DocFlow

Appointment Health Care Management Platform

The platform aims to simplify doctor appointment booking and enhance the patient experience. The system provides access to a network of trusted doctors, helping to overcome challenges such as difficulty scheduling appointments, and a lack of information about care providers. The project also offers features like personalized recommendations and appointment reminders, contributing to increased efficiency and convenience for patients in their health journey.

Introduction

Purpose

The purpose of this platform is to enhance the patient experience by:

- Simplifying doctor appointment bookings.
- Managing patient-doctor interaction efficiently through secure profiles.

Scope

The system supports:

1. Appointment management (booking, rescheduling, and cancellation).
2. Doctor, patient, and admin authentication with role-based access.
3. Secure user accounts for patients and doctors to manage appointments and health data.
4. Real-time availability updates for doctors.
5. Automated reminders and notifications for patients.
6. Payment integration to facilitate online transactions.

General Description

Functions

1. **Appointment Booking:** Patients can book, cancel, or reschedule appointments with doctors.
2. **Health Record Management:** Patients can view appointment history and doctor notes.
3. **User Dashboards:** Separate dashboards for patients, doctors, and admins.
4. **Personalized Recommendations:** Patients receive doctor or service recommendations based on history.
5. **Reminders & Notifications:** Automated emails or SMS reminders for upcoming appointments.
6. **Online Payment Integration:** Patients can pay for consultations online.
7. **Admin Control Panel:** Admins control users, doctors, and appointments.

User Community

1. **Patients :**End users booking and managing their appointments.
2. **Doctors:** Healthcare providers offering services through the platform.
3. **Administrators:** Users managing platform operations and overseeing doctors and appointments.

Functional Requirements

1. **User Registration**
    - Patients and doctors must register with an email and password..
    - Users should have the option to securely log out from the platform
2. **User Profiles**
    - Patients should have secure profiles to deal with their personal information, including medical history, personal details such as name, and contact information.
    - Doctors should have secure profiles to add their professional information, including qualifications, experience, and availability, allowing them to easily update their information and set available appointment times.
3. **Doctor Filtering**
    - Patients should be able to show doctors based on:
    - **Specialty**: Allowing patients to find doctors who specialize in their specific health concerns.
    - **Availability**: Displaying real-time availability so patients can view open time slots for each doctor.
4. **Related Recommendations**
    - Recommendations are based on:

        - visits or Filtering.

5. **Appointment Scheduling**
    - **Booking, Rescheduling, and Cancellation:** Patients must be able to book, reschedule, and cancel appointments through the platform with ease.
    - **Real-Time Availability:** The system should display real-time availability of doctors, allowing patients to see open time slots for each doctor.
    - **Doctor Dashboard:** Doctors can view and manage (confirm or cancel) appointments through their dashboard, including the ability to confirm or decline patient requests.
    - **Confirmation Alerts:** Automated alerts should be displayed on the platform upon booking, rescheduling, or cancellation of appointments.
    - **Appointment History:** Patients can view their past and upcoming appointments, including details like date, time, doctor, and status (confirmed, cancelled).
6. **Admin Panel**
    - Administrators should have the ability to control user accounts, doctor profiles, and appointment schedules.
7. **payment**

    - **Payment Options:** The platform should support both online payment options (credit/debit cards, digital wallets) and cash payments.
    - **payment Confirmation:** Patients should receive confirmation alerts for successful online transactions or when cash payments are recorded.

User Interface Requirements

Software Interfaces

1. **Patient User Interface:**

   - Search and book appointments.
   - View appointment history and upcoming bookings.
   - Access personalized recommendations.

2. **Doctor Dashboard:**

   - Deal with doctor profile and availability.
   - View and manage(confirm or decline) appointments.

3. **Admin Panel:**

   - Deal with users, doctors, and appointments.
   - Monitor system performance.
   - Handle maintenance tasks.

Non-Functional Requirements

1. **Security**
   - **Encryption:** All sensitive data must be protected using secure techniques, including hashing methods for data integrity.
   - **Authentication:** The system should implement JWT for user authentication.
   - **Role-Based Access Control**: User roles should be enforced so that administrators have full data access while standard users can only access their personal information.
2. **Scalability**

   - The system should be able to handle up to 10,000 concurrent users without degradation in performance by using **MongoDB Atlas or** host the system on a reliable **cloud-based infrastructure, such as AWS**.

3. **Usability**
   - **User Interface**: The user interface should be easy to navigate for users of all ages and technical backgrounds.
   - **User documentation and support** should be available to assist patients and doctors in using the platform.
4. **Reliability**
   - **System Uptime**: The platform must maintain an uptime of 99.9%, ensuring users have access whenever needed

- **Scheduled Maintenance**: Maintenance activities should be planned and communicated in advance to users, taking place during off-peak hours (2 AM to 4 AM) to reduce disruptions and ensure minimal impact on user experience.

5. **Compatibility**
   - **Device Compatibility**: The platform should be responsive, supporting screen widths from a minimum of 320 pixels (small phones) to 1920 pixels (large screens).
   - **Browser Support**: The platform must be compatible with major browsers, including Chrome, Firefox, Safari, and Edge, for versions released within the last two years.
6. **Performance**
   - **Response Time:** Each operation in the system must complete within 3 ms.
   - Enhance it throught **MongoDB Atlas (**create indexes**) and Cloudinary(**CDN caching to improve media delivery**)**
7. **Safety**
   - **Secure Password Policy**: Passwords must be strong, consisting of a minimum of 8 characters, including uppercase and lowercase letters, numbers, and special characters.
8. **Availability**
   - **Continuous Access**: The system should be hosted on a reliable cloud-based infrastructure, such as AWS or Azure, to ensure uninterrupted access for users at any time across all regions.
9. **Reusability**
   - **Modular Components**: The platform should be developed using modular, independent components (such as React components for the frontend and API modules), making them easily reusable in other projects or applications.

Schedule and Budget

Timeline

- **Phase 1: Planning (2 weeks)**
  This phase includes defining the project scope, gathering requirements, and creating a detailed project plan.
- **Phase 2: Analysis and Design (3 weeks)**
  In this phase, the system architecture and design specifications are created based on the requirements gathered during the planning phase.
- **Phase 3: Development (1.5 month)**
  This is where the actual coding, database creation, and backend/frontend development will happen.
- **Phase 4: Testing and Launch (1 month)**
  In this phase, the system undergoes rigorous testing for bugs and usability, followed by deployment to production.

Cost Estimate

The initial cost estimate is **$500**.