

Frontend Developer: ChatGPT like (or better) Chat Interface

Instructions:

1. **Create a "ChatGPT-like Chat Interface"** using **React.js** and **Next.js** as the frontend framework.
2. You should utilize **HTML/CSS** for structuring and styling the application. You can choose any CSS library (e.g., Bootstrap, Tailwind CSS, Material-UI or any other component library) to assist with the design and layout.
3. The focus should be on building a clean, responsive UI that mimics the functionality and design of a basic chat interface like ChatGPT.

Application Overview:

The goal of this application is to simulate a chat interface where users can send and receive messages in a conversation format. The application should include the following features:

Features:

- **Home View**
 - A single-page application layout displaying the chat interface.
 - A conversation area where all chat messages between the user and the system are displayed in a chat bubble format.
 - A text input area where the user can type messages.
 - A "Send" button to submit the message.
 - Display a simple mock response from the system for each user input (you can hardcode responses).
- **Chat Interface Functionality**
 - As a user, I should be able to type a message in the input box and click "Send" to see my message appear in the chat window.
 - After sending a message, a mock response should appear after a brief delay, simulating a conversation.
- **Integrate the API with LLM Backend**
 - Integrate the Chat interface with a 3rd party LLM API and provide a real chat experience utilizing any of free (OpenAI, LLama hosted, Groq or any free APIs)
- **Left Pane**
 - Historical Chats
 - Every chat should scroll automatically to the latest message when a new message is added
 - A "settings" menu item should be displayed in the bottom left corner of the pane

Bonus Features: To impress us, consider adding the following optional features:

- **Message History:**

- Store the chat history in the browser's local storage so that when the user refreshes the page, the conversation remains intact.
- **Typing Indicator:**
 - Show a typing indicator (e.g., "System is typing...") before the system's response appears.
- **Theming:**
 - Implement light and dark mode toggles for the chat interface.
- **Responsiveness:**
 - Ensure the interface is fully responsive and works seamlessly on both desktop and mobile devices.
- **Streaming:** Since OpenAI or any LLM API may be slow, you can implement a streaming chat like chatgpt

What We Are Looking For:

- **UI/UX Design:**
 - The chat interface should be intuitive and visually appealing. Pay attention to spacing, alignment, and responsiveness.
- **Clean, Maintainable Code:**
 - Your code should be well-organized, modular, and easy to read. Use appropriate abstractions where necessary.
- **Front-End Proficiency:**
 - Demonstrate your expertise in React.js and Next.js, along with your understanding of modern CSS techniques.

How to submit the assignment [Mandatory] ?

- Include a brief video demo (5 minutes max). You can upload the video to a Google Drive and ensure to have it public access, give us this URL over email.
- Zip your code and upload it to the same folder in google drive for us to access & review. Ensure to have readme instructions on how to run the application locally
- Be prepared to discuss your implementation decisions in the review call
- Unit tests are not required for this assignment.
- Ensure the application works well in the latest version of Chrome.

We are excited to see how you approach building this chat interface.

Most importantly, your originality should shine through the UI.

Good luck!