CPRO 1102

# Front End Web Development II Project Proposal

-By Avi Pancholi

& Pawel Leszczynski

& Walter Serrano

## 1. PROJECT TITLE

**Code Viewer** – An Interactive Platform for Viewing and Testing Code

---

## 2. PROJECT INTRODUCTION AND DESCRIPTION

### a) Project Overview

The **Code Viewer** project provides an interactive platform for users to view, edit, and test code snippets across multiple programming languages. The platform features essential tools such as syntax highlighting, a light-dark theme switcher, and real-time preview mode, creating an optimal environment for beginners to experiment and learn code.

### b) Problem or Need

Many beginners struggle to find a simple yet effective platform to view and test code. Existing tools can be too complex or require additional setup, which can overwhelm new programmers. The **Code Viewer** addresses this need by offering an easy-to-use platform with minimal distractions, allowing users to focus on learning and practicing code.

### c) Functionality and Features

The **Code Viewer** will include the following core features:

1. **Light-Dark Mode Theme Switcher:** A toggle for switching between light and dark themes for optimal viewing.

2. **API Integration:** Connects to third-party APIs to fetch or store code snippets, using an iframe to process and view the code externally.

3. **Responsive Code Viewer:** Includes syntax highlighting, a display selector, a copy button for code, and a preview mode that adapts to screen sizes.

4. **Responsive Navigation:** A navigation bar that disappears or collapses when the screen width is small, ensuring a clean mobile experience.

5. **Express Library:** Uses the Express library to host the application locally for testing and development.

### d) Technologies and Tools

- **Programming Languages:** HTML, CSS, JavaScript

- **Libraries/Frameworks:** Express (for local hosting), CodeMirror (or Ace Editor) for syntax highlighting

- **Tools:** Visual Studio Code for development, GitHub for version control, and Figma for UI design.

---

## 3. FEATURES

For each feature, here's a breakdown of title, purpose, functionality, and integration:

1. **Feature Title:** Light-Dark Mode Theme Switcher
    - **Purpose:** Provides an accessible way for users to switch between light and dark modes for comfortable viewing.
    - **Functionality:** A toggle button in the UI allows users to change themes, updating both the editor and preview area.
    - **Integration:** Affects the overall appearance of the editor and preview mode, ensuring a consistent theme across the application.
2. **Feature Title:** API Integration (via Iframe)
    - **Purpose:** Allows the platform to fetch and process code snippets via third-party APIs, displayed within an iframe for viewing and testing.
    - **Functionality:** The iframe is used to load an external service that processes and displays the code. This could be platforms like JSFiddle, CodePen, or Repl.it, allowing users to see their code in action without the need for backend processing.

- **Integration:** The iframe dynamically loads and displays the external code preview, interacting with the front-end code to update the display based on user input.

3. **Feature Title:** Responsive Code Viewer

    - **Purpose:** Provides an adaptable code viewer that adjusts to various screen sizes, ensuring an optimal viewing experience on any device.

    - **Functionality:** The viewer includes syntax highlighting, a display selector, a copy button for users to quickly copy code, and a preview mode to see real-time results of the code.

    - **Integration:** Integrated into the main interface, automatically adjusting to screen size while keeping the user interface clean and usable.

4. **Feature Title:** Responsive Navigation

    - **Purpose:** Enhances user experience by making the navigation bar responsive, so it collapses or disappears on smaller screens.

    - **Functionality:** The navigation bar hides or condenses into a hamburger menu when the screen width is reduced.

    - **Integration:** Tied to the CSS and JavaScript that handles the layout, this ensures the navigation remains functional and user-friendly on mobile devices.

5. **Feature Title:** Express Library for Local Hosting

    - **Purpose:** Uses Express.js to set up a local server for hosting the application during development and testing.

    - **Functionality:** Runs a local server using Express to serve the project files, allowing for live previews of changes.

    - **Integration:** Integrates with the front-end and back-end of the application, making it easier to test functionality on a local machine before deployment.

---

## 4. CONTRIBUTION OF EACH MEMBER

**Member 1:** Avi

- **Role:** Ui/Ux designer

- **Responsibilities:** Set up Express to host the project locally, manage the API integrations via iframe, and oversee project structure.

**Member 2:** Pawel

- **Role:** Lead Tester, Documentation Specialist
- **Responsibilities:** Build the responsive navigation bar, implement the light-dark mode theme switcher, and design the user interface using HTML and CSS.

**Member 3:** Walter Serrano

- **Role:** JavaScript Developer
- **Responsibilities:** Implement the responsive code viewer features, including syntax highlighting and real-time preview, and assist with the iframe-based API integration.