after the dom is fully loaded
document.addEventListener('DOMContentLoaded', () => {
    initializeLivePreview();
    setupLivePreviewStudio();
});
```