AfyaCall Field Project - Health Appointment Booking System

Project Overview

This project is about building a **Health Appointment Booking System** for AfyaCall. Patients should be able to **book consultations with doctors**, doctors should manage their appointments, and admins should oversee the system.

The goal is to expose you to **real-world problem solving with coding** while building something useful in the health domain.

Deliverables

1. A fully functional web application *(Laravel/Django preferred).*
2. User roles with different capabilities (Patient, Doctor, Admin).
3. Appointment booking flow from request → approval → confirmation.
4. Notification system (Email/SMS) to alert patients and doctors.
5. Dashboards for patients, doctors, and admins.
6. Database schema & migrations.
7. Documentation (*README* + system overview).
8. A working deployment (local/Cloud server).
9. Final demo presentation to the AfyaCall team.

Core Modules

1. Authentication & Roles

- Patient: register, login, book appointments.
- Doctor: manage schedule, accept/reject appointments.
- Admin: manage doctors, specialties, and system setup.

2. Doctor & Specialty Management

- Admin can add doctors and assign them specialties / Categories (psychiatry, orthopaedics, etc.).
- Doctors can set their available time slots.

3. Appointment Booking System

- Patients can view available doctors by specialty.
- Patients can request an appointment by selecting doctor, date, and time.
- Doctors can accept or reject requests.
- Patients can cancel or reschedule bookings before a certain cutoff time.
- Doctors can also mark unavailable slots.
- System updates availability automatically.
- This teaches state management and improves real-world usability

4. **Notifications & Alerts**

- Patients receive a confirmation or rejection notice.
- Send automatic reminders to patients (via SMS/Email) 24 hours or a few hours before their appointment.
- Doctors also get notified of their upcoming schedule.
- Notification channels:
  - Email
  - SMS

5. **Dashboards**

- *Patient Dashboard:* upcoming appointments, booking history.
- *Doctor Dashboard:* pending requests, confirmed appointments.
- *Admin Dashboard:* overview of doctors, specialties, system stats.

6. **Reports & Analytics**

- Admin can generate simple reports:
  - Number of appointments per doctor/specialty.
  - Daily/weekly/monthly bookings
  - Most requested specialties.

- Display data in charts/graphs on the admin dashboard.

7. **Deployment**

- Code hosted in a public GitHub repository.
- Working application deployed locally (preferably Linux server).
- Proper documentation on how to run the system.

Learning Goals

Throughout this project, you will gain hands-on experience with:

1. Structuring a software project
2. Building authentication & role-based access control.
3. Designing and implementing a relational database.
4. Working with CRUD operations in real use cases.
5. Integrating with external APIs (Email/SMS notifications).
6. Designing dashboards and user interfaces.
7. How to Automate Tasks ie: schedule jobs / cronjobs
8. Data Aggregation & Visualization through dashboards
9. Data Analysis from reports
10. Writing unit tests for core booking logic.
11. Deploying applications in a local/Linux environment.
12. Documenting and presenting a complete software solution.