**CODECOLLAB**

**1. Introduction**

CodeCollab is a user-friendly web app designed for collaborative coding using modern technology. It lets developers securely log in, create virtual coding rooms, and work together in real-time. You can use a multi-language code editor to edit and save code simultaneously with others. The platform includes sections for coding tasks, where teams can solve problems together. A built-in chat box allows for instant communication, making it easy to discuss ideas and coordinate effectively. CodeCollab aims to enhance teamwork and productivity by providing a seamless environment for collaborative coding sessions.

### 2. Features

1. **User Registration and Room Creation**
   * Users must be able to sign up for an account to access CodeCollab.
   * Upon registration, users should have the ability to create virtual rooms for collaborative coding sessions.
2. **Joining Rooms via URL**
   * Participants should join a room by clicking on a shared URL provided by the room creator.
3. **Multi-language Code Editor**
   * CodeCollab must support a multi-language code editor that allows simultaneous editing and viewing of code in multiple programming languages within a virtual room.
4. **Problem Statements**
   * The platform needs to include a dedicated section where users can view and work on assigned coding tasks or problems.
5. **Real-time Chat Interface**
   * CodeCollab should feature a real-time chat box within each virtual room to facilitate instant communication among participants.
6. **Save Code Feature**
   * Provides an option for users to save the code.

### 3. User Roles

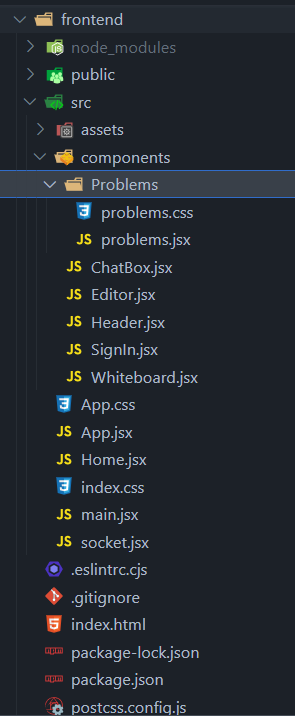
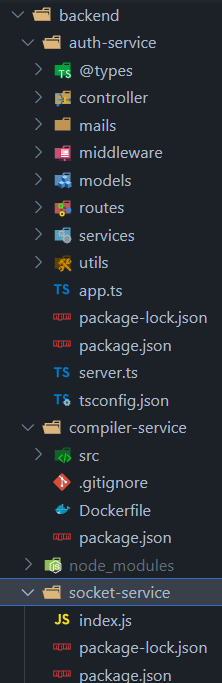
CodeCollab caters to two main types of users:

* **Developers**: Individuals actively coding, either solo or in teams.
* **Students**: Aspiring developers seeking practical coding experience and opportunities to enhance their skills.

**4. Architecture Overview**

CodeCollab is structured with a modular architecture leveraging the MERN (MongoDB, Express.js, React.js, Node.js) stack:

* **Frontend**:
  + Developed using React.js, the frontend interacts seamlessly with backend APIs to fetch data and render the user interface.
  + Enhanced with Tailwind CSS for efficient styling and responsive design, ensuring a modern and cohesive user experience.
* **Backend**:
  + Powered by Express.js and Node.js, the backend delivers RESTful APIs primarily for user authentication and collaborative coding functionalities.
* **Database**:
  + MongoDB serves as the persistent storage solution for user data and saved code snippets, ensuring robust data management and retrieval.

### 5. Frontend Design

The frontend of CodeCollab is built using React.js and styled with TailwindCSS, offering a modern and responsive user interface. It comprises the following key components:

1. **Header Section**: Provides intuitive navigation links for seamless exploration of different features and sections within CodeCollab.
2. **Editor Section**: Supports concurrent editing and viewing of code across various programming languages, enhancing collaborative coding sessions.
3. **SignIn Bar**:Allows users to securely sign in, create new accounts, and manage their authentication status to access CodeCollab's features.
4. **Whiteboard Section**: Enhances communication alongside coding activities by enabling users to brainstorm, sketch ideas, and visually discuss concepts.
5. **Chat Interface**: Includes a real-time chat box embedded within each coding room for instant communication among participants.
6. **Problem Statements Section**: Allows users to view problem statements and collaborate on solutions.

**6. Database Design**

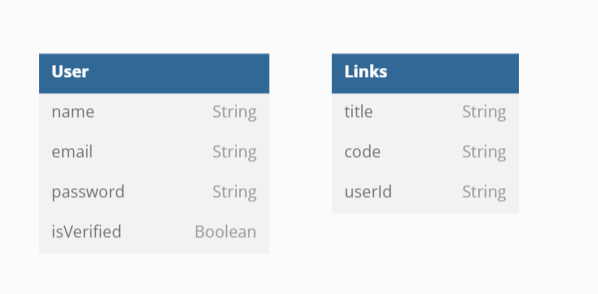
The MongoDB database consists of the following collections:

1. **Users**:

* **Purpose**: Stores user information and manages user authentication within CodeCollab.
* **Usage**: Supports user registration and authentication.

1. **Links**:

* **Purpose**: Used to save and manage code snippets.
* **Usage**: Enables users to save and retrieve code snippets or links, facilitating collaborative coding sessions within virtual rooms.

****

**7. API Design**

CodeCollab backend provides the following RESTful APIs:

**Socket Service:** Handles socket connections, disconnections, and real-time events between clients and the server.

* 1. **connection Event**: Triggered when a new client connects to the socket.io server.
  2. **disconnect Event**: Triggered when a client disconnects from the server.
  3. **join-room Event**: Allows clients to join a specific room identified by a room code.
  4. **code-update, input-update, run-code, output-update, theme-update, lang-update, whiteboard-update, toggled-whiteboard, color-update Events**: Emits updates to all clients within the same room, ensuring synchronized state across participants.
  5. **send-message Event**: Handles chat messages sent by clients within a room.
  6. **leave-room Event**: Allows clients to leave a specific room.

**Compiler Service**

* 1. **POST '/compile':** This route is used for executing or compiling code submitted by clients.
  2. **GET '/compile/list'**: This function would fetch and return a list of programming languages or compilers that the server supports for code execution or compilation.

**Authorisation Service**

**User Routes:**

1. **POST '/api/registration'**: Creating a new user record in the database, and sending a confirmation email.
2. **POST '/api/activate-user'**: Responsible for verifying and activating a user account based on the activation token provided.
3. **POST '/api/login'**: Upon successful authentication, it generates and returns JWT tokens that allow the user to access protected routes.
4. **GET '/api/logout'**: Upon verification, the logoutUser function invalidates the user's access token, effectively logging them out.
5. **GET '/api/me'**: If authenticated, the getUserInfo function retrieves and returns information about the user (e.g., name, email, profile details).
6. **PUT '/api/update-user-info'**: Upon authentication, the updateUserInfo function updates the user's profile information in the database.
7. **PUT '/api/ update-user-password'**: Used to update the user's password in the database.

**Link Routes:**

1. **POST '/api/create'**: Handles the creation of a new link, typically storing it in a database.
2. **GET '/api/all'**: Fetches all links from the database that belong to the authenticated user and returns them.
3. **POST '/api/save/:id'**: Upon successful authentication, updates the code associated with the specified link.
4. **DELETE '/api/delete/:id'**: Upon authentication, the deleteLink function is called to delete the specified link from the database.

**8. Technologies Used**

* **Front-end**: JavaScript, React.js, Tailwind CSS
* **Back-end**: Node.js, Express.js, Mongoose
* **Database**: MongoDB
* **Authentication**: JWT, Bcrypt.js, dotenv
* **Real-time Communication**: Socket.io

**9. Security**

* User data is securely stored and transmitted using encryption.
* Authentication tokens are used to ensure secure user access.

**10. Future Enhancements**

* **Add Video and Audio Chat**: Enhance collaboration by integrating real-time video and audio communication capabilities. This feature will allow users to communicate face-to-face during coding sessions, facilitating clearer communication and quicker issue resolution.
* **Record Sessions**: Implement session recording functionality to enable users to capture and save their collaborative coding sessions. Recorded sessions can be reviewed later for learning purposes or shared with team members who couldn't attend the live session. This feature enhances productivity by providing a valuable resource for retrospective analysis and knowledge sharing.

**11. Conclusion**

CodeCollab is a dedicated platform designed to enable real-time collaborative coding sessions. It offers essential features like a multi-language code editor, problem statements, and real-time chat, catering to developers and students alike. With CodeCollab, users can efficiently collaborate on coding tasks, enhancing productivity and fostering a collaborative coding culture.