Hiring Assignment - SDE Interns

SDE Intern Assignment: EMR Feature Implementation

This assignment combines the requirements of the Frontend and Backend tasks, focusing on a single goal: making the **Appointment Management View** functional by implementing the necessary data layer.

Objective

The goal is to implement a functional, end-to-end feature: **Appointment Scheduling and Queue Management (Feature B)**. You must design the data contract, implement the backend service, and integrate it with the provided frontend component to handle filtering and state changes.

Core Technology Stack

- **Frontend (UI):** React and Tailwind CSS
- **Backend (API/Logic):** Python 3.x for Lambda, simulating the use of AppSync/GraphQL.
- **Data Layer:** PostgreSQL (simulated via Python classes/dictionaries).

Task 1: Backend Service Implementation (The API Contract)

You must create a Python class or file (appointment_service.py) that contains the core logic for the **Scheduling & Queue Microservice (3.3)**.

1. **Data Mocking:** Create a hardcoded list of **at least 10 mock appointments** (simulating an Aurora fetch). Each item must include fields necessary for the frontend: name, date, time, duration, doctorName, status (Confirmed, Scheduled, Upcoming, Cancelled), and mode.
2. **Query Function:** Implement a Python function, get_appointments(filters), that accepts optional arguments (date: String, status: String) and filters the mock list accordingly.
3. **Mutation Function:** Implement a Python function, update_appointment_status(id, new_status), that updates the status of an appointment in your mock data. In comments, explain where this action would trigger an **AppSync Subscription** and an **Aurora transactional write**.

Task 2: Frontend Integration and Functionality

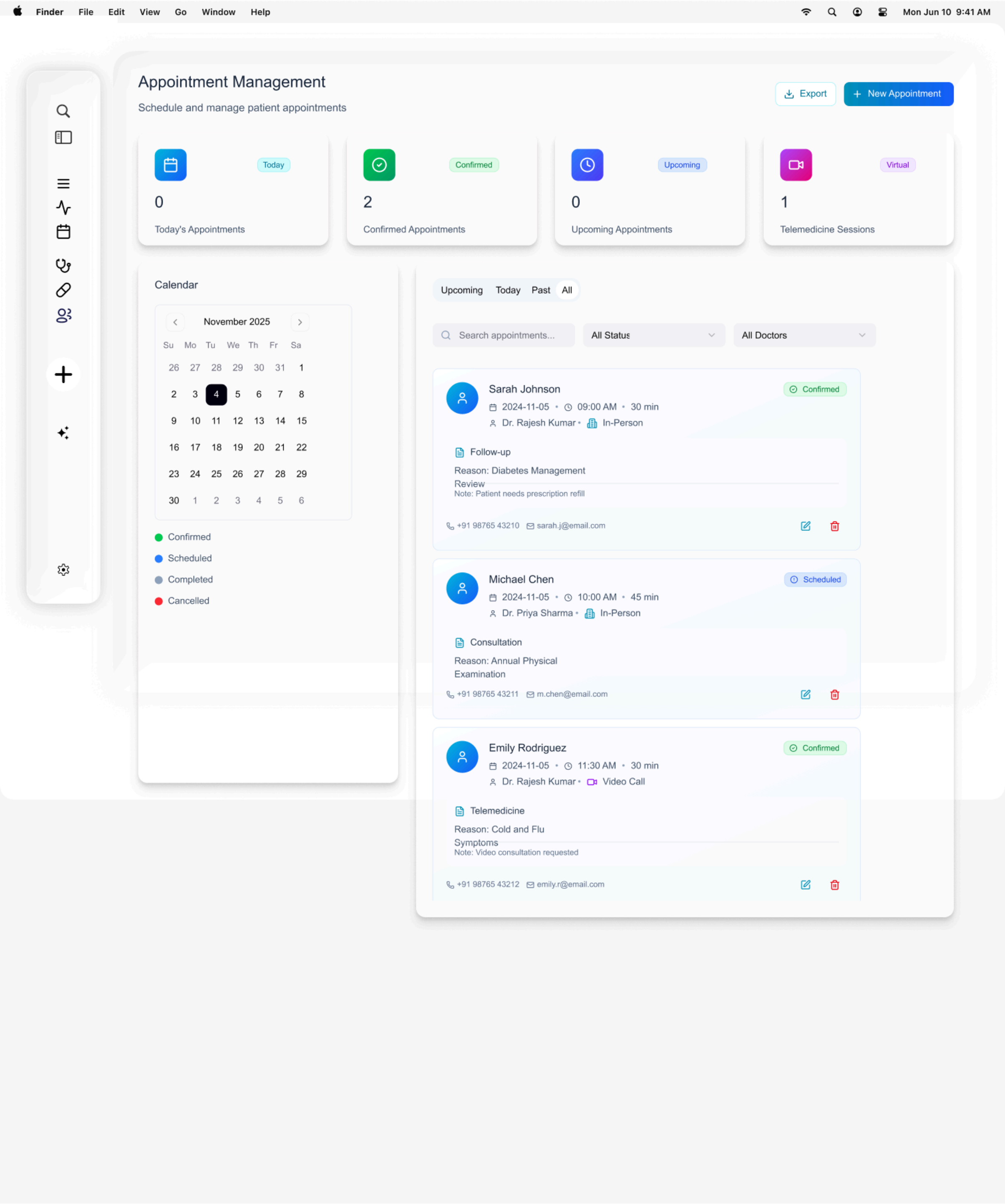
You must integrate the logic from **Task 1** into the provided EMR_Frontend_Assignment.jsx file to make the Appointment Management View functional.

1. **Data Fetching:** In the AppointmentManagementView component, use a React hook (useState/useEffect) to initialize the component with data fetched from your **Python get_appointments() function** (simulated by importing and calling the function directly).
2. **Calendar Filtering:** Implement the click handler for the **Calendar Widget**. When a date is clicked:
   - Set the local state for the selectedDate.
   - Call your **Python get_appointments() function**, passing the selectedDate as a filter.
   - Update the list of appointments displayed in the main right panel.
3. **Tab Filtering:** Implement logic for the **Tabs (Upcoming, Today, Past)**. When a tab is selected, filter the displayed appointments based on the appointment's status or date relative to today.
4. **Status Update:** Implement the functionality to update an appointment status (e.g., clicking a button next to an appointment card). This should call your **Python update_appointment_status() function** and immediately refresh the local component state to reflect the change (simulating real-time UI updates).

Submission Guidelines

1. **Single Repository:** Provide a link to a Git repository containing the complete project.
2. **Frontend File:** The primary implementation must be in the provided **EMR_Frontend_Assignment.jsx** file (or a similar .jsx file if using a local React setup).
3. **Backend File:** A separate Python file named **appointment_service.py** containing the functions defined in Task 1.
4. **Live Link:** A working, publicly hosted link (e.g., Vercel, Netlify) to the application.
5. **Technical Explanation:** A brief README explaining the **GraphQL query structure** you designed for the getAppointments function and how your Python functions ensure data consistency upon update.
6. The Assignment is to be completed in 3 days from the date it is shared.

UI Mockups for reference



Appointment Management

Schedule and manage patient appointments

Today	Confirmed	Upcoming	Virtual
0	2	0	1
Today's Appointments	Confirmed Appointments	Upcoming Appointments	Telemedicine Sessions

Calendar

November 2025

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

- 🟢 Confirmed
- 🔵 Scheduled
- ⚪ Completed
- 🔴 Cancelled

Sarah Johnson — Confirmed

2024-11-05 • 09:00 AM • 30 min

Dr. Rajesh Kumar • In-Person

Follow-up

Reason: Diabetes Management Review

Note: Patient needs prescription refill

+91 98765 43210 sarah.j@email.com

Michael Chen — Scheduled

2024-11-05 • 10:00 AM • 45 min

Dr. Priya Sharma • In-Person

Consultation

Reason: Annual Physical Examination

+91 98765 43211 m.chen@email.com

Emily Rodriguez — Confirmed

2024-11-05 • 11:30 AM • 30 min

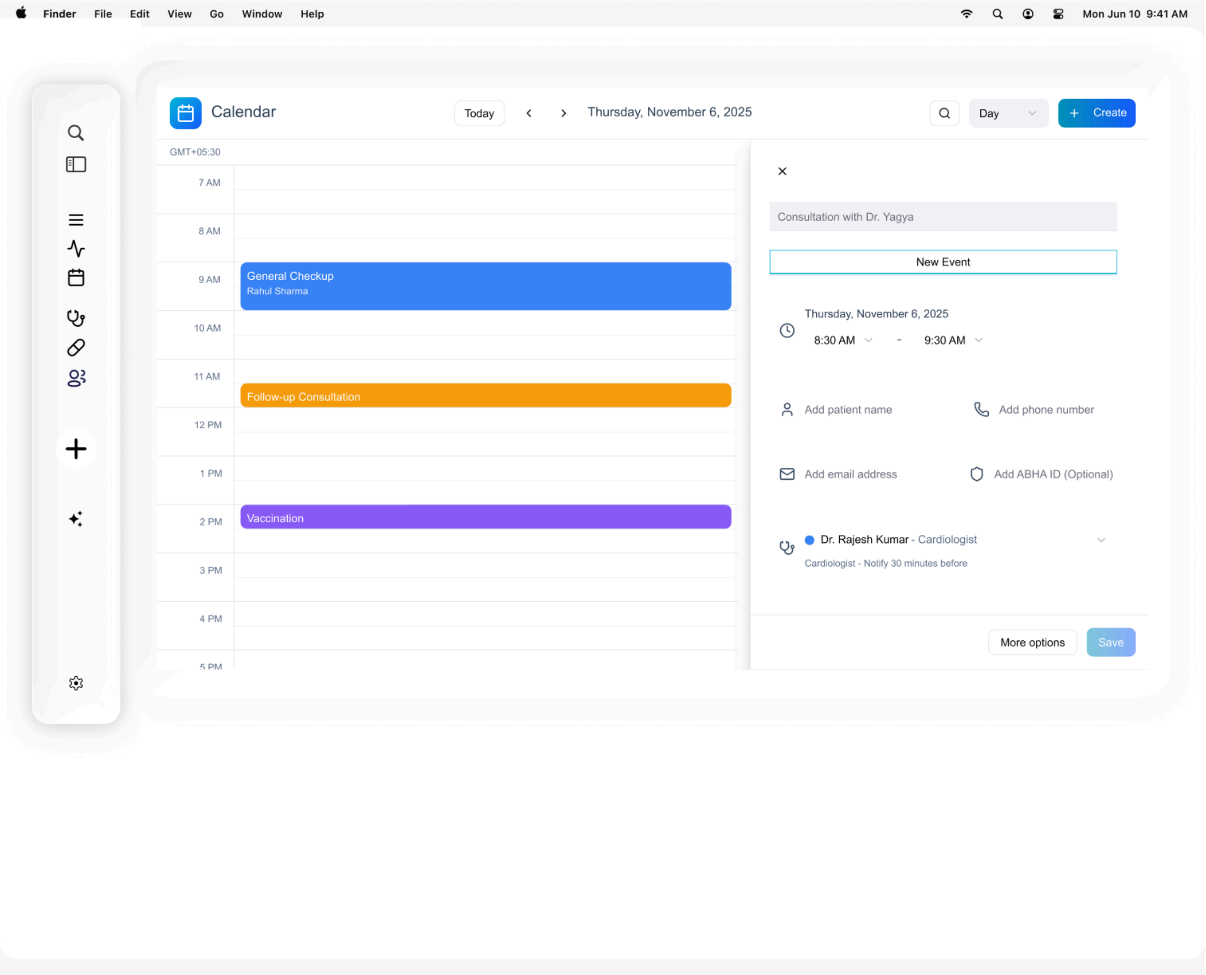
Dr. Rajesh Kumar • Video Call

Telemedicine

Reason: Cold and Flu Symptoms

Note: Video consultation requested

+91 98765 43212 emily.r@email.com



Thursday, November 6, 2025

GMT+05:30

7 AM

8 AM

9 AM General Checkup
Rahul Sharma

10 AM

11 AM Follow-up Consultation

12 PM

1 PM

2 PM Vaccination

3 PM

4 PM

5 PM

Consultation with Dr. Yagya

New Event

Thursday, November 6, 2025
8:30 AM - 9:30 AM

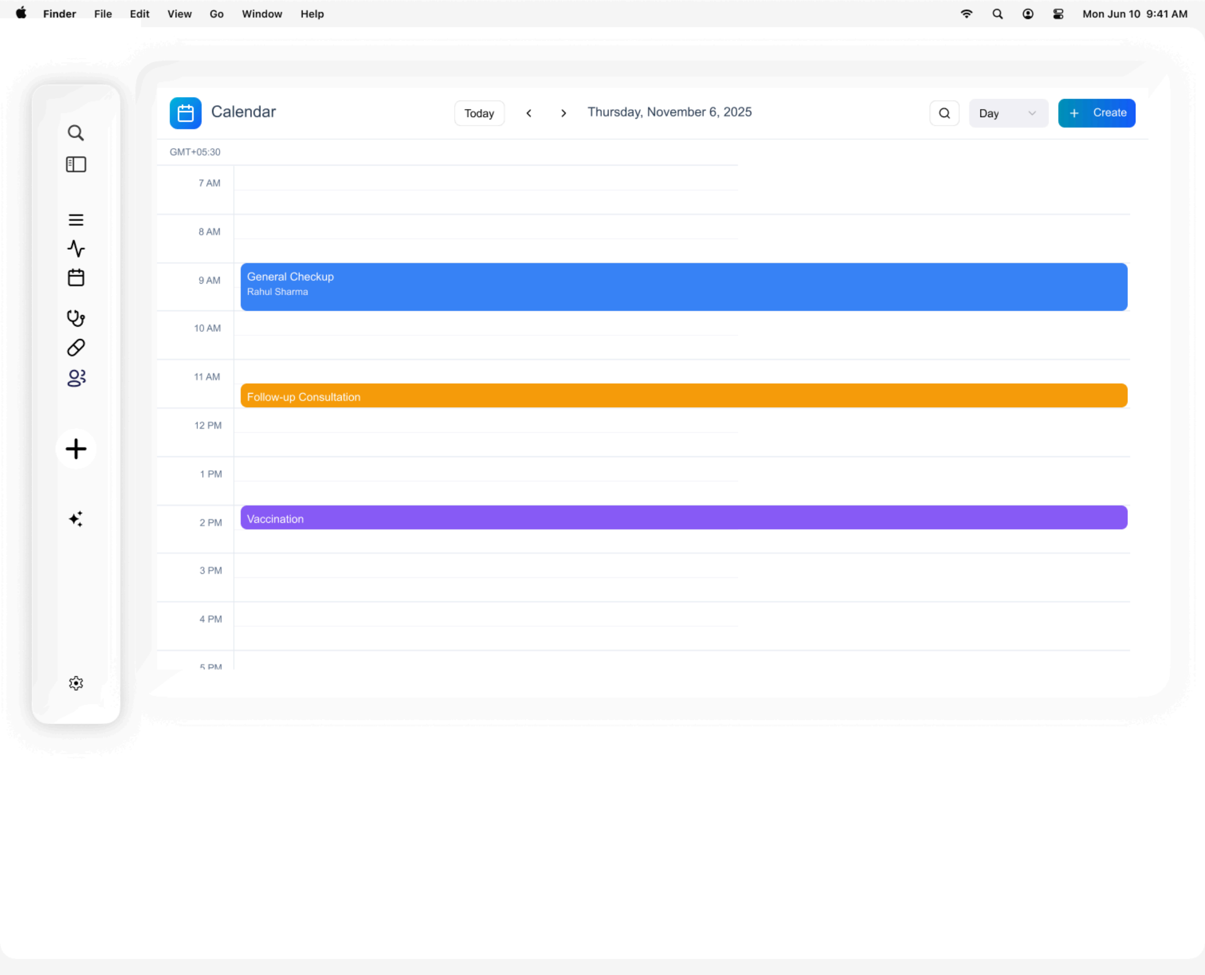
Add patient name

Add phone number

Add email address

Add ABHA ID (Optional)

Dr. Rajesh Kumar - Cardiologist
Cardiologist - Notify 30 minutes before



General Checkup
Rahul Sharma

Follow-up Consultation

Vaccination