Acknowledgement

We like to thank many people who helped us in completing this project. Foremost I would like to thank Almighty Allah for giving us strength and ability to complete this project. We would like to express our gratitude to all those who gave us the possibility to complete this project and we want to thank the Department of faculty for helping us to do the necessary research. We furthermore, thank our class Sir. Sagar Dalbanjan for giving us complete support and advising us in the project. Especially, we would like to thank to our colleagues for the valuable suggestions.

Introduction

At the heart of every thriving healthcare institution lies an efficient and streamlined management system. We are proud to introduce our cutting-edge Hospital Management System, designed to revolutionize the way healthcare facilities operate.

Our system is meticulously crafted to cater to the diverse needs of hospitals, clinics, and medical centers, offering a comprehensive suite of tools to enhance efficiency, improve patient care, and optimize resource allocation.

Existing Scenario:

In today's healthcare landscape, hospital management faces a myriad of challenges and opportunities shaped by technological advancements, evolving patient expectations, regulatory requirements, and financial constraints. Understanding the existing scenario is crucial for identifying areas of improvement and implementing effective solutions to enhance patient care, optimize operations, and ensure organizational sustainability.

Proposed Solution

To propose a comprehensive solution for a hospital management project, it's essential to outline specific steps and features that address the core needs and challenges faced by healthcare institutions. Here's how each component of the proposed solution can help improve hospital management:

1. Integrated Hospital Management System (HMS):

- Centralizes all hospital operations, including patient management, staff scheduling, inventory control, billing, and analytics, into a single platform.

- Streamlines workflows, reduces administrative burden, and improves operational efficiency by eliminating siloed systems and manual processes.

2. Electronic Health Records (EHR) System:

- Digitizes patient records, making them easily accessible to authorized healthcare providers anytime, anywhere.

- Enhances patient safety and care continuity by ensuring accurate and up-to-date medical information is readily available.

3. Telemedicine Platform:

- Expands access to healthcare services by offering remote consultations, follow-ups, and monitoring, particularly beneficial for patients in remote or underserved areas.

- Reduces patient wait times, travel costs, and healthcare disparities while increasing convenience and flexibility.

4. Data Analytics and AI-driven Insights:

- Analyzes large volumes of healthcare data to identify trends, patterns, and opportunities for improvement.

- Predicts patient outcomes, identifies high-risk populations, and optimizes resource allocation to improve efficiency and patient outcomes.

Functional Requirements

Functional requirements for a hospital management software outline the specific capabilities and features necessary to meet the needs of healthcare providers, patients, and administrative staff. Here's how each functional requirement can help people within and outside the hospital setting:

1. **Patient Management:**

   - **Appointment Scheduling:** Enables patients to schedule appointments conveniently and efficiently, reducing wait times and improving access to care.

   - **Patient Registration:** Simplifies the registration process for patients, ensuring accurate capture of demographic information and medical history.

2. **Clinical Documentation:**

   - **Electronic Health Records (EHR):** Digitizes patient medical records, enabling healthcare providers to access comprehensive patient information quickly and securely.

   - **Clinical Notes:** Allows clinicians to document patient encounters, assessments, and treatment plans electronically, improving accuracy and completeness of clinical documentation.

3. **Billing and Financial Management:**

   - **Billing and Invoicing:** Automates billing processes, generates accurate invoices, and facilitates claims submission to insurance providers, reducing billing errors and streamlining revenue cycle management.

4. **Staff Management:**

   - **Staff Scheduling:** Simplifies staff scheduling, assigns shifts, and manages staff availability, ensuring adequate staffing levels and reducing scheduling conflicts.

   - **Time and Attendance:** Tracks staff attendance, hours worked, and overtime, automating payroll processing and ensuring compliance with labor regulations.

Login Panel

Enter correct username and password then login and perform specific tasks the application automatic check who is login Admin/Petient/ Doctor etc.

Creating a login panel for a Hospital Management system requires careful consideration of security, user roles, and data privacy. Here's a general outline of how youmight structure such a system:

1. User Roles:

In a hospital management system, various user roles have distinct responsibilities and permissions tailored to their specific functions within the healthcare facility. Here are some common user roles:

- **Administrator:** Manages system configuration, user accounts, and permissions
- Oversees overall system functionality and ensures compliance with regulatory requirements.
- Responsible for system maintenance, upgrades, and troubleshooting.
  .
- **Doctor / Physician:** Accesses patient records, medical history, and diagnostic results.

- Enters clinical notes, treatment plans, and prescriptions.

- Orders tests, procedures, and medications.

2. Login Panel for Patient:

- Username

- Email

- Password

- " Already have an account?

- "Male or Female (Choosing)
- Phone No

3. Authentication:

- Use secure protocols like HTTPS.

- Implement password hashing and salting to store passwords securely.

- Utilize session management techniques to maintain user sessions securely.

4. Patient Information Management:

Patient Information Management is a crucial aspect of hospital management, encompassing collection, storage, retrieval, and utilization of patient-related data throughout their healthcare journey. Here's how patient information management contributes to effective hospital management

5. Hospital Management:

Hospital management software plays a crucial role in improving efficiency, enhancing patient care, and streamlining administrative processes within healthcare facilities. Here's how it helps people at various levels

- Patients: Improved Access to Care: Patients can easily schedule appointments, access medical records, and communicate with healthcare providers through patient portals or mobile apps, leading to better access to healthcare services.

- Enhanced Patient Experience: Hospital management software reduces wait times, streamlines registration processes, and provides personalized care, resulting in improved patient satisfaction and overall experience.

1. Patient Requests Appointment:

- The patient logs into the hospital management system.

- The patient navigates to the appointment scheduling section.

- The patient selects the preferred date and time for the appointment and chooses the type of consultation (e.g., general check-up, specific medical concern).

- The patient submits the appointment request.

2. Administrator Receives Appointment Request:

- The administrator receives a notification or sees the pending appointment request in the system.

- The administrator reviews the request and checks the doctor's availability for the requested time slot.

- If the requested time slot is available, the administrator proceeds to schedule the appointment; otherwise, they may suggest alternative available times to the patient.

3. Administrator Schedules Appointment:

- The administrator selects the doctor for the appointment based on the patient's preference or availability.

- The administrator confirms the appointment details, including the date, time, doctor's name, and type of consultation.

- The administrator saves the appointment details in the system.

4. Confirmation Sent to Patient:

- The system sends a confirmation message to the patient, including the appointment details and any instructions or preparations required before the appointment.

- The patient receives the confirmation and acknowledges it.

Alternative Flow

- **Doctor Availability Conflict**:

  - If the doctor is not available for the requested time slot, the administrator suggests alternative available times to the patient.

  - The patient either selects one of the suggested times or chooses a different date and time for the appointment.

- **Postconditions:**

- The appointment is scheduled and recorded in the hospital management system.

- The doctor and patient are notified of the scheduled appointment.

- The patient is expected to attend the appointment at the scheduled date and time.

- **Patient Cancels Appointment**:

- If the patient needs to cancel the appointment, they can do so through the hospital management system.

- The system cancels the appointment, updates the appointment status, and notifies the doctor and administrator of the cancellation.

Technologies Used:

- **HTML and CSS:** Used to design the user interface for patient registration, appointment scheduling, and appointment management. CSS is utilized for styling and layout.

- **PHP:** Handles server-side scripting for processing patient registration, appointment booking, rescheduling, and cancellation requests. Interacts with the MySQL database to fetch and update appointment data.

- **JavaScript:** Enhances user interaction and validation in the appointment scheduling process. Provides dynamic features like calendar selection and real-time availability checks.

- **MySQL:** Stores patient information, appointment details, and scheduling data. Utilizes relational database structure to ensure data integrity and efficient retrieval.

Types of Specialist Doctors

1. Neurologist:

Neurologists specialize in diagnosing and treating disorders of the nervous system, including the brain, spinal cord, nerves, and muscles. They treat conditions such as epilepsy, stroke, multiple sclerosis, Parkinson's disease, and Alzheimer's disease.

2. General Surgeon:

- General surgeons perform surgical procedures on various parts of the body, including the abdomen, chest, and extremities. They treat a wide range of conditions, such as appendicitis, hernias, gallbladder disease, and trauma injuries.

3. Cardiologist:

- Cardiologists specialize in diagnosing and treating diseases and conditions of the heart and circulatory system. They treat conditions such as coronary artery disease, heart failure, arrhythmias, and valvular heart disease.

4. Pediatrician:

1. Pediatricians specialize in providing medical care to infants, children, and adolescents. They focus on promoting the health and well-being of young patients and treating conditions specific to this age group, such as childhood infections, growth and development issues, and behavioral disorders.

1. Neurologist:

2. **Diagnosis:** Neurologists use a combination of clinical examination, medical history review, and diagnostic tests (such as MRI, CT scans, and EEG) to diagnose neurological conditions accurately. They may also collaborate with other specialists, such as neurosurgeons or neuroradiologists, for comprehensive evaluation and diagnosis.

3. **Treatment:** Once a diagnosis is made, neurologists develop personalized treatment plans tailored to each patient's specific condition and needs. Treatment options may include medication management, physical therapy, occupational therapy, speech therapy, and lifestyle modifications. In some cases, neurologists may also perform surgical procedures or recommend interventional therapies.

4. **Management of Chronic Conditions:** Neurological disorders often require long-term management and monitoring. Neurologists work closely with patients to manage symptoms, optimize treatment efficacy, and improve quality of life. They may also provide education and support to help patients and their families cope with the challenges associated with chronic neurological conditions.

5. **Research and Innovation:** Neurologists are actively involved in clinical research and medical advancements aimed at improving the diagnosis, treatment, and understanding of neurological disorders. They contribute to scientific studies, clinical trials, and collaborative research initiatives to develop new therapies and enhance patient care.

6. **Collaboration with Multidisciplinary Teams:** Neurologists often collaborate with other healthcare professionals, including neurosurgeons, neuropsychologists, neurointerventionalists, physical therapists, and social workers, to provide comprehensive care for patients with complex neurological conditions. This multidisciplinary approach ensures that patients receive holistic and integrated care across various medical specialties.

2. General Surgeon:

General surgeons play a crucial role in hospital management, especially in surgical departments. Here's how they contribute:

1. Surgical Procedures: General surgeons perform a wide range of surgical procedures, from routine appendectomies to complex gastrointestinal surgeries. Their expertise ensures that patients receive appropriate surgical care.

2. Patient Care: General surgeons oversee the preoperative, intraoperative, and postoperative care of patients. They collaborate with other healthcare professionals to ensure optimal patient outcomes.

3. Clinical Leadership: General surgeons often hold leadership positions within surgical departments, providing guidance and direction to other surgeons, nurses, and support staff.

4. Quality Improvement: They participate in quality improvement initiatives to enhance patient safety, streamline processes, and improve outcomes. They analyze surgical outcomes and implement best practices to ensure high-quality care.

5. Resource Management: General surgeons are involved in managing resources within the surgical department, including equipment, supplies, and personnel. They ensure that resources are utilized efficiently to meet patient need

3. Cardiologist:

Cardiologists play a significant role in hospital management, particularly in departments related to cardiovascular care. Here's how they contribute:

1. Clinical Expertise: Cardiologists provide specialized care for patients with cardiovascular diseases, including coronary artery disease, heart failure, arrhythmias, and more. Their expertise ensures that patients receive accurate diagnosis and appropriate treatment.

2. Patient Care Coordination: Cardiologists oversee the management of patients with cardiovascular conditions, coordinating care across various healthcare professionals, including nurses, cardiac surgeons, interventional cardiologists, and other specialists.

3. Quality Improvement: Cardiologists participate in quality improvement initiatives focused on enhancing the quality and safety of cardiovascular care. They analyze outcomes, identify areas for improvement, and implement evidence-based practices to optimize patient outcomes.

4. Clinical Leadership: Many cardiologists hold leadership positions within cardiovascular departments or hospital administration. They provide guidance and direction to other healthcare professionals, ensuring that departmental goals are aligned with the overall mission of the hospital.

5. Resource Management: Cardiologists are involved in managing resources within cardiovascular departments, including diagnostic equipment, catheterization labs, and cardiac rehabilitation programs.

4. **Pediatrician**

Pediatricians play a crucial role in hospital management, especially in departments focused on pediatric care. Here's how they contribute:

1. **Clinical Expertise**: Pediatricians provide specialized medical care for infants, children, and adolescents. They diagnose and treat a wide range of pediatric conditions, from common illnesses to complex medical issues, ensuring that young patients receive appropriate care tailored to their unique needs.

2. **Patient Care Coordination**: Pediatricians oversee the management of pediatric patients within the hospital, coordinating care among various healthcare professionals, including nurses, pediatric subspecialists, therapists, and other allied health professionals. They ensure that care plans are comprehensive and effectively implemented.

3. **Quality Improvement**: Pediatricians participate in quality improvement initiatives aimed at enhancing the quality and safety of pediatric care. They analyze outcomes, identify areas for improvement, and implement evidence-based practices to optimize patient outcomes and satisfaction.

4. **Clinical Leadership**: Many pediatricians hold leadership positions within pediatric departments or hospital administration. They provide leadership and direction to other healthcare professionals, ensuring that departmental goals align with the hospital's mission and vision for pediatric care.

5. **Resource Management:** Pediatricians are involved in managing resources within pediatric departments, including pediatric-specific equipment, medications, and specialized facilities such as pediatric intensive care units (PICUs) and neonatal intensive care units (NICUs).

Feedback Of Patient

1. Feedback Channels: Establish multiple channels for collecting feedback to ensure accessibility for patients. This can include surveys, suggestion boxes, online forms, and direct communication with staff.

2. Timely Collection: Gather feedback throughout the patient's journey, from admission to discharge. Promptly address any issues or concerns as they arise to demonstrate responsiveness and commitment to patient satisfaction.

3. Anonymous Option: Offer patients the choice to provide feedback anonymously to encourage honest and candid responses. This can help mitigate any fear of reprisal and facilitate more open communication.

4. Quality Metrics: Develop standardized metrics to measure patient satisfaction and experience. These can include aspects such as communication with healthcare providers, cleanliness of facilities, wait times, and overall treatment outcomes.

5. Data Analysis: Utilize data analytics tools to analyze feedback data and identify trends/patterns. This can provide valuable insights into common issues or areas requiring improvement, allowing for targeted interventions.

6. Staff Training: Provide training to healthcare staff on the importance of patient feedback and how to effectively address patient concerns. Empower staff to act on feedback and implement necessary changes to enhance the patient experience.

7. Continuous Improvement: Implement a culture of continuous improvement where feedback is used to drive ongoing enhancements in service delivery and patient care practices.

8. Transparency and Communication: Keep patients informed about the actions taken in response to their feedback. Transparency fosters trust and demonstrates a commitment to accountability and improvement.

9. Celebrate Successes: Acknowledge and celebrate achievements and improvements resulting from patient feedback. Recognizing positive changes reinforces the value of patient input and encourages continued engagement.

Frontend Architectur

Patient feedback plays a crucial role in the effective management of hospitals. It serves as a valuable tool for understanding patient experiences, identifying areas for improvement, and enhancing overall quality of care. Here are some key aspects to consider when incorporating patient feedback into a hospital management system

1. Client-Side Technologies: PHP is primarily a server-side scripting language, so the frontend architecture often relies on client-side technologies such as HTML, CSS, and JavaScript. HTML provides the structure of web pages, CSS styles the visual presentation, and JavaScript adds interactivity and dynamic behavior.

2. Separation of Concerns: Following the principles of separation of concerns, the frontend architecture separates the presentation layer (HTML/CSS) from the application logic (PHP). This separation allows for easier maintenance, scalability, and collaboration among developers.

3. Template Engines: PHP-based projects often use template engines like Twig, Blade, or Smarty to manage the presentation layer efficiently. These template engines allow developers to create reusable templates with placeholders for dynamic content generated by PHP.

4. MVC Architecture: While PHP itself doesn't enforce a specific frontend architecture, many PHP frameworks, such as Laravel, Symfony, and CodeIgniter, follow the Model-View-Controller (MVC) architecture. In MVC, the Model represents the data and business logic, the View handles the presentation layer, and the Controller manages the application's flow and user interactions.

5. RESTful APIs: In modern web development, frontend architectures often interact with backend APIs to fetch and manipulate data asynchronously. PHP projects can implement RESTful APIs using frameworks like Laravel or Slim, enabling frontend applications to communicate with the backend server via HTTP requests.

6. Frontend Libraries and Frameworks: Frontend architectures may incorporate popular JavaScript libraries and frameworks like React, Vue.js, or Angular to build dynamic and interactive user interfaces. These libraries/frameworks facilitate the development of single-page applications (SPAs) or progressive web apps (PWAs) that offer a seamless user experience.

7. Asset Management: Managing frontend assets such as stylesheets, scripts, and images is essential for performance optimization and maintainability. PHP projects typically use tools like Webpack, Gulp, or Laravel Mix to bundle, minify, and optimize assets for production.

8. Responsive Design: With the proliferation of mobile devices, responsive design has become a standard practice in frontend architecture. CSS frameworks like Bootstrap or Tailwind CSS help in building responsive and mobile-friendly interfaces that adapt to different screen sizes and resolutions.

9. Security Considerations: Frontend architectures should address security concerns such as cross-site scripting (XSS), cross-site request forgery (CSRF), and input validation. PHP frameworks often provide built-in security features and best practices to mitigate common vulnerabilities.

10. Testing and Debugging: Effective frontend architectures include provisions for testing and debugging UI components to ensure functionality and usability. Tools like PHPUnit for PHP testing and Jest for JavaScript testing help developers validate frontend code and identify potential issues.

Backend Architecture

In a PHP project using XAMPP, the backend architecture typically involves setting up a local development environment for building and testing web applications. XAMPP is a popular solution stack that includes Apache, MySQL, PHP, and Perl, making it easy to run PHP-based applications on a local machine. Here's an explanation of the backend architecture in a PHP project using XAMPP

1. XAMPP Stack: XAMPP provides a bundled package of software components required to run a web server environment locally. It includes:
   - Apache: The web server responsible for serving web pages to clients.
   - MySQL: A relational database management system (RDBMS) used for storing and managing application data.
   - PHP: A server-side scripting language used for building dynamic web pages and web applications.
   - Perl: A general-purpose scripting language (though less commonly used in PHP projects).

2. Apache Configuration: Apache serves as the HTTP server for the PHP project. Developers configure Apache to handle PHP requests by setting up virtual hosts, configuring directory permissions, enabling modules like mod_rewrite for URL rewriting, and managing server-side scripting.

3. MySQL Database: MySQL serves as the backend database management system for storing application data. Developers use tools like phpMyAdmin (included in XAMPP) or MySQL Workbench to create databases, tables, and manage data.

4. PHP Backend Logic: PHP scripts handle the backend logic of the application, including processing form submissions, interacting with the database, handling user authentication, and generating dynamic content. Developers write PHP code to implement business logic, connect to the MySQL database using MySQLi or PDO extensions, and execute queries to retrieve or manipulate data.

5. MVC Architecture: While XAMPP itself doesn't enforce a specific architecture, PHP frameworks like Laravel, Symfony, or CodeIgniter often follow the Model-View-Controller (MVC) architecture. In MVC, the Model represents the data and business logic, the View handles the presentation layer, and the Controller manages the application's flow and user interactions. Developers structure their PHP code according to MVC principles for better organization and maintainability.

6. Development Workflow: In a typical development workflow using XAMPP, developers write PHP code, HTML, CSS, and JavaScript files in a local development environment. They use text editors or integrated development environments (IDEs) to write code and test it locally using the XAMPP server stack. Changes are made iteratively, and developers can debug PHP scripts using tools like Xdebug or built-in PHP error handling.

7. Deployment: Once the development and testing phases are complete, developers deploy the PHP application to a production server for public access. Deployment involves transferring files to a remote server, configuring the server environment, setting up database connections, and ensuring security measures are in place to protect the application from vulnerabilities:

Testing Proceduers

Testing a PHP project involves verifying that the application functions correctly, performs as expected, and meets the specified requirements. Here's an explanation of the testing procedure for a PHP project:

1. Unit Testing:
- **Explanation:** Unit testing involves testing individual units or components of the PHP code in isolation. This ensures that each unit behaves as expected and fulfills its specific functionality.
- **Procedure:** Developers write unit tests using testing frameworks like PHPUnit. They create test cases to cover various scenarios, including edge cases and boundary conditions. Unit tests focus on functions, methods, and classes to validate their behavior and outputs.

2. Integration Testing:
- **Explanation:** Integration testing verifies that different components of the PHP application work together seamlessly when integrated. It ensures that the interactions between modules, classes, and subsystems are functioning correctly.
- **Procedure:** Developers create integration test cases to test the interaction between multiple units or modules within the application. This includes testing database interactions, API integrations, and external dependencies. Integration tests help identify any compatibility issues or communication failures between components.

3. Functional Testing:
- **Explanation:** Functional testing evaluates the overall functionality of the PHP application from the end user's perspective. It verifies that the application behaves according to the specified requirements and user expectations.
- **Procedure:** Testers create functional test cases based on user stories, use cases, or requirements documentation. They simulate user interactions with the application, such as submitting forms, navigating pages, and performing actions. Functional tests ensure that the application's features work as intended and produce the expected results.

4. User Interface (UI) Testing:
- **Explanation:** UI testing focuses on verifying the graphical user interface (GUI) elements of the PHP application, including layout, design, and usability. It ensures that the UI elements render correctly across different devices and browsers.
- **Procedure:** Testers use automated testing tools like Selenium or Cypress to automate UI tests. They create test scripts to interact with the application's UI elements, validate UI behavior, and check for visual consistency. UI testing helps identify issues related to responsiveness, accessibility, and user experience.

5. Performance Testing:
- **Explanation:** Performance testing assesses the responsiveness, scalability, and stability of the PHP application under varying workloads and conditions. It measures factors such as response time, throughput, and resource utilization.
- **Procedure:** Testers use performance testing tools like Apache JMeter or Gatling to simulate concurrent user traffic and stress test the application. They analyze performance metrics to identify bottlenecks, optimize code, and improve the overall performance of the application.

6. Security Testing:

- **Explanation:** Security testing identifies vulnerabilities and weaknesses in the PHP application that could be exploited by malicious actors. It ensures that the application is secure against common security threats and attacks.

- **Procedure:** Testers perform security assessments using tools like OWASP ZAP or Burp Suite to scan for vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication flaws. They conduct penetration testing to identify potential security loopholes and recommend security best practices to mitigate risks

ER Diagram

Fig -1 ER diagram

Project Screenshots

User module (patient):

1. Dashboard: In this section, patients can view the his/her profile, Appointments and Book Appointment.

2. Book Appointment: In this section, Patient can book his/her appointment.

3. Appointment History: In this section, Patients can see his/her own appointment history.

4. Medical History: In this section, Patients can see his/her own appointment history.

5. User can update his/her profile, change the password and recover the password.

Fig-2 Patient Registration Page

Fig-3 Patient Login Page

Fig-4 Patient Home Page

Fig-5 Petient Appointment Booking

Fig-6 Petient Appointment History

Fig-7 Petient Prescription

Doctor module:

1. Dashboard: In this section, doctor can view his/her own profile and online appointments.

2. Appointment History: In this section, Doctor can see patient's appointment history.

3. Patients: In this section, doctor can manage patients (Add/Update).

4. Search: In this section, doctor can search patient with the help of patient name and mobile number.

5. Doctor can also update his profile, change the password and recover the password.

Fig-8 Docter login Page

Fig-9 Doctor Home Page

Fig-10 Appointment List

Fig-11 Prescription List

Admin module:

1. Dashboard: In this section, admin can view the Patients, Doctors, Appointments and New queries.Dashboard: In this section, admin can view the

2. Patients, Doctors, Appointments and New queries.

3. Users: In this section, admin can view users detail(who take online appointment) and also have right to delete irrelevant user.

4. Patients: In this section, admin can view patient's details.

5. Appointment History: In this section, admin can view appointment history.

6. Contact us Queries: In this section, admin can view queries which are send by users.

7. Doctor Session Logs: In this section, admin can see login and logout time of doctor.

8. User Session Logs: In this section, admin can see login and logout time of user.

9. Reports: In this section, admin can view reports of patients in particular periods.

10. Patient Search: In this section, admin can search patient with the help of patient name and mobile number.

11. Admin can also change his/her own password

Feedback Of Patient:

10. Feedback Channels: Establish multiple channels for collecting feedback to ensure accessibility for patients. This can include surveys, suggestion boxes, online forms, and direct communication with staff.

11. Timely Collection: Gather feedback throughout the patient's journey, from admission to discharge. Promptly address any issues or concerns as they arise to demonstrate responsiveness and commitment to patient satisfaction.

12. Anonymous Option: Offer patients the choice to provide feedback anonymously to encourage honest and candid responses. This can help mitigate any fear of reprisal and facilitate more open communication.

13. Quality Metrics: Develop standardized metrics to measure patient satisfaction and experience. These can include aspects such as communication with healthcare providers, cleanliness of facilities, wait times, and overall treatment outcomes.

14. Data Analysis: Utilize data analytics tools to analyze feedback data and identify trends/patterns. This can provide valuable insights into common issues or areas requiring improvement, allowing for targeted interventions.

15. Staff Training: Provide training to healthcare staff on the importance of patient feedback and how to effectively address patient concerns. Empower staff to act on feedback and implement necessary changes to enhance the patient experience.

16. Continuous Improvement: Implement a culture of continuous improvement where feedback is used to drive ongoing enhancements in service delivery and patient care practices.

17. Transparency and Communication: Keep patients informed about the actions taken in response to their feedback. Transparency fosters trust and demonstrates a commitment to accountability and improvement.

18. Celebrate Successes: Acknowledge and celebrate achievements and improvements resulting from patient feedback. Recognizing positive changes reinforces the value of patient input and encourages continued engagement.
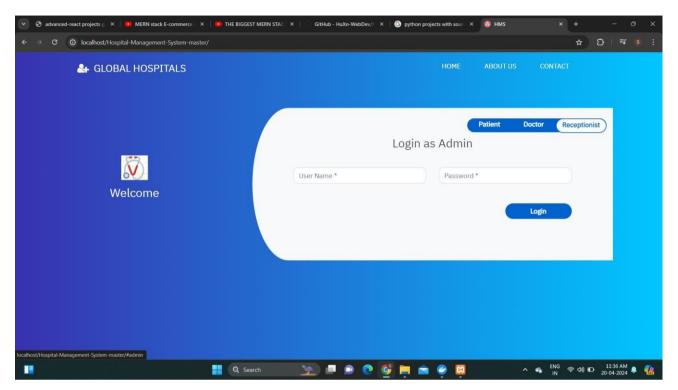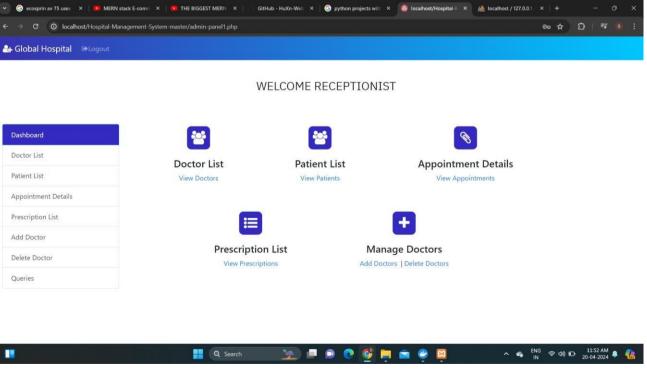
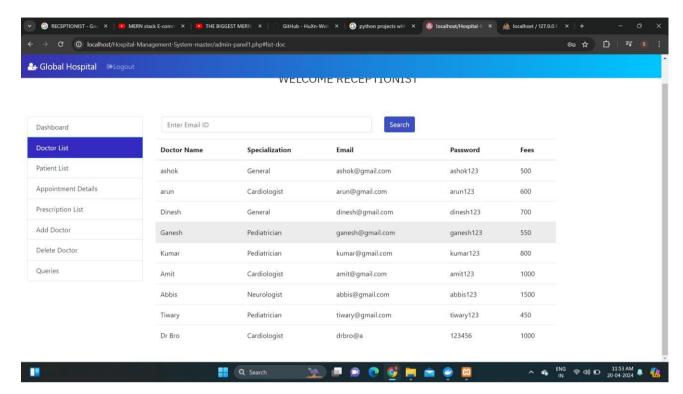Fig-12 Admin or Reseptionist login

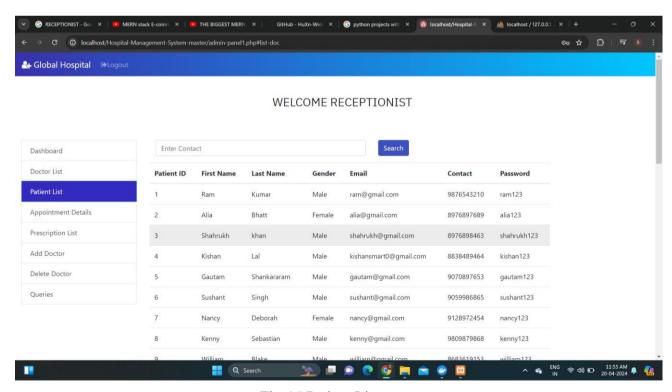Fig-13 Admin DashBoard

Fig-14 Doctors List
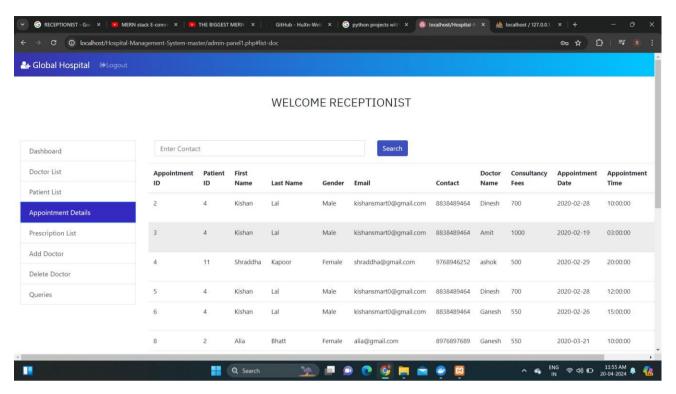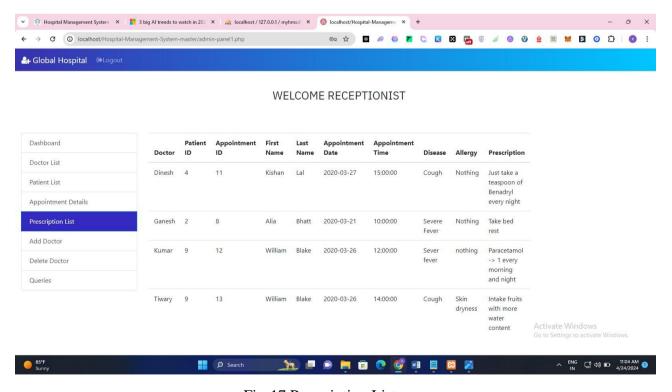

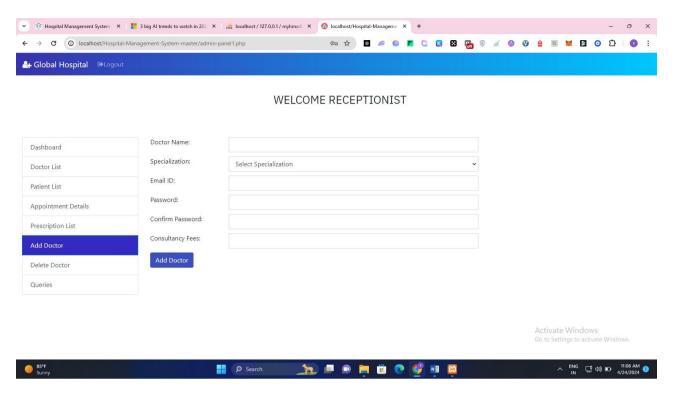Fig-15 Patient List
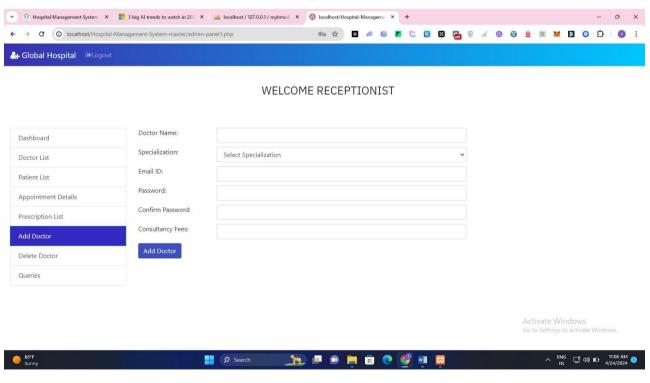
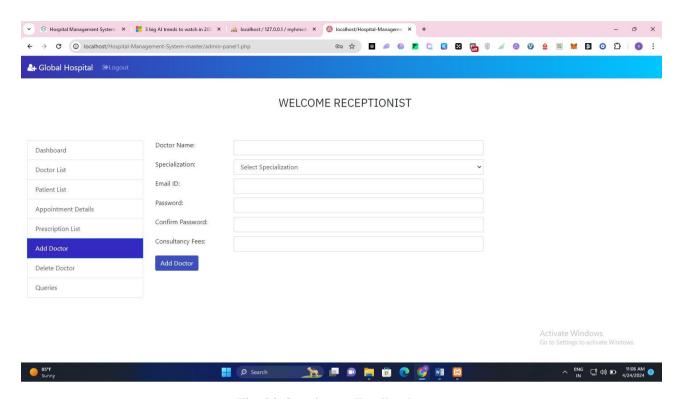Fig-16 Appointment Details

Fig-17 Prescription List

Fig-18 Add Doctors

Fig-19 Delete Doctors

Fig-20 Queries or Feedback