# Online Code Editor with Live Preview — Project Report

## Objective

The objective of this project was to create a browser-based Integrated Development Environment (IDE) that allows users to write and preview HTML, CSS, and JavaScript code in real time. The solution should be simple, accessible, and effective for both beginners and experienced developers who want to quickly test code snippets.

## Tools and Technologies

- **taiwind.js**: For building the front-end application and handling component-based UI.

- **Monaco Editor**: Provides the powerful code editing experience (used in VS Code) inside the browser.

- **iframe**: Used to render and sandbox the live preview of the written code.

- **Tailwind CSS (optional)**: For quick styling and layout.

## Features Implemented

1. **Code Editing**:

   - Three separate editors for HTML, CSS, and JavaScript.

   - Powered by Monaco Editor for syntax highlighting, indentation, and a VS Code-like experience.

2. **Live Preview**:

   - Code is compiled and rendered into an iframe in real-time with a slight debounce for performance.

   - Sandboxed iframe ensures the user's JavaScript runs safely without accessing the parent page.

3. **Templates**:

   - Built-in templates such as a vanilla HTML/JS starter and a React (CDN) starter.

   - Quick way for users to jump into coding without starting from scratch.

4. **Sharing**:

   - Code can be encoded and shared via URL query parameters.

   - When a shared link is opened, the editor automatically loads the code from the URL.

5. **Layout Views**:

    ○ **Side-by-side**: Editor and preview displayed next to each other.

    ○ **Preview only**: Hides editors and shows only the live output.

    ○ **Editor only**: Hides preview to focus purely on coding.

**Optional / Future Enhancements**

- **Snippet Storage**: Save code snippets to cloud storage (e.g., Firebase, AWS, or database) for retrieval across devices.

- **Collaboration**: Enable multiple users to code together in real-time (like Google Docs or CodeSandbox).

- **Multi-language Support**: Add support for additional languages such as TypeScript, Markdown, or frameworks.

- **Mobile Responsiveness**: Improve the editor experience for tablets and mobile devices.

## Implementation Summary

The application is structured as a React component. Monaco Editor instances manage the code for HTML, CSS, and JS. Whenever the code changes, it is injected into an iframe for preview. For sharing, code is serialized, base64 encoded, and appended to the URL. On load, the application parses the URL to restore the code session.

## Conclusion

The online code editor provides an effective, browser-based environment for testing HTML, CSS, and JavaScript snippets. It combines the familiar Monaco Editor with live iframe previews, making it a lightweight yet powerful tool. The ability to share code via URLs and apply quick templates enhances usability, while future improvements like cloud storage and collaboration could make it even more feature-rich.

*End of Report*