**Hackathon Project Documentation: Revolutionizing HealthCare Scheduling: A Seamless Appointment Booking Solution**

**Team Name:** How I Met Your Bug  
**Project Name:** Mobile application for scheduling an appointment for a patient with a doctor.  
**Hackathon Name:** Veersa Technologies Hackathon  
**Date:** 5th March, 2025

**1. Introduction**

**1.1 Problem Statement**

Doctors are lifesavers, and access to healthcare should be quick and hassle-free. However, patients often face difficulties in booking appointments due to limited availability, long waiting times, and inefficient scheduling.

**1.2 Objective**

Our goal is to develop a mobile application that enables patients to book doctor appointments anytime, from anywhere, ensuring:

* Easy access to doctor availability based on specialty & location.
* Conflict-free appointment booking with instant confirmation.
* Reminder notifications with driving directions.
* Secure access and authentication for users.

**2. Features of the App**

**2.1 Core Features**

Doctor Availability by Specialty

* Patients can filter & view available doctors based on their specialization (e.g., Cardiologist, Dermatologist).

Find Doctors Near You

* Uses GPS location to find the nearest doctors, sorted by distance.

Appointment Booking & Confirmation

* Patients can select a doctor, choose an available time slot, and receive instant confirmation.

Conflict-Free Scheduling

* Ensures no overlapping bookings for doctors or patients.

Reminder Notification & Driving Directions

* Sends a reminder 1 hour before the appointment.
* Provides a Google Maps link for driving directions.

**3. Technology Stack**

Mobile Application:

|  |  |
| --- | --- |
| Component | Technology Used |
| Mobile App | Flutter |
| Backend | Firebase |
| Database | Firebase |
| Maps API | Google Maps API |
| Notifications | Flutter local notifications plugin |
|  |  |

Website:

|  |  |
| --- | --- |
| Component | Technology Used |
| Backend | Node.js / Express.js |
| Architecture | Model-View-Controller (MVC) |
| Frontend | EJS Templating, Tailwind CSS |
|  |  |

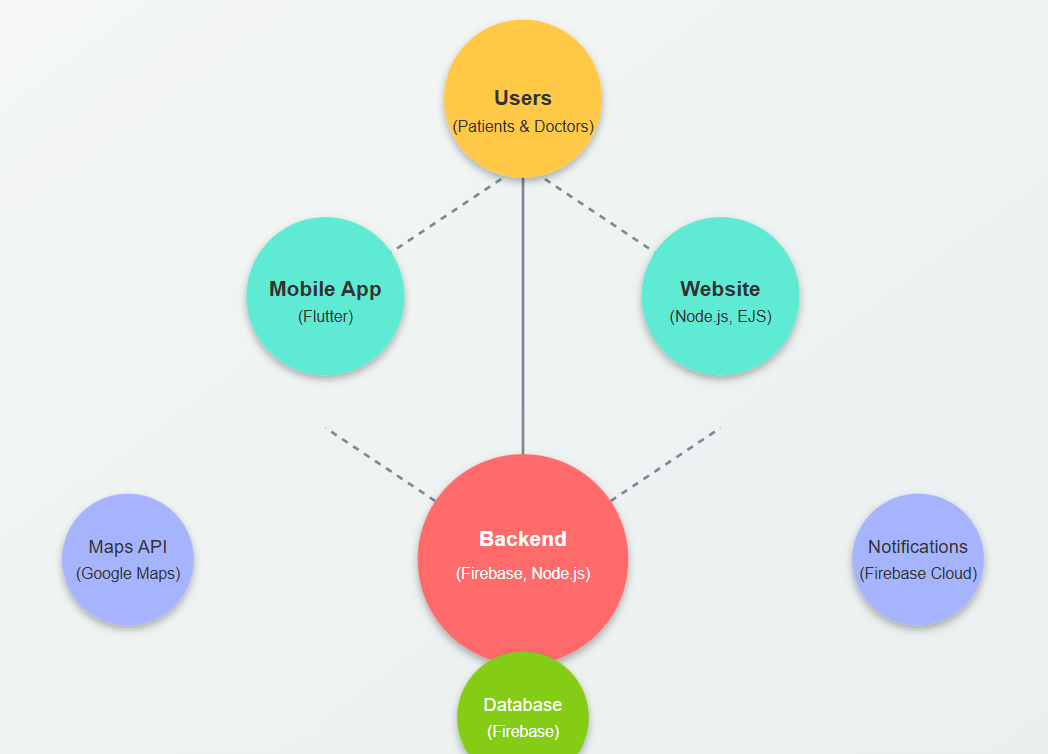
**4. System Architecture**

**4.1 Overview**

The system consists of:

* **User Interface (Mobile App & Website):** Patients and doctors interact via the app or web portal.
* **Backend Server:** Firebase (for the mobile app) and Node.js/Express.js (for the website) manage appointments, users, and notifications.
* **Database:** Firebase for storing doctor details, availability, and appointments.
* **Maps Integration:** Google Maps API for location-based doctor search and navigation.
* **Security Layer:** Implements user authentication, data encryption, and role-based access.

**4.2 Architecture Diagram**

****

**5. User Flow**

**5.1 Patient Flow**

1. Open the app → Login/Register
2. Search for a doctor by specialty or location
3. Select doctor & available time slot
4. Confirm appointment & receive notification
5. Get a reminder & directions before the appointment

**5.2 Doctor Flow**

1. Open app → Login/Register
2. Set availability schedule
3. Receive & manage appointment requests
4. Confirm or reschedule bookings
5. View daily appointment list

**6. Security Measures**

* Data Encryption: End-to-end encryption for sensitive data.
* Role-Based Access Control: Patients and doctors have different access levels.
* Secure API Calls: HTTPS and token-based authentication for backend requests.

**7. Key Challenges & Solutions**

| Challenge | Solution |
| --- | --- |
| Avoid conflicting appointments | Real-time appointment validation before booking |
| Finding doctors efficiently | GPS-based nearest doctor search |
| Ensuring smooth user experience | Intuitive UI/UX with easy navigation |
| Sending real-time notifications | Firebase Cloud Messaging for push notifications |
| Securing user data | Encryption and secure authentication |

**8. Future Enhancements**

🔹 Teleconsultation Feature – Allow video consultation with doctors.  
🔹 E-Prescriptions – Enable doctors to send digital prescriptions.  
🔹 Payment Integration – Add online payment for consultation fees.  
🔹 AI-based Doctor Suggestions – Recommend doctors based on symptoms.  
🔹 Multi-language Support – Enhance accessibility for diverse users.

**9. Testing & Deployment**

* Testing Approach:
* Unit Testing: Developers write automated unit tests to check individual functions.
* Integration Testing: Ensures different components work together correctly.
* User Acceptance Testing (UAT): End-users test the app to validate functionality and usability.
* Manual Test Cases: Documented test scenarios to verify UI, workflows, and expected outcomes.
* Automated API Testing: API endpoints are tested using tools like Postman or Jest to ensure reliability.
* Deployment Strategy:
  + Mobile App Deployment: Google Play Store / Apple App Store.
  + Backend Hosting: AWS / Firebase
  + CI/CD Pipeline: GitHub Actions / Jenkins for continuous deployment.

**10. Conclusion**

The Doctor Appointment Scheduling App provides a seamless experience for both patients and doctors, ensuring efficient appointment booking, real-time availability, and automated reminders.  
This solution aims to modernize healthcare accessibility and reduce waiting times by making doctor appointments instant and hassle-free.

**11. Links & References**

* GitHub Repository: git clone https://github.com/daksh-deep/How-i-met-your-bug