**Dental Appointment Chatbot**

This is the demo of Dental Appointment chatbot but it carries the full stack architecture.

**System Workflow**

* **User Registration / Login**
  + A user first registers or logs in using their email and password.
  + The backend verifies credentials and returns a **JWT token** for authentication.
  + This token is stored temporarily on the frontend for secure API requests.
* **Chat Interaction**
  + Once logged in, the user can send messages through a chat interface.
  + Each message is sent to the **Flask API**, which processes the input and generates a chatbot reply.
  + The chatbot may provide answers related to appointment booking, dental services, or other demo responses.
* **Database Storage**
  + All user details (name, email, hashed password) are stored in a **SQLite database**.
  + SQLite was chosen for simplicity and portability during demo/testing.
* **Frontend–Backend Communication**
  + The frontend communicates with the backend through **REST API endpoints** (/api/auth/login, /api/auth/register, /api/chat).
  + Authentication is handled using a **Bearer token** in the HTTP header.

**Tech Stack Used:**

|  |  |  |
| --- | --- | --- |
| **Component** | **Technology** | **Role** |
| **Frontend** | **Next.js (React + TypeScript)** | Provides a clean chat interface and handles login/register forms. |
| **Backend API** | **Flask (Python)** | Manages authentication, chat message handling, and database communication. |
| **Database** | **SQLite** | Stores user credentials securely (email, password hash, name) |
| **Auth** | **JWT (JSON Web Tokens)** | Used for session management and secure API communication. |
| **Password Hashing** | **bcrypt** | Encrypts user passwords before storing them. |

**Note:**  
**Before running the project, make sure to install all required dependencies.**

* Run **npm install** to install all Node.js packages for the frontend.
* Install **SQLite** separately using:

**npm install sqlite**

* Additionally, ensure that all required Python libraries for the Flask microservice are installed by running:

**pip install -r requirements.txt**