Homework - Create a full-stack clone of ChatGPT with streaming functionality using free LLMs

Objective: Implement a clone of ChatGPT using free LLMs such as Gemini, Cohere or any other with streaming functionality.

Tasks:

1. **Backend:**
   - Simple backend using Express.js (using a backend/ai week boilerplate).
   - Integrate the chosen free LLM (Gemini/Cohere, etc) for text generation. OpenAI's GPT is allowed too.
   - Create endpoints for handling chat requests and streaming responses using three layer architecture (controller, service, router)
2. **Frontend Setup:**
   - Simple frontend using React/Next.js.
   - Implement a chat interface to send user messages to the backend and display responses.
3. **Streaming Functionality:**
   - Implement real-time streaming of LLM responses from the backend to the frontend.
4. **Database:**
   - Set up a database to store chat history.
   - Implement automatic synchronization of chat data between the backend and the database.

Levels:

1. **Level 1:** Basic ChatGPT Clone
   - Ensure the LLM request works fine and retrieves a response.
   - Display the response in the chat interface.
2. **Level 2:** Streaming Functionality
   - Implement real-time streaming of LLM responses.
   - Ensure the frontend displays streaminCreate a g responses dynamically as they are received from the backend.
3. **Level 3:** Database storage
   - Set up a database (e.g MongoDb with mongoose orm or PostgreSQL with Prisma orm) to store chat history.
   - Implement automatic synchronization of chat data between the backend and the database.

    - Ensure chat history is persisted and can be retrieved upon request.

Setup:

- Fork https://github.com/effuone/websockets-example
- Remove live-coding-specific functionality (e.g roadmaps, JSON streaming functionality)
- Submit to AirTable (available at Notion)