**Key Features to Emphasize**

1. **Frontend (React)**
   * **Integrate a basic code editor.**
   * **Develop a user-friendly interface with essential features.**
   * **Support real-time code editing for individual sessions.**
2. **Backend (Node.js)**
   * **Create a basic API for code compilation and execution.**
   * **Implement user authentication and project management.**
3. **Database (MongoDB)**
   * **Store user profiles, code snippets, and project data.**
4. **Containerization (Docker)**
   * **Set up Docker for consistent development environments.**

**Revised Timeline and Milestones**

**Weeks 1-2: Initial Setup and Basic Functionality**

* **Project Setup**
  + **Establish repositories (GitHub/GitLab).**
  + **Create an initial Dockerfile and Docker Compose configuration.**
* **Frontend Development**
  + **Initialize the React project.**
  + **Integrate a code editor (e.g., Monaco Editor or CodeMirror).**
  + **Design a basic UI (home page and code editor page).**
* **Backend Development**
  + **Start the Node.js project with Express.**
  + **Develop a basic API for code submission and execution.**
  + **Set up MongoDB and design the basic schema for users and projects.**

**Weeks 3-4: Core Features and Integration**

* **Frontend Enhancement**
  + **Add user authentication features (sign up, login).**
  + **Implement code editor functionalities (syntax highlighting, basic real-time editing).**
* **Backend Enhancement**
  + **Develop APIs for user authentication and project management.**
  + **Integrate language compilers/interpreters for JavaScript, Python, Java, and C/C++.**
* **Database Integration**
  + **Complete CRUD operations for user data, code snippets, and projects.**

**Weeks 5-6: Real-Time Collaboration and Security**

* **Real-Time Features**
  + **Implement WebSocket or similar technology for real-time code updates.**
* **Security**
  + **Introduce basic security measures (input validation, secure storage).**

**Weeks 7-8: Testing, Optimization, and Deployment**

* **Testing**
  + **Write and run unit and integration tests for both frontend and backend.**
* **Optimization**
  + **Optimize performance and scalability.**
  + **Ensure seamless Docker container operation.**

**06/26/2024 (Additional Features to be Implemented)**

1. **User Profile**
   * **Replace the top two buttons (Sign In and Sign Up) with the user's first name and a profile icon. Clicking the icon should reveal a well-integrated log-out button. Logging out should redirect to the home page with the Sign In and Sign Up buttons. [COMPLETED]**
2. **Post-Sign-Up Redirect**
   * **Redirect to the sign-in page after a successful sign-up, instead of the home page. [COMPLETED]**
3. **Project Access**
   * **Restrict access to projects if the user is not signed in. [COMPLETED]**
4. **Project Saving Restriction**
   * **Prevent saving projects if not signed in and redirect to the Sign In page. Ensure that code editor content is preserved if signed in after redirection. [COMPLETED]**
5. **Project Display and Interaction**
   * **Display a list of projects associated with the current signed-in user. Projects should be clickable and open in the code editor when selected. [COMPLETED]**
6. **Duplicate Project Handling**
   * **Show a message "File already exists, do you want to replace?" when attempting to save a project with an existing name. Allow replacement if confirmed, otherwise, keep the original. [COMPLETED]**
7. **Save Project Button Design**
   * **Design the save project button and project name interface nicely. [COMPLETED]**
8. **My Projects Page Design**
   * **Finalize the design for the My Projects page and overall UI/UX. [COMPLETED]**
9. **Project Finalization**
   * **Complete the project. [COMPLETED]**