# MVP -DevMountain

1. **Project Setup**:
   * **Version Control**: Initiate a Git repository for your project. This will make tracking changes and collaborating easier.
   * **Project Structure**: Set up a basic structure with folders for the frontend and backend.
2. **Backend Basics**:
   * **Server**: Set up a basic Node.js server using Express.js.
   * **Database**: Choose a database and integrate it. Initialize tables/collections needed for users and chat messages.
   * **WebSocket Setup**: Integrate **socket.io** or another WebSocket library to establish real-time communication capabilities.
3. **Frontend Skeleton**:
   * **React Setup**: Use Create React App or another boilerplate to initialize your React app.
   * **Chat Interface**: Build a basic chat interface component with a text input and message display area.
   * **Connect Frontend with WebSocket**: Implement basic message sending and receiving functionality.
4. **User Authentication**:
   * Implement user sign-up and login functionality on the backend.
   * Create frontend components for login and sign-up forms.
   * Integrate the frontend and backend authentication. Ensure that users are authenticated before they can send messages.
5. **ChatGPT Debate Topic Integration**:
   * Use the OpenAI API (or another provider's API) to generate debate topics.
   * Implement a button or an event trigger on the frontend to fetch a topic.
   * Display the topic prominently in the chat interface.
6. **Frontend Enhancements**:
   * **Styling**: Use CSS and potentially other libraries to style and polish your chat interface.
   * **User Feedback**: Add features like displaying "user is typing..." and read receipts.
7. **Backend Enhancements**:
   * **Message Storage**: Store chat messages in the database for history or potential future features.
   * **User Profiles**: Store and retrieve user profile data like avatars, usernames, etc.
8. **Testing**:
   * **Unit Tests**: Implement unit tests for both frontend and backend functionalities.
   * **End-to-End Tests**: Use tools like Cypress or Selenium to test the overall functionality and interactions of your website.
9. **Deployment**:
   * Set up a deployment environment (e.g., Heroku, Vercel, AWS).
   * Deploy both frontend and backend. Make sure to secure any sensitive information, like API keys.