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.The feature must support creating an appointment via backend API (no frontend-only state mutations)

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**.

4. **Create Function:** Implement a Python function create_appointment(payload) that:

- Validates required fields: patientName, date, time, duration, doctorName, mode
- Generates a unique appointment id on the backend
- Sets default status (Scheduled) unless explicitly passed
- Prevents time conflicts for the same doctor on the same date (overlap detection)
- Returns the created appointment object

5. **Delete Appointment Function (optional but strong signal)**

Implement delete_appointment(id) and update your mock data.

6. **Explain data consistency**

explain how you would enforce consistency in a real system (transaction, unique constraints, idempotency key).

Example contract

- `get_appointments(filters: { date?: string, status?: string, doctorName?: string }) -> Appointment[]`
- `create_appointment(input: CreateAppointmentInput) -> Appointment`
- `update_appointment_status(id: string, new_status: string) -> Appointment`
- `delete_appointment(id: string) -> boolean`

Task 2: Frontend Integration and Functionality

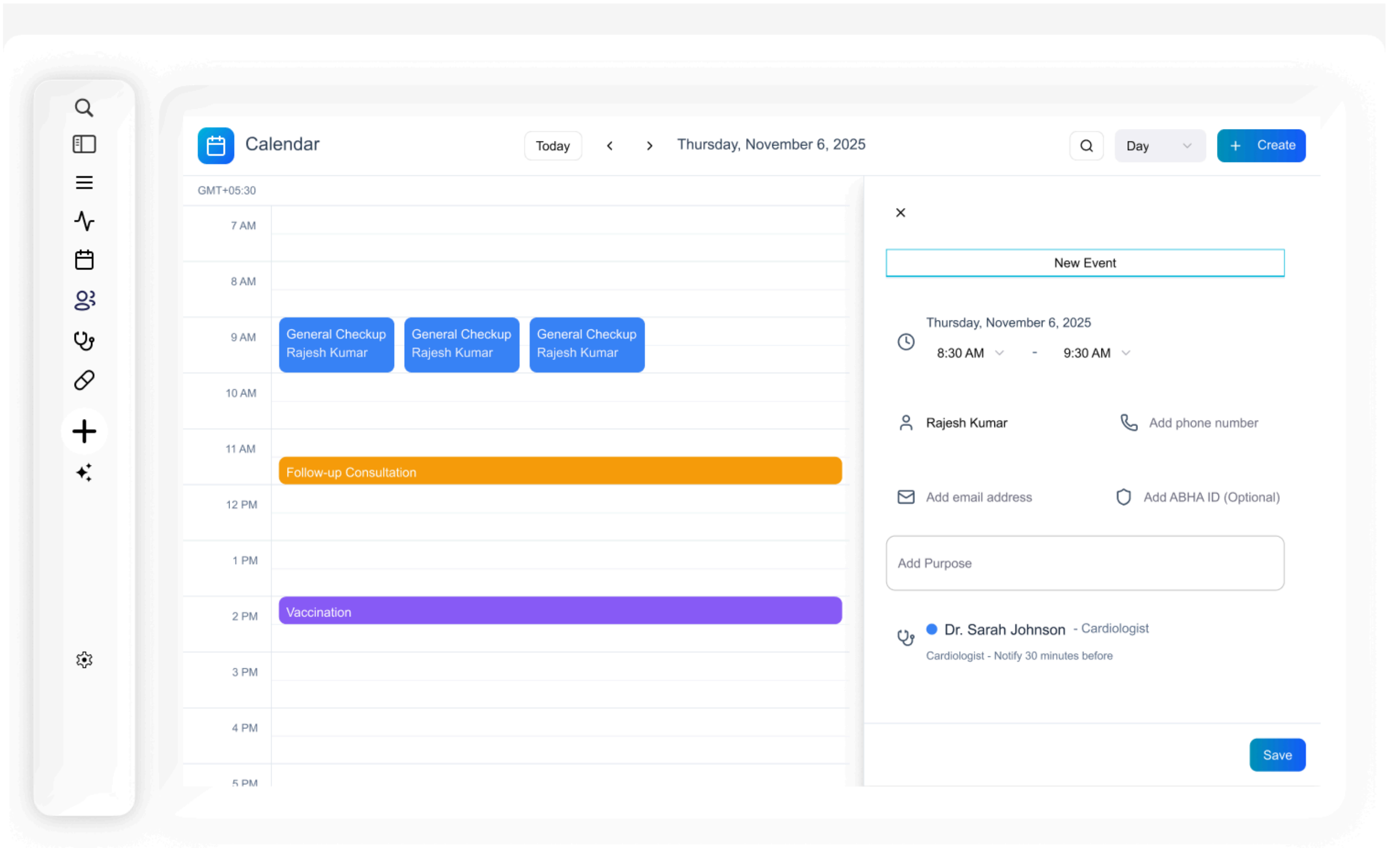
You must integrate the logic from **Task 1** into the provided EMR Frontend_Assignment.jsx file to make the Appointment Management View functional.
1. **Data Fetching:** In the AppointmentManagementView component, use a React hook (useState/useEffect) to initialize the component with data fetched from 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).
5. Create Appointment: Implement the "New Appointment" button and form.

- On submit, call `create_appointment(payload)` from `appointment_service.py` On success, refresh the appointment list by calling `get_appointments()` again (or append returned object to state if you can justify consistency)
- Do not directly mutate the appointments array in the component without calling backend

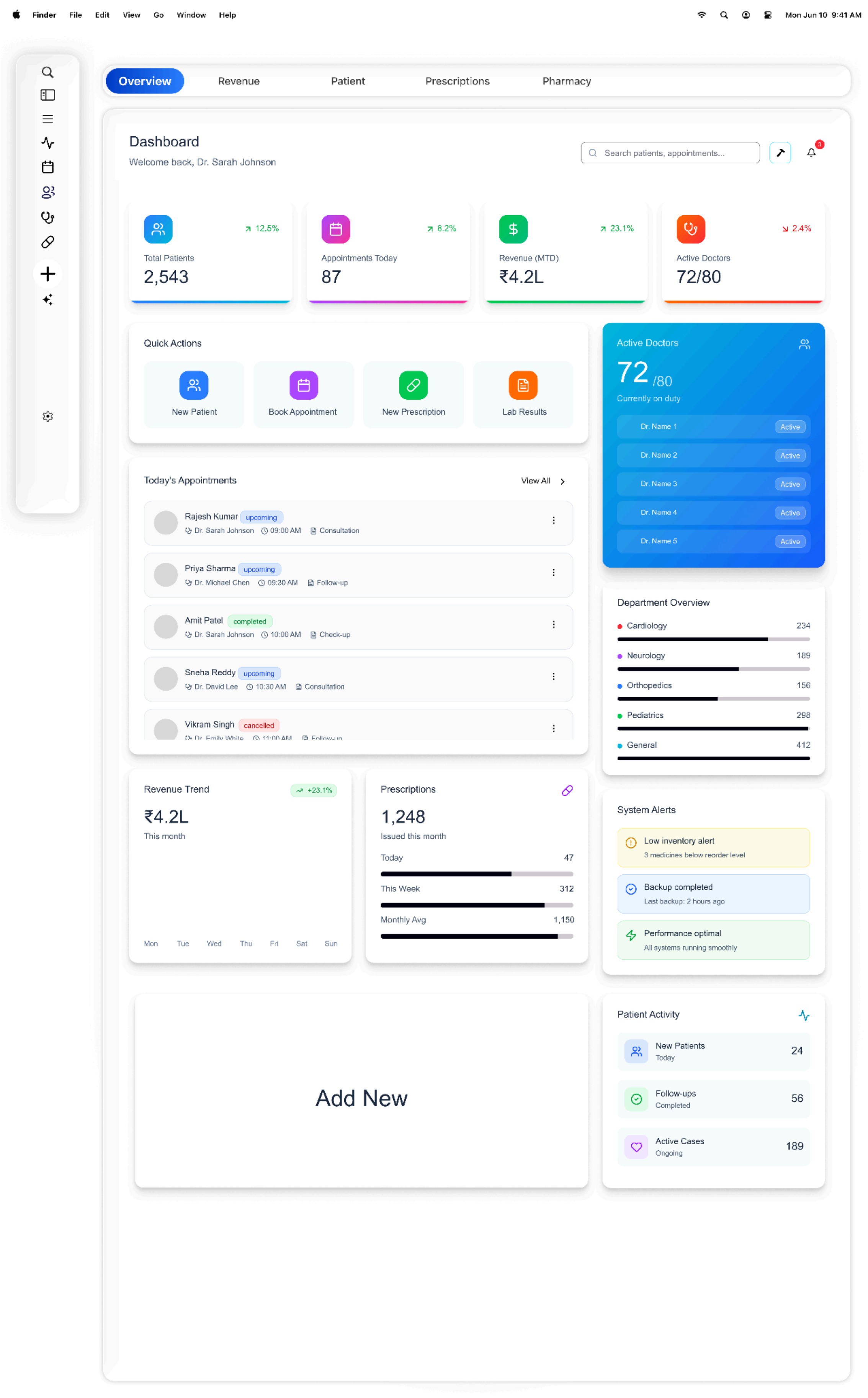
Submission Guidelines
1. **Single Repository:** Provide a link to a Git repository containing the complete project.
2. **Frontend File:** The primary implementation must be in the provided **EMR_Frontend__Assignment.jsx** file (or a similar .jsx file if using a local React setup).
3. **Backend File:** 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



General Checkup Rajesh Kumar
General Checkup Rajesh Kumar
General Checkup Rajesh Kumar
Follow-up Consultation
Vaccination

New Event
Thursday, November 6, 2025
8:30 AM - 9:30 AM
Rajesh Kumar
Add phone number
Add email address
Add ABHA ID (Optional)
Add Purpose
Dr. Sarah Johnson - Cardiologist
Cardiologist - Notify 30 minutes before



Dashboard

Welcome back, Dr. Sarah Johnson

↗ 12.5%	↗ 8.2%	↗ 23.1%	↘ 2.4%
Total Patients	Appointments Today	Revenue (MTD)	Active Doctors
2,543	87	₹4.2L	72/80

Quick Actions

Today's Appointments

View All

- Rajesh Kumar upcoming — Dr. Sarah Johnson · 09:00 AM · Consultation
- Priya Sharma upcoming — Dr. Michael Chen · 09:30 AM · Follow-up
- Amit Patel completed — Dr. Sarah Johnson · 10:00 AM · Check-up
- Sneha Reddy upcoming — Dr. David Lee · 10:30 AM · Consultation
- Vikram Singh cancelled — Dr. Emily White · 11:00 AM · Follow-up

Active Doctors

72 /80

Currently on duty

Dr. Name 1	Active
Dr. Name 2	Active
Dr. Name 3	Active
Dr. Name 4	Active
Dr. Name 5	Active

Department Overview

Cardiology	234
Neurology	189
Orthopedics	156
Pediatrics	298
General	412

Revenue Trend

+23.1%

₹4.2L

This month

Prescriptions

1,248

Issued this month

Today	47
This Week	312
Monthly Avg	1,150

System Alerts

Low inventory alert
3 medicines below reorder level

Backup completed
Last backup: 2 hours ago

Performance optimal
All systems running smoothly

Add New

Patient Activity

New Patients Today	24
Follow-ups Completed	56
Active Cases Ongoing	189