

|    |  |
|----|--|
| 1. | <b>Project Setup</b> <ul style="list-style-type: none"><li>• Set up your project directory.</li><li>• Initialize a Git repository and connect it to GitHub.</li><li>• Set up Node.js environment and install necessary dependencies using npm or yarn.</li><li>• Set up MongoDB for the database.</li></ul>  |
| 2. | <b>Backend Development</b> <ul style="list-style-type: none"><li>• Implement server-side logic using Node.js.</li><li>• Set up WebSocket communication for real-time messaging using libraries like Socket.IO.</li><li>• Create routes and controllers for handling user authentication, creating chat rooms, sending and receiving messages, fetching chat history, displaying chat rooms, deleting chat rooms, etc.</li><li>• Implement authentication using JWT (JSON Web Tokens) for user sessions.</li><li>• Implement CRUD operations for users, chat rooms, and messages in the database.</li></ul> |
| 3. | <b>Frontend Development</b> <ul style="list-style-type: none"><li>• Set up React.js for building the frontend.</li><li>• Create components for user authentication (login, signup), chat room creation, chat interface, chat room list, etc.</li><li>• Implement forms for user input (login/signup forms, message input).</li><li>• Use React Router for client-side routing.</li><li>• Integrate WebSocket client to communicate with the server for real-time messaging.</li></ul>  |
| 4. | <b>Database Management</b> <ul style="list-style-type: none"><li>• Design and create the database schema for users, chat rooms, and messages.</li><li>• Set up MongoDB and establish a connection with Node.js.</li><li>• Implement database queries and operations using an ORM like Mongoose.</li></ul>  |
| 5. | <b>Testing</b> <ul style="list-style-type: none"><li>• Write unit tests and integration tests for backend APIs and frontend components.</li><li>• Test the application for edge cases, error handling, and performance.</li></ul>  |
| 6. | <b>Deployment</b> <ul style="list-style-type: none"><li>• Deploy the application to a hosting service like Heroku or AWS.</li><li>• Set up continuous integration and deployment using GitHub Actions or other CI/CD tools.</li><li>• Configure environment variables for sensitive information like database credentials and JWT secrets.</li></ul>   |

## 7. **Documentation**

- Write comprehensive documentation for the project, including setup instructions, API documentation, and architecture overview.
- Document the codebase with comments and explanations for better maintainability.

## 8. **Maintenance and Updates**

- Monitor the application for bugs and security vulnerabilities.
- Provide regular updates and improvements based on user feedback.
- Keep dependencies up-to-date to ensure compatibility and security.

Throughout the development process, make sure to follow best practices for security, scalability, and performance. Also, consider usability and user experience to create an intuitive and engaging messaging application.