Take-Home Assessment: Patient Dashboard and Chatbot Access

Objective: To assess the candidate's ability to build a patient dashboard with access to a chatbot, including session management, interaction logging, and displaying linked doctor information.

Task:

1. **Create a Patient Dashboard:**
    - Implement a dashboard interface where patients can view their profile, interaction history, and linked doctor information.
    - Display a list of past interactions with the chatbot.
2. **Integrate a Chatbot Interface:**
    - Embed a chatbot widget into the dashboard.
    - Allow patients to submit text queries to the chatbot.
    - Display the chatbot's responses in real-time.
3. **Session Management:**
    - Implement session tracking to maintain the context of patient interactions.
    - Store interaction histories in a database (e.g., DynamoDB or PostgreSQL).
4. **Database Schema:**
    - Design a database schema to store patient profiles, interaction histories, and linked doctor information.
    - Ensure efficient retrieval of past interactions and linked doctor details.

Examples

1. Database Schema Design:

- **Patient Interactions Table:**

```
CREATE TABLE PatientInteractions (
    InteractionID SERIAL PRIMARY KEY,
    PatientID INT,
    DoctorID INT,
    Query TEXT,
    Response TEXT,
    InteractionDate TIMESTAMP,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)
);
```

2. Sample Dummy Data:

- **Patient Interactions:**

```
[
  {
    "InteractionID": 1,
    "PatientID": 1,
    "DoctorID": 1,
    "Query": "What are the symptoms of a heart attack?",
    "Response": "The symptoms of a heart attack can include chest
pain, shortness of breath, and nausea.",
    "InteractionDate": "2023-07-01T13:00:00Z"
  },
  {
    "InteractionID": 2,
    "PatientID": 2,
    "DoctorID": 2,
    "Query": "What should I do if I have a severe headache?",
    "Response": "If you have a severe headache, you should rest in a
quiet, dark room and take over-the-counter pain medication if
needed.",
    "InteractionDate": "2023-07-02T13:00:00Z"
  }
]
```

Technical Requirements:

- Frontend: React (recommended)
- Backend: Node.js with Express (recommended) or any backend technology of your choice
- Database: PostgreSQL, MySQL, MongoDB, or DynamoDB (candidate's choice)

Deliverables:

- Source code for the frontend and backend.
- Database schema design.
- Screenshots, video, or PowerPoint presentation demonstrating the output.

Evaluation Criteria:

- Functionality: Does the patient dashboard and chatbot interface work as expected? Are linked doctor profiles displayed correctly?
- Session Management: Is the session tracking and interaction logging implemented correctly?
- Code Quality: Is the code well-organized and documented?
- User Experience: Is the UI user-friendly and responsive?
- Integration: Is the chatbot integrated seamlessly into the dashboard?

Additional Instructions:

- Feel free to make any assumptions necessary to complete the assessment.
- Use any examples or dummy data you find suitable.
- Utilize GenAI tools or any other resources that can help you.
- This assessment is open-ended; focus on showcasing your skills and creativity.