Chatbot App Documentation

Mehdi Ben Fekhta

This project is a complete chatbot application using Flutter for the frontend, Node.js + TypeScript with Prisma for the backend, and PostgreSQL on Neon for the database. The goal is to allow users to chat with an intelligent assistant while saving conversation history and providing a responsive interface.

1. Technologies Used

- Frontend: Flutter (UI, state management, API integration)

- Backend: Node.js + TypeScript (REST API, business logic)

- ORM: Prisma

- Database: PostgreSQL on Neon

2. Main Features

- User authentication (Google/Email)

- Responsive chat interface

- Conversation history

- Optional multilingual support

- Secure communication between frontend and backend

3. Architecture

Flutter Frontend <--> Node.js/TypeScript Backend <--> Prisma <--> PostgreSQL (Neon)

- Frontend: handles UI and user interactions

- Backend: provides API and business logic

- Prisma: ORM for database

- PostgreSQL: persistent data storage

4. Installation

Frontend (Flutter)

1. Install Flutter from https://docs.flutter.dev/get-started/install

2. Clone the project:

git clone <https://github.com/mehdibf100/ScaleX-Innovation-chatbot-AI-test.git>

cd frontend

3. Install dependencies: flutter pub get

4. Run the app: flutter run

5. APK version available: Download APK : <https://drive.google.com/file/d/14V10Ut7pyyLFGS4AHxzEOjOc-05RWZ46/view?usp=sharing>

Backend (TypeScript + Prisma)

1. Install Node.js >=18 and npm/yarn

2. Navigate to the backend folder: cd backend

3. Install dependencies: npm install or yarn install

4. Create .env file:

DATABASE\_URL="postgresql://neondb\_owner:npg\_bYmx4TiFC1NA@ep-purple-smoke-ad1h9y41-pooler.c-2.us-east-1.aws.neon.tech/neondb?sslmode=require&channel\_binding=require"

PORT=4000

5. Generate Prisma client and apply migration:

npx prisma generate

npx prisma migrate dev --name init

6. Run the server in development: npm run dev or yarn dev

5. Configuration

- Backend: .env with DATABASE\_URL and PORT

- Frontend: API\_BASE\_URL in lib/config.dart

6. Useful Scripts

- Backend: npm run dev, npm run build, npx prisma studio

- Frontend: flutter run, flutter build apk, flutter build ios

7. Backend Deployment

* The backend is already deployed on Railway.