Problem Statement: LLM-Powered Chatbot with FastAPI and SQL Integration

You are required to build a simple **LLM-powered chatbot application** using **Python (FastAPI)** that allows users to input a free-text query. The system should use a **Groq-hosted LLM endpoint** (such as **Llama3.1** or **Mistral-7B**) to process the user's query and generate responses based on data stored in a **SQL database**.

Requirements

1. Tech Stack

- **Backend:** FastAPI
- **Frontend:** ReactJS (basic form to take user input and display response)
- **Database:** SQLite3 or PostgreSQL (your choice)
- **LLM Endpoint:** Groq (free tier) using Llama 3.1 or Mistral-7B model

2. Database Schema

You must create a basic customer database with the following fields:

- `customer_id` (Primary Key)
- `name` (Text)
- `gender` (Text)
- `location` (Text)

Seed it with sample entries (at least 5).

3. Expected Functionality

- The user inputs a **natural language query** via the UI.
  Example: "Show me all female customers from Mumbai"
- The backend (FastAPI) sends the query to the LLM via the Groq endpoint.
- The LLM must interpret the query and generate a corresponding **SQL query**.

- The backend should **execute the SQL query**, retrieve the results, and return them to the user.

- The frontend displays the formatted results.

Bonus Points

- Add logging for incoming queries and SQL generation for debugging.

- Include simple error handling (e.g., for invalid queries or SQL errors).

- Use environment variables or a `.env` file to manage API keys and config.

- Secure the API (optional, can be just a token-based check).

Deliverables

- FastAPI backend code

- Sample SQLite/PostgreSQL schema and seed script

- ReactJS frontend (or Postman test instructions if frontend is not included)

- ReadMe file with setup and run instructions

- **All code must be committed to your public GitHub profile and shared as a link.**