Project Title	**Doctor Appointment Booking System**
Technologies	MERN
Domain	Healthcare
Project Level	Difficult

Table

Table

4. Submission requirements:

1. Problem Statement:

The goal of the doctor appointment system is to automate the current manual method with the use of computerized tools and comprehensive computer software, meeting their needs, in order to retain their important data for a longer length of time with simple access and manipulation. It may result in a management system that is error-free, secure, dependable, and quick.

The major goal is to handle the information for doctors, patients, appointments, bookings, and schedules. It controls all the information regarding doctors, their prices, their schedules, and their services. Administrator will always have access. The project's goal is to create an application software that will less the amount of manual effort required to manage medical appointments and expenses. It keeps track of every aspect of the patient, booking, and appointment schedule.

Features:

- Provides the searching facilities based on various factors. Such as doctor, patient, Booking, Doctor schedule.
- It also manages the Doctor fees details online for Booking details, doctor schedule details.
- It tracks all the information of Appointment, doctor fees, Booking etc.
- It deals with monitoring the information and transactions of booking.

- Editing, adding and updating of records is improved which results in proper resource management of doctor data.
- Manage the information of booking.
- Integration of all records of doctor schedule.

2. Modules of Doctor Appointment System:

- The doctor management module is utilized to handle the doctor's information.
- Doctor Schedule Module: This module is used to manage the specifics of the doctor
- The Doctor Fees Module is used to manage the specifics of physician fees.
- Patient Module: For handling patient information.
- Login Module: Used to control login information.
- Users Module: This module is used to manage system users.

3. Project Evaluation metrics:

3.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

3.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

3.3. API Details or User Interface:

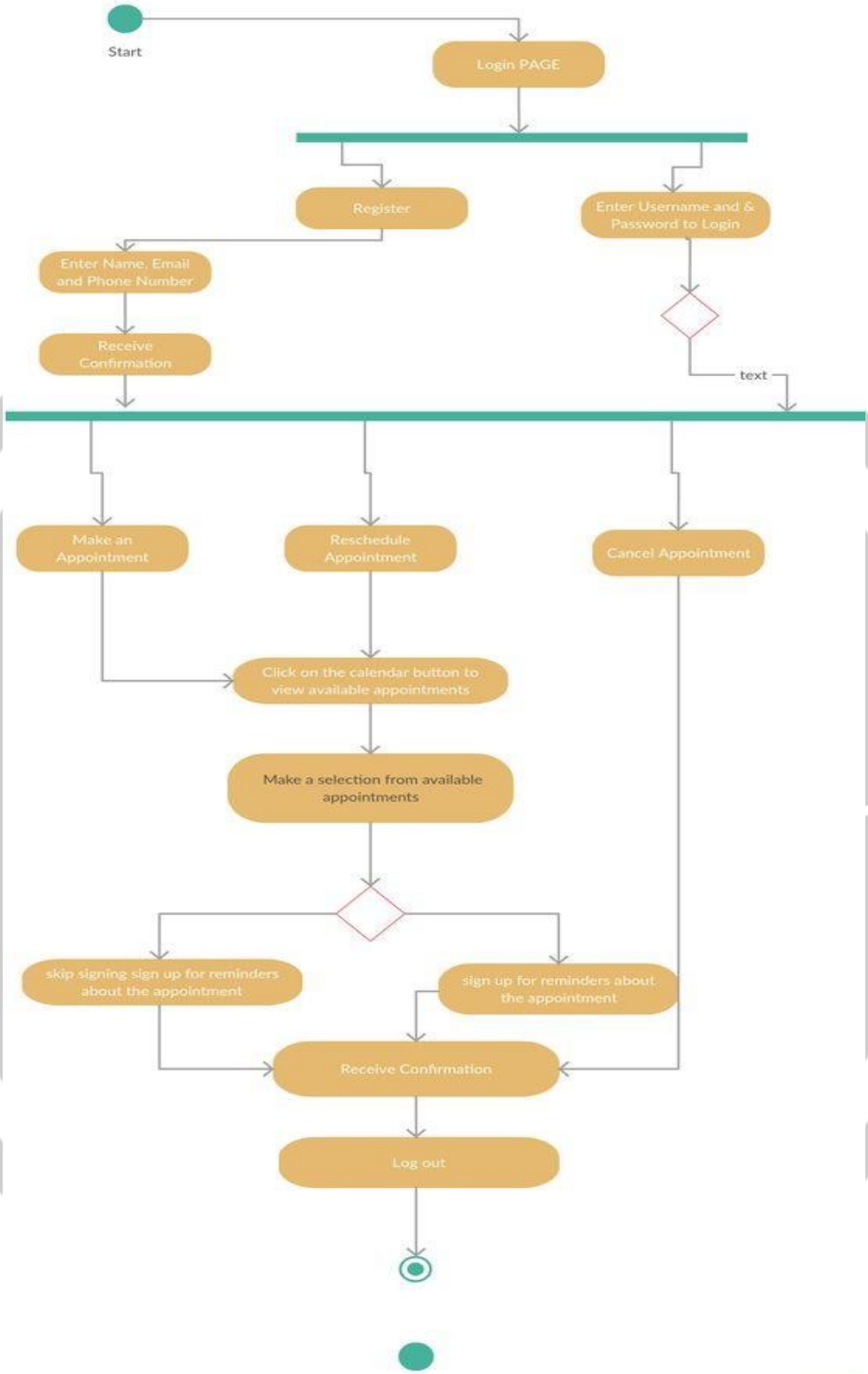
You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

3.4. Deployment:

Deploy the application on your preferred service.

3.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents**.**



3.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

3.7. Optimization of solutions:

Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

4. Submission requirements:

4.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link

4.2. level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: Link

4.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link

4.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link

4.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

4.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.
Demo link: DPR sample link

4.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link.