CHAT INTERFACE

TanStack Query

Backend API

Send message to the backend

User inputs a message in mess chat interface

Frontend sends message to backend API

CHAT CONTROLLER

Express JS

Processes Request

Generates response (with Fallback Mechanism)

OpenAI API

DRIZZLE ORM

PostgreSQL

Chat history is updated in the UI

OUTPUT

Tech Flow Diagram

Frontend Tech Stack: React + TypeScript + TanStack Query+ Tailwind + Shadcn UI

The user submits a message, and React captures the input. TanStack Query triggers a POST request to the backend API with the user's message. The UI remains responsive and can optimistically render the message while waiting for the response.

Backend Tech Stack: Express + Drizzle ORM + PostgreSQL + OpenAI API

The Express chat controller receives the user's message. The controller calls the OpenAI service, which sends the input to the OpenAI API. If the API returns a response, it is returned to the controller. If not, a fallback response is generated. The final AI-generated response is :
- Stored in PostgreSQL via Drizzle ORM for history/logging
- Returned to the frontend as the response payload

KEY   Backend   Frontend