CHAPTER – 1 (OVERVIEW)

1.1 INTRODUCTION

The Hospital Management System (HMS) is a robust web application designed to streamline the management of hospital activities efficiently. Developed using Java as the core technology and MySQL as the backend database, this system enables seamless handling of patient records, doctor management, and appointment scheduling. With a user-friendly interface implemented through JSP, HTML, CSS, and JavaScript, the HMS ensures ease of access and navigation for all users, including administrators, doctors, and receptionists. By adhering to the MVC architecture and utilizing Servlets for server-side processing, the system maintains a structured and organized approach to data management. Whether it's adding, updating, or viewing information, the HMS empowers administrators to oversee the entire hospital workflow while enabling doctors and receptionists to fulfill their roles effectively. Through its comprehensive features and technology stack, the HMS serves as a reliable tool for enhancing hospital operations and delivering quality patient care in an online environment.

Advantages:

1. **Efficient Data Management:** The HMS centralizes patient records, doctor information, and appointment schedules in one digital platform, making it easier to access and manage data efficiently.
2. **Improved Patient Care:** With quick access to patient information and appointment schedules, doctors can provide timely and personalized care, leading to improved patient outcomes and satisfaction.
3. **Streamlined Workflow:** The system automates various administrative tasks such as appointment scheduling, patient registration, and staff management, leading to a more streamlined workflow and increased productivity.
4. **Enhanced Communication:** HMS facilitates better communication between different departments within the hospital, enabling seamless coordination and collaboration among doctors, receptionists, and administrators.
5. **Data Security:** With robust security measures in place, including user authentication and access controls, the HMS ensures the confidentiality and integrity of patient data, adhering to regulatory compliance standards.
6. **Cost-Efficiency:** By reducing manual paperwork and streamlining administrative processes, the HMS helps in cost reduction and resource optimization, making healthcare operations more financially sustainable.
7. **Accessibility:** Being a web-based application, the HMS can be accessed from anywhere with an internet connection, allowing doctors and staff to manage hospital activities remotely, if necessary.
8. **Customization:** The HMS can be customized to meet the specific needs and requirements of different healthcare facilities, ensuring flexibility and adaptability to changing demands.

1.2 BACKGROUND

The Hospital Management System (HMS) serves as a crucial tool in modern healthcare facilities, addressing the need for efficient management of hospital activities and patient care. With the increasing complexity of healthcare services and the growing volume of patient data, there arises a necessity for streamlined processes and digital solutions to enhance operational efficiency and improve patient outcomes.

Traditionally, hospital management relied heavily on manual paperwork, making it prone to errors, delays, and inefficiencies. Tasks such as patient registration, appointment scheduling, doctor management, and record-keeping were labor-intensive and time-consuming, leading to bottlenecks in workflow and hindering the delivery of quality healthcare services.

In response to these challenges, the development of the Hospital Management System emerged as a solution to modernize and optimize hospital operations. By leveraging advancements in technology, particularly in the fields of Java programming, web development, and database management, the HMS revolutionizes how healthcare facilities manage their daily activities and interact with patients.

Through the integration of Java-based technologies such as JSP, Servlets, and MySQL database, the HMS provides a robust and scalable platform for storing, processing, and accessing patient records, doctor information, and appointment schedules. The adoption of a web-based architecture enables easy accessibility from any location with an internet connection, facilitating remote management of hospital activities and enhancing communication and collaboration among healthcare professionals.

Furthermore, the implementation of the HMS adheres to industry standards and regulations regarding data security and patient confidentiality, ensuring compliance with healthcare privacy laws and regulations. Robust security measures, including user authentication and access controls, safeguard sensitive patient information from unauthorized access and data breaches, maintaining trust and integrity within the healthcare ecosystem.

Overall, the Hospital Management System represents a significant advancement in healthcare technology, offering a comprehensive solution to optimize hospital management, improve patient care, and enhance the overall efficiency and effectiveness of healthcare delivery. Its adoption signifies a shift towards a more digital and patient-centric approach to healthcare management, ushering in a new era of innovation and excellence in the healthcare industry.

1.3 OBJECTIVE

The primary objective of implementing the Hospital Management System (HMS) is to streamline and optimize hospital operations while improving the quality of patient care. By digitizing and automating various administrative tasks such as patient registration, appointment scheduling, and record-keeping, the HMS aims to enhance operational efficiency, reduce paperwork, and minimize errors. Additionally, the system seeks to facilitate better communication and collaboration among healthcare professionals, ensuring seamless coordination in delivering timely and personalized patient care. Overall, the objective of the HMS is to modernize healthcare management, improve workflow efficiency, and ultimately enhance the overall patient experience within the healthcare facility.

1.4 PURPOSE

The purpose of developing the Hospital Management System (HMS) is to provide a comprehensive and efficient digital platform for managing all aspects of hospital activities and patient care. This system aims to streamline administrative tasks such as patient registration, appointment scheduling, and record-keeping, thereby reducing manual paperwork and increasing operational efficiency. Additionally, the HMS facilitates better communication and coordination among healthcare professionals, ensuring timely and personalized patient care. By digitizing hospital management processes, the system aims to improve overall workflow efficiency, enhance the quality of patient care, and ultimately contribute to the smooth and effective functioning of the healthcare facility.

1.5 SCOPE

- Patient Management: The HMS facilitates efficient patient registration, record-keeping, and management, ensuring accurate and up-to-date patient information.
- Doctor Management: The system provides tools for managing doctor profiles, scheduling appointments, and tracking doctor-patient interactions.
- Appointment Management: The HMS enables the scheduling, rescheduling, and cancellation of patient appointments, ensuring optimal utilization of doctor's time and resources.
- Record Management: It includes the management of patient records, medical histories, treatment plans, and other relevant documentation in a secure and organized manner.
- Staff Management: The system allows for the management of hospital staff, including receptionists, nurses, and other support staff, by tracking their schedules, roles, and responsibilities.
- Reporting and Analytics: The HMS generates reports and analytics on various aspects of hospital operations, such as patient flow, appointment scheduling, and resource utilization, to aid in decision-making and performance evaluation.

LIMITATIONS

- Dependency on Technology: The HMS relies heavily on technology infrastructure, including servers, networks, and software applications. Any technical issues or system failures can disrupt hospital operations and affect patient care.
- Training Requirements: Users, including administrators, doctors, and receptionists, require training to effectively use the HMS. Training programs may be time-consuming and require ongoing updates as the system evolves.
- Data Security Concerns: Despite robust security measures, there is always a risk of data breaches or unauthorized access to patient information. Hospitals must continuously monitor and update security protocols to mitigate these risks.
- Integration Challenges: Integrating the HMS with existing hospital systems, such as electronic health records (EHR) or billing systems, can be complex and time-consuming. Compatibility issues may arise, requiring additional resources and expertise to resolve.
- Limited Customization: While the HMS may offer some degree of customization, it may not fully meet the unique needs of every healthcare facility. Customization options may be limited, leading to compromises in functionality or workflow efficiency.

1.6 APPLICABILITY

The Hospital Management System (HMS) is applicable to a wide range of healthcare facilities, including hospitals, clinics, medical centers, and specialty care centers. It is particularly suitable for environments where efficient management of hospital activities and patient care is essential. The applicability of the HMS extends to various stakeholders within the healthcare ecosystem, including:

1. Hospital Administrators: HMS assists administrators in managing hospital resources, personnel, and finances effectively. It provides tools for tracking patient data, appointment scheduling, and staff management, enabling administrators to optimize hospital operations and enhance overall efficiency.
2. Medical Practitioners: Doctors benefit from the HMS by accessing patient records, managing appointments, and communicating with patients and colleagues efficiently. The system streamlines clinical workflows, allowing

doctors to focus on providing quality patient care and improving treatment outcomes.

3. Receptionists and Administrative Staff: Receptionists and administrative staff utilize the HMS to manage patient appointments, maintain patient records, and handle administrative tasks. The system helps in coordinating patient flow, ensuring smooth operations at the front desk, and enhancing the overall patient experience.
4. Patients: Patients interact with the HMS through online appointment scheduling, access to medical records, and communication with healthcare providers. The system empowers patients to take control of their healthcare journey, improving accessibility to healthcare services and enhancing patient engagement and satisfaction.
5. Regulatory Authorities: The HMS aids regulatory authorities in monitoring and ensuring compliance with healthcare regulations and standards. It provides mechanisms for securely managing patient data, maintaining confidentiality, and adhering to privacy laws such as HIPAA.
6. Healthcare Providers: Healthcare providers, including specialists, nurses, and support staff, utilize the HMS to access patient information, coordinate care plans, and communicate with other members of the care team. The system facilitates collaboration among healthcare providers, leading to improved care coordination and patient outcomes.

Chapter - 2 (Feasibility Study and Stakeholders)

2.1 FEASIBILITY STUDY

1. **Technical Feasibility:** The Hospital Management System (HMS) is technically feasible as it leverages commonly used technologies such as Java, JSP, Servlets, MySQL, and

NetBeans. These technologies are well-established and widely supported, with ample resources and expertise available for development and implementation.

2. **Operational Feasibility:** The HMS aims to streamline hospital operations by digitizing administrative tasks, improving workflow efficiency, and enhancing communication among stakeholders. Operational feasibility is high as it aligns with the goals of modernizing healthcare management practices and improving patient care.
3. **Economic Feasibility:** While there are initial costs associated with software development, hardware infrastructure, and training, the long-term benefits of implementing the HMS outweigh the investment. These benefits include reduced administrative costs, improved resource utilization, and enhanced patient outcomes, making it economically feasible.
4. **Schedule Feasibility:** The development and implementation timeline of the HMS is dependent on factors such as project scope, resource availability, and complexity. With proper planning and allocation of resources, meeting project deadlines is feasible, ensuring timely delivery of the system.

1. Technical Feasibility Study

1. **Technology Requirements:** Assessing whether the required technologies, such as Java, JSP, Servlets, MySQL, and NetBeans, are readily available and compatible with the existing technology infrastructure of the healthcare facility.
2. **Skillset Availability:** Evaluating whether there are skilled developers and IT professionals available who possess the expertise in the selected technologies and can effectively develop, deploy, and maintain the HMS.
3. **Hardware and Software Compatibility:** Ensuring that the hardware and software components required for hosting the HMS, including servers, databases, and development environments like NetBeans, are compatible and meet the system's requirements.
4. **Scalability:** Determining whether the HMS can accommodate potential future growth in terms of user base, data volume, and functionality expansion without compromising performance or requiring significant architectural changes.
5. **Integration Capability:** Assessing whether the HMS can integrate seamlessly with existing hospital systems, such as electronic health records (EHR), laboratory information systems (LIS), and billing systems, to ensure interoperability and data exchange.
6. **Security Considerations:** Addressing security concerns related to patient data confidentiality, access controls, encryption mechanisms, and compliance with healthcare regulatory standards such as HIPAA (Health Insurance Portability and Accountability Act).
7. **Data Backup and Recovery:** Implementing robust data backup and recovery mechanisms to ensure data integrity and availability in case of system failures, disasters, or data breaches.
8. **Training and Support:** Providing training and support to users, including administrators, doctors, receptionists, and IT personnel, to familiarize them with the HMS functionalities and ensure smooth adoption and usage.

2. Economic Feasibility Study

- **Cost Analysis:** Evaluate the upfront costs associated with developing, implementing, and maintaining the HMS. This includes expenses such as software development, hardware infrastructure, licensing fees, training, and ongoing maintenance costs.

- **Benefit Analysis:** Identify and quantify the potential benefits of implementing the HMS. These benefits may include cost savings from reduced administrative overhead, improved resource utilization, increased operational efficiency, and enhanced patient outcomes.

- **Return on Investment (ROI):** Calculate the ROI of the HMS by comparing the total benefits gained from its implementation to the total costs incurred. A positive ROI indicates that the benefits outweigh the costs and that the investment in the HMS is economically feasible.

- **Payback Period:** Determine the payback period, which is the amount of time it takes for the benefits of implementing the HMS to offset the initial investment costs. A shorter payback period indicates a quicker return on investment and greater economic feasibility.

- **Risk Analysis:** Identify potential risks and uncertainties that may affect the economic feasibility of the HMS, such as changes in technology, regulatory requirements, or market conditions. Assess the impact of these risks on the overall financial viability of the project.

- **Sensitivity Analysis:** Conduct sensitivity analysis to evaluate how changes in key variables, such as cost estimates, benefit projections, or discount rates, impact the economic feasibility of the HMS. This helps assess the robustness of the economic feasibility findings under different scenarios.

- **Cost-Benefit Ratio:** Calculate the cost-benefit ratio by dividing the total benefits gained from the HMS implementation by the total costs incurred. A ratio greater than 1 indicates that the benefits outweigh the costs, supporting the economic feasibility of the project

3 Financial Feasibility Study

- **Cost Analysis:** Evaluate the total costs associated with developing, implementing, and maintaining the HMS. This includes software development costs, hardware expenses, licensing fees, training costs, and ongoing maintenance expenses.

- **Revenue Generation:** Identify potential revenue streams or cost-saving opportunities resulting from the HMS implementation. This may include increased patient volume, improved billing efficiency, reduced administrative costs, and enhanced operational efficiency.

- **Break-Even Analysis:** Calculate the point at which the cumulative benefits from the HMS equal the total costs incurred, indicating when the project becomes financially self-sustaining.

- **Return on Investment (ROI):** Determine the expected ROI by comparing the net financial benefits of the HMS to the total investment. A positive ROI indicates a financially viable project.

- **Payback Period:** Estimate the time it will take for the cumulative financial benefits of the HMS to cover the initial investment costs. A shorter payback period signifies a quicker return on investment.

- **Sensitivity Analysis:** Assess the impact of varying key variables, such as cost estimates, revenue projections, and discount rates, on the financial feasibility of the HMS.

- **Risk Assessment:** Identify potential financial risks and uncertainties associated with the HMS project, such as cost overruns, revenue variability, and market fluctuations. Develop risk mitigation strategies to address these challenges.

3. Operational Feasibility Study

- **Current Process Evaluation:** Evaluate the current hospital management processes, including patient registration, appointment scheduling, record-keeping, and communication among staff.

- **Stakeholder Analysis:** Identify and engage key stakeholders, including hospital administrators, doctors, nurses, receptionists, and IT personnel, to understand their needs, requirements, and expectations from the HMS.

- **System Requirements Definition:** Define the functional and non-functional requirements of the HMS based on stakeholder inputs and organizational goals. This includes features, user interfaces, performance criteria, security requirements, and compliance standards.

- **Gap Analysis:** Identify any gaps or discrepancies between the current processes and the desired functionalities of the HMS. Determine what aspects of hospital management need improvement or enhancement through the implementation of the HMS.

- **User Acceptance Testing (UAT):** Conduct user acceptance testing with representatives from different user groups (administrators, doctors, receptionists) to evaluate the usability, effectiveness, and suitability of the HMS in meeting their operational needs.

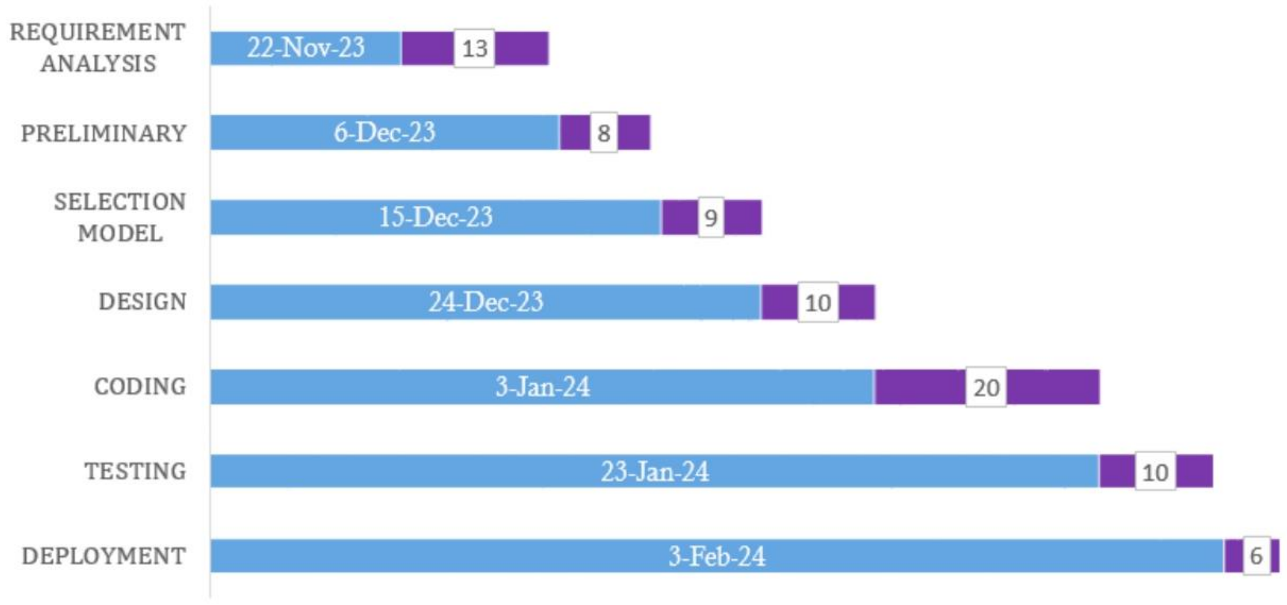
- **Training and Change Management:** Develop a comprehensive training program to familiarize users with the HMS functionalities and workflows. Implement change management strategies to address resistance to change and ensure smooth transition to the new system.

- **Integration with Existing Systems:** Assess the compatibility and integration capabilities of the HMS with existing hospital systems, such as electronic health records (EHR), laboratory information systems (LIS), and billing systems. Ensure seamless data exchange and interoperability.

2.2 STAKEHOLDERS

1. **Hospital Administrators:** Hospital administrators are responsible for overseeing the implementation of the HMS and ensuring that it aligns with the strategic goals and objectives of the healthcare facility. They provide direction, support, and resources for the project.
2. **Medical Practitioners (Doctors, Specialists, Nurses):** Medical practitioners are primary users of the HMS, utilizing it to manage patient records, appointments, treatment plans, and communication with other healthcare professionals. Their input and feedback are essential for ensuring that the system meets their clinical needs.
3. **Receptionists and Administrative Staff:** Receptionists and administrative staff are responsible for managing patient appointments, registrations, and administrative tasks at the front desk. They rely on the HMS to streamline these processes and facilitate smooth patient flow within the hospital.
4. **Patients:** Patients are key stakeholders in the HMS project as they interact with the system to schedule appointments, access medical records, communicate with healthcare providers, and receive care. Their input and satisfaction with the system are critical for improving patient experience and engagement.
5. **IT Personnel:** IT personnel are responsible for the technical aspects of implementing and maintaining the HMS, including software development, system integration, data security, and technical support. They ensure that the system operates effectively and securely within the hospital's IT infrastructure.
6. **Regulatory Authorities:** Regulatory authorities oversee compliance with healthcare regulations and standards, such as HIPAA (Health Insurance Portability and Accountability Act), related to patient data security and privacy. Their input and guidance are essential for ensuring that the HMS meets regulatory requirements.
7. **Software Developers:** Software developers are responsible for designing, developing, and testing the HMS software. They work closely with other stakeholders to understand their requirements and ensure that the system meets their needs effectively.
8. **Training Personnel:** Training personnel are responsible for providing training and support to users of the HMS, including administrators, medical practitioners, receptionists, and IT personnel. They ensure that users are proficient in using the system and maximize its benefits.

Chapter - 3 (Gantt Chart)



Chapter – 4 (Software Requirement Specification)

4.1SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements

Component	**Description**
Computer/Laptop	Minimum: Dual-core processor, 4GB RAM, 100GB HDD Recommended: Quad-core processor, 8GB RAM, SSD
Monitor	Resolution:1920x1080 or higher
Internet Connection	Broadband or high-speed internet

Software Requirements

Component	**Description**
Operating System	Windows, MacOS or any type of OS
Java Development Kit	JDK 8 or later
NetBeans IDE	Neatbeans 8.2 or later , here I'm using NetBeans IDE 8.0.2
Application Server	Apache Tomcat 9 or later / Glassfish Server
Database	MySQL 5.7 or later

4.2FUNCTIONAL REQUIREMENTS

- **Patient Registration:**
  - Ability to register new patients into the system.
  - Capture patient demographics, contact information, and medical history.
  - Assign unique patient identifiers for easy identification and tracking.
- **Appointment Scheduling:**
  - Allow users to schedule, reschedule, and cancel patient appointments.
  - View available time slots for doctors and other healthcare providers.
  - Send automated appointment reminders to patients via email or SMS.
- **Doctor Management:**
  - Maintain a database of doctors with their specialties, contact information, and availability.
  - Assign doctors to specific departments or clinics within the hospital.
  - Enable doctors to view their schedules and manage their appointments.
- **Patient Management:**
  - Maintain comprehensive patient records, including medical history, diagnoses, treatments, and medications.
  - Track patient visits, procedures, and outcomes.

4.3 NON-FUNCTIONAL REQUIREMENTS

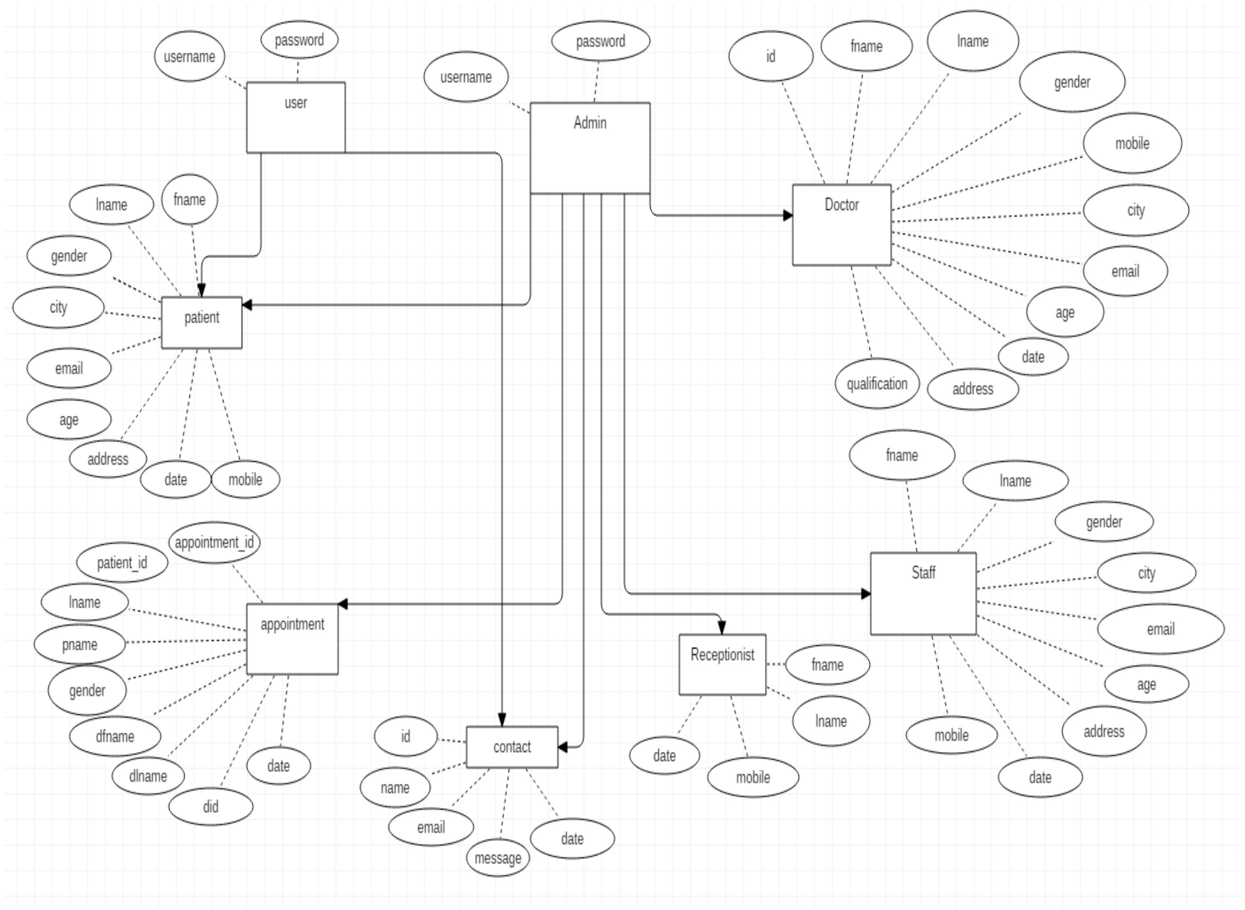
1. **Performance:**
   - The system should respond to user actions and queries promptly, with minimal latency.
   - It should be able to handle concurrent user requests efficiently,
2. **Reliability:**
   - The HMS should be highly reliable, with minimal downtime and system failures.
   - It should have built-in redundancy and failover mechanisms to ensure uninterrupted operation in case of hardware or software failures.
3. **Usability:**
   - The system should have an intuitive and user-friendly interface that is easy to navigate for users of varying technical proficiency.
   - Navigation should be consistent and logical, with clear labeling and organization of features and functionalities.
4. **Security:**
   - The HMS should adhere to industry best practices for data security and privacy, especially regarding patient health information (PHI).
   - Access to sensitive information should be restricted based on user roles and permissions, with strong authentication and authorization mechanisms in place.
5. **Scalability:**
   - The system should be designed to accommodate future growth and increasing user volumes without significant performance degradation.
   - It should be scalable both vertically (adding more resources to existing servers) and horizontally (adding more servers to distribute load).
6. **Interoperability:**
   - The HMS should be interoperable with other healthcare systems and standards, allowing for seamless integration with existing hospital infrastructure and external systems.
7. **Maintainability:**
   - The system should be easy to maintain and update, with modular architecture and well-documented code.
   - Changes and updates should be deployable with minimal disruption to ongoing operations.

.

4.4CONCEPTUAL MODELS (DATA FLOW DIAGRAMS)

1. E-R Diagram

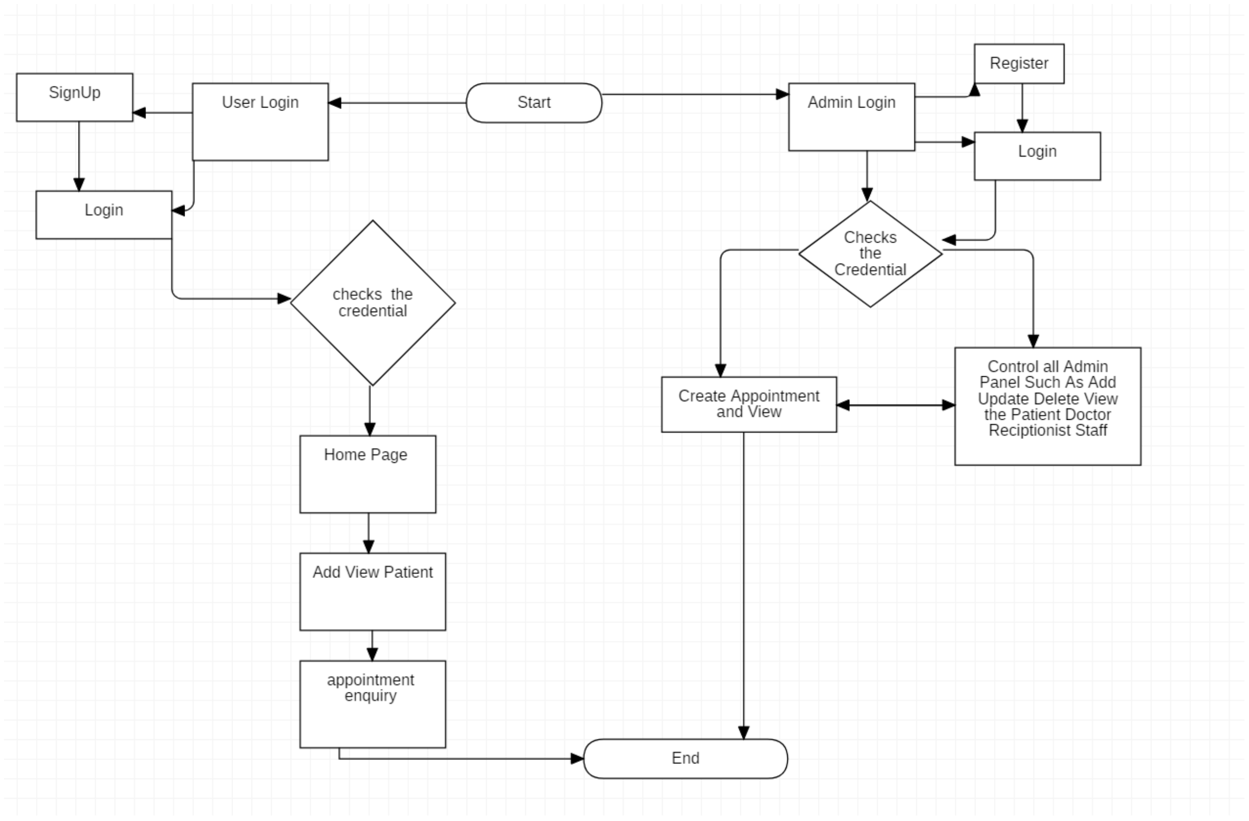
An Entity-Relationship (ER) diagram is a visual representation of the entities, attributes, relationships, and constraints within a database schema. It is a conceptual modeling technique used in database design to depict the structure and organization of data in a system. ER diagrams are essential for designing databases, as they provide a clear and concise overview of the data model and help stakeholders understand the relationships between different entities.



2. Flowchart Diagram

A flowchart is a visual map that shows the steps and decisions in a process. It uses symbols to represent actions and decisions, making it easy to understand. Flowcharts are used in many fields to illustrate and analyze processes, making communication and problem-solving simpler.
-solving.

-

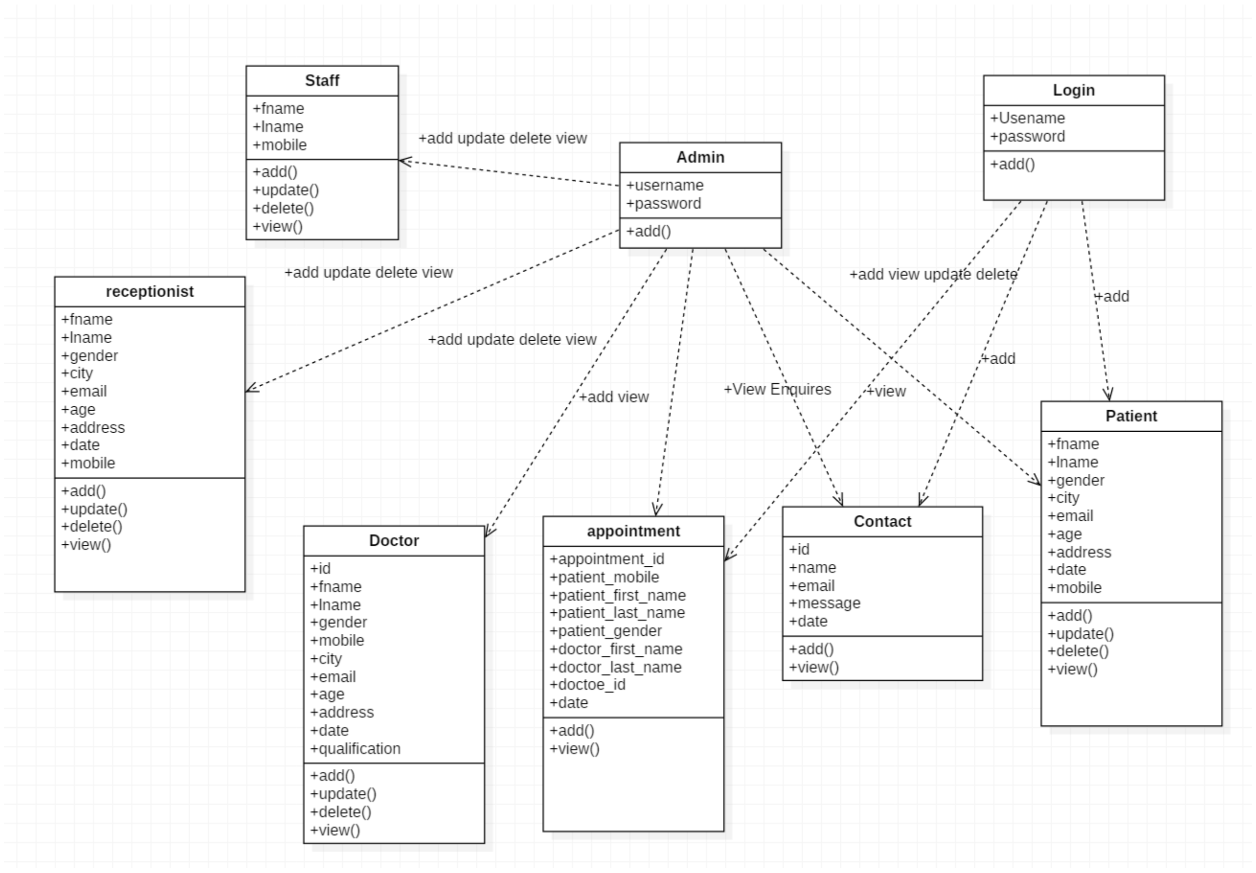


3. Class Diagram

A Class Diagram is a type of static structure diagram in the Unified Modeling Language (UML) that illustrates the structure of a system by showing the classes of objects, their attributes, methods, and relationships among the classes. Class diagrams are widely used in software development to visualize the object-oriented design of a system.

Class diagrams are **useful** for:

Understanding the structure and organization of a system's classes and their relationships.
Communicating the design of a system to stakeholders, including developers, designers, and clients.
Serving as a blueprint for implementing the system in code, especially in object-oriented programming languages like Java, C++, and Python.
Overall, Class Diagrams provide a high-level overview of the static structure of a system, focusing on the classes, their attributes, methods, and relationships, and are an essential tool in software development for designing and documenting object-oriented systems.



4. Use-case Diagram

Actors: These are the people, systems, or devices outside the system that interact with it. They look like stick figures or symbols.

Use Cases: These are the different things the system can do. Each use case represents a specific task or feature that the system can perform. They are shown as ovals.

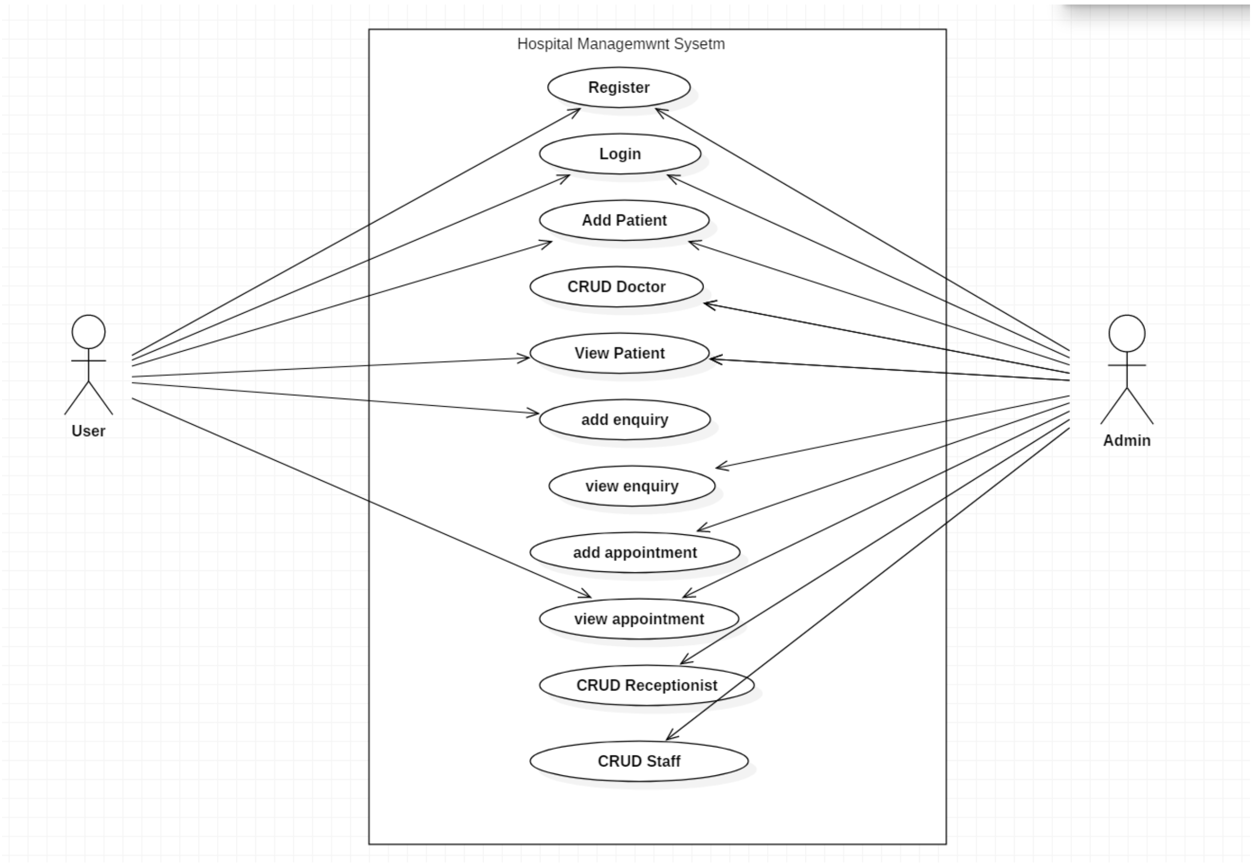
Association: Lines connecting actors to use cases, showing that they interact. Think of it like arrows connecting people to the things they use in the system.

System Boundary: This is like a box around the use cases, showing where the system starts and stops. It helps define what's inside the system and what's outside.

Include Relationship: A dashed arrow showing that one use case includes the functionality of another. It's like saying one task is part of another.

Extend Relationship: Another dashed arrow, but this time showing that one use case can add extra behavior to another under certain conditions. It's like saying, "If something special happens, do this extra thing."

Use case diagrams are useful for:
Requirements Analysis: Figuring out what the system needs to do based on what users want.
System Design: Creating a visual picture of how the system will work, making it easier to plan and build.



5. Activity Diagram

An activity diagram, a component of the Unified Modeling Language (UML), illustrates the dynamic flow within a system or business process. It focuses on actions, decisions, and transitions. Key components include:

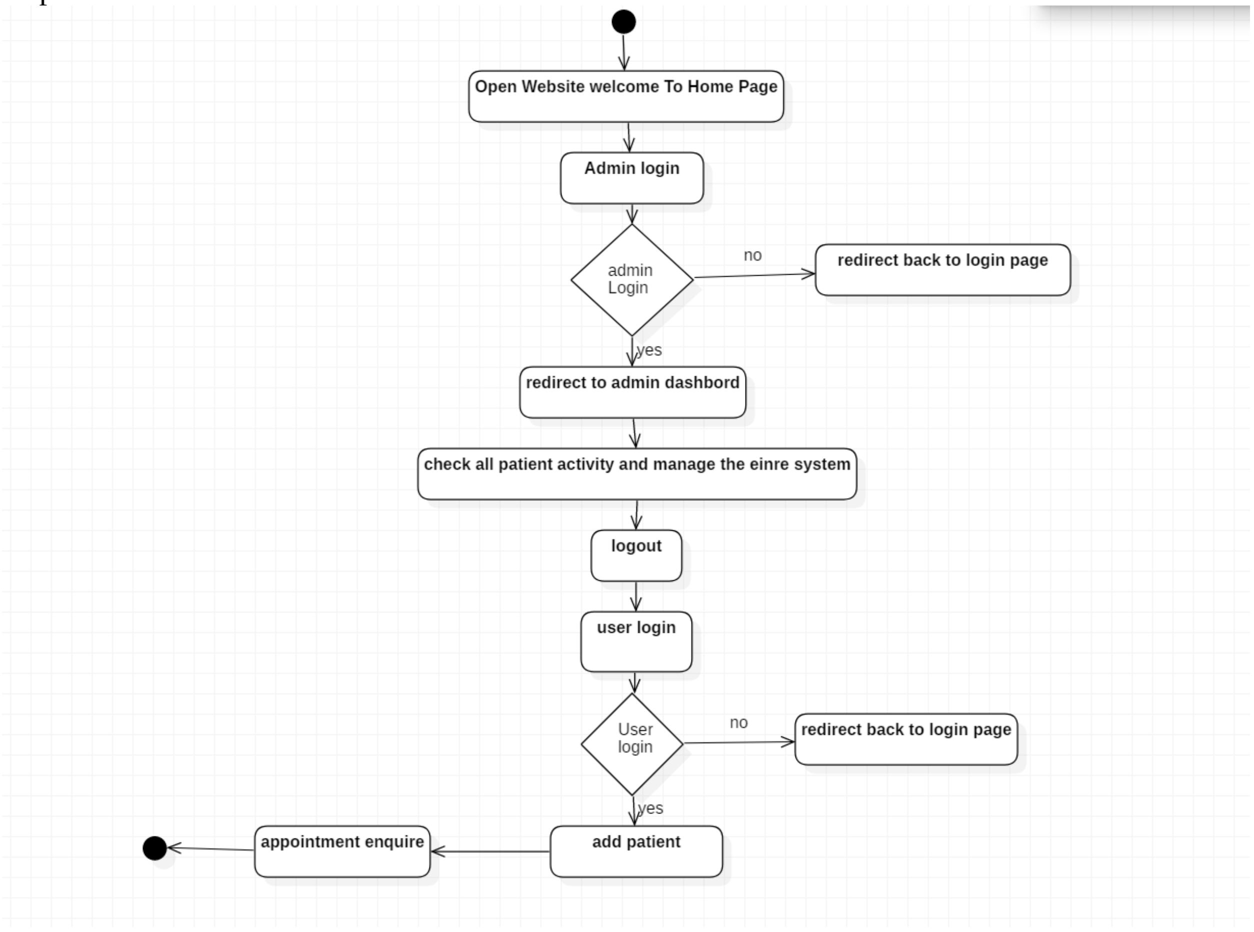
Activity:
Represents a specific action or task.
Depicted as rounded rectangles.
Encompasses simple actions (e.g., sending an email) or complex processes (e.g., order processing).

Control Flow:
Illustrates the sequence of activity execution.
Arrows connect activities, defining the order of execution.

Decision Node:

Marks a decision point where control flow branches based on conditions.
Depicted as diamonds, labeled with decision criteria.

Initial Node:
Marks the starting point of the activity diagram.
Depicted as a solid circle with an outward arrow.

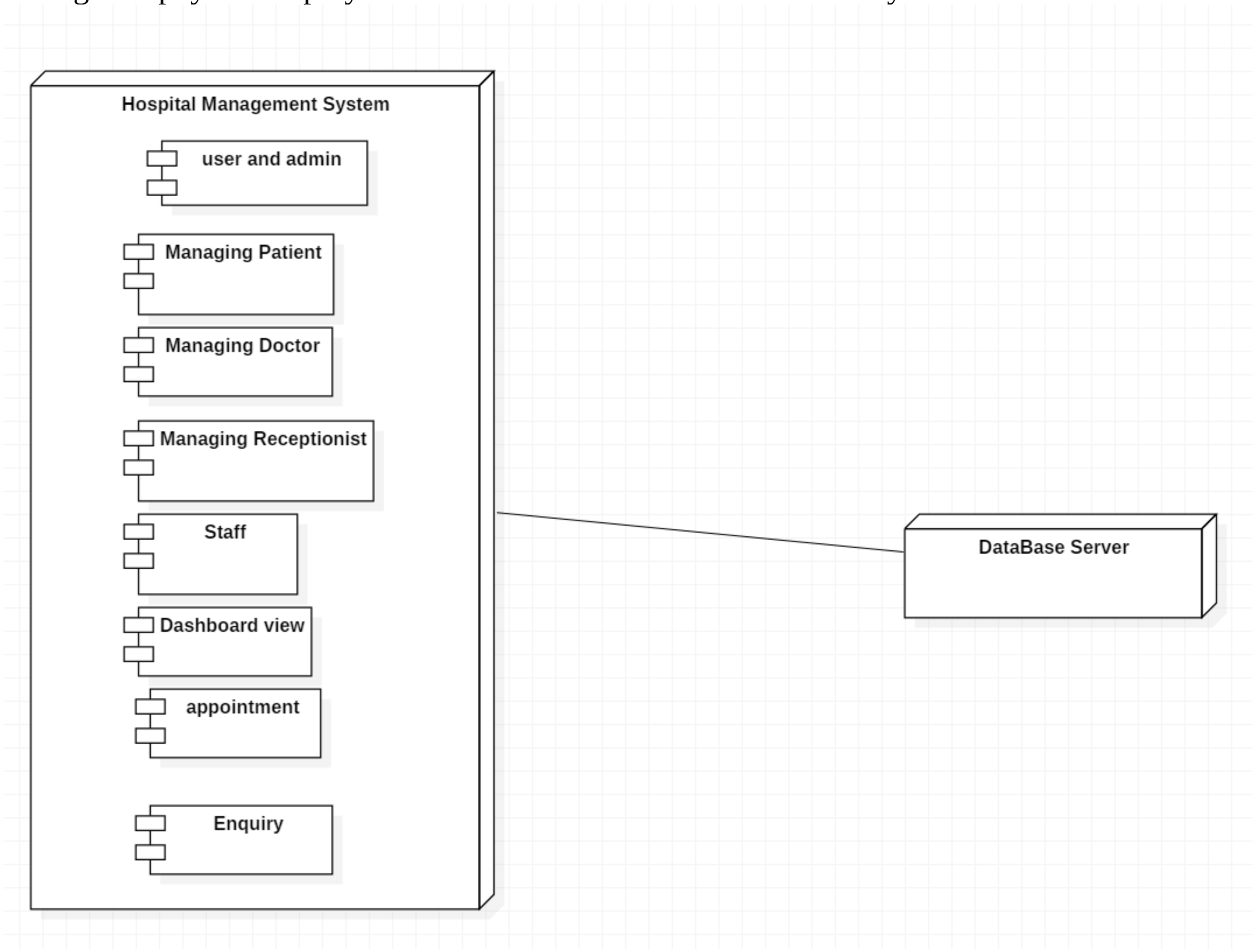

Open Website welcome To Home Page
Admin login
admin Login
no
redirect back to login page
yes
redirect to admin dashbord
check all patient activity and manage the einre system
logout
user login
User login
no
redirect back to login page
yes
add patient
appointment enquire

6. Deployment Diagram

A deployment diagram in UML illustrates how software components and hardware nodes are physically deployed in a distributed system. It uses nodes to represent physical devices and components to represent software artifacts. Key components include nodes, components, artifacts, associations, and deployment specifications.

Nodes: Represent physical hardware or software execution environments.
Components: Represent software artifacts or modules.
Artifacts: Represent physical files or data objects deployed on nodes.
Associations: Show relationships between nodes, components, and artifacts.
Deployment Specification: Specifies deployment details and constraints.

Deployment diagrams are useful for visualizing system architecture, planning deployment strategies, and analyzing deployment requirements. They help stakeholders understand and manage the physical deployment of software and hardware effectively.



7. Sequence Diagram

A sequence diagram in UML illustrates how objects or components in a system interact over time. It uses lifelines, activations, and messages to show the order of actions. Here are key elements:

Lifeline: Represents an object's existence over time, shown as a vertical dashed line labeled with the object's name.

Activation: Represents the period an object is active, depicted as a solid rectangle on the lifeline.

Message: Represents communication between objects, shown as arrows between lifelines, indicating the flow of actions.

Return Message: Represents the response from an object after processing a message, depicted as dashed arrows.

Interaction Fragment: Groups related messages or interactions, allowing for the representation of complex control flow.

Sequence diagrams are useful for:
Describing Object Interactions.
Modeling System Behavior.
Designing Software Systems.
Testing and Debugging.

8. Package Diagram

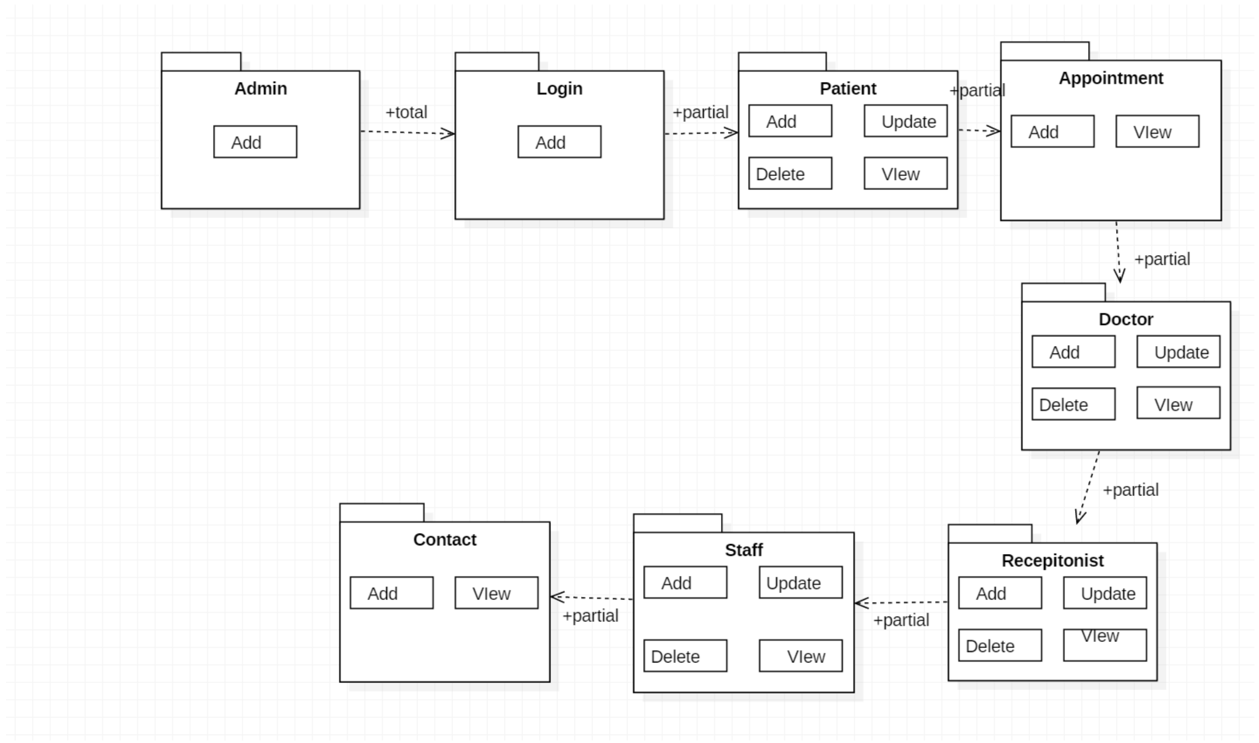
A package diagram in the Unified Modeling Language (UML) provides a high-level view of the organization and structure of a system through the grouping of related elements into packages. It's a way to organize and modularize a system, highlighting dependencies and relationships between different components.
Key Components of a Package Diagram:

Package:
Represents a container for organizing elements.
Typically depicted as a rectangle with a folded corner.
Dependency:

Represents a relationship between packages, indicating that one package depends on another.
Shown as a dashed arrow pointing from the dependent package to the package being depended upon.

Association:

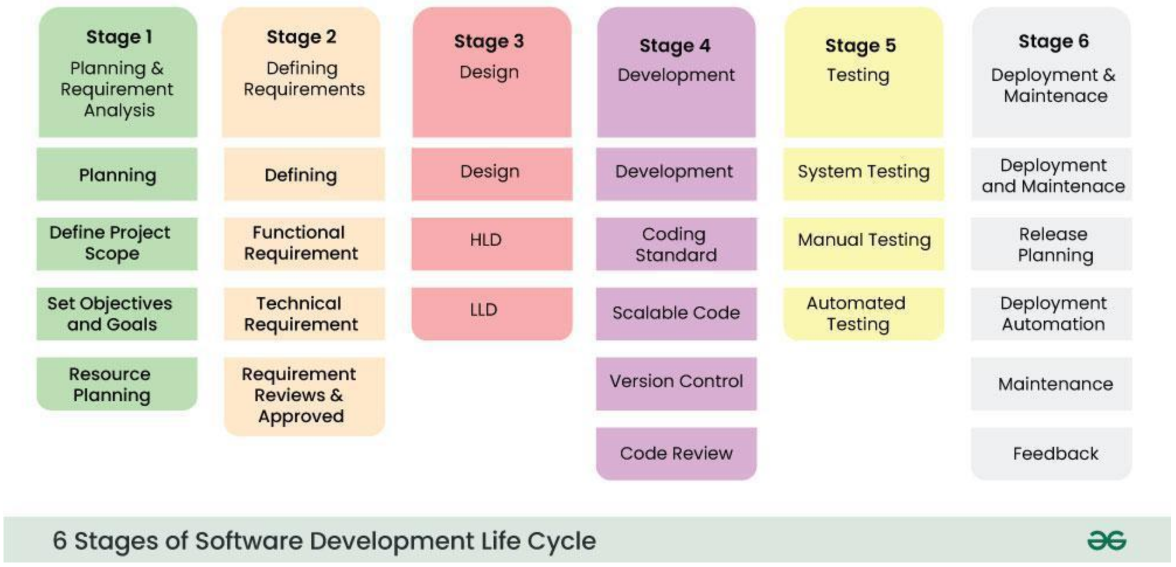
Represents a relationship between classes or components within a package.
Illustrated by a solid line connecting the related elements.


Admin
Add
+total
Login
Add
+partial
Patient
Add
Update
Delete
VIew
+partial
Appointment
Add
VIew
+partial
Doctor
Add
Update
Delete
VIew
+partial
Recepitonist
Add
Update
Delete
VIew
+partial
Staff
Add
Update
Delete
VIew
+partial
Contact
Add
VIew

Chapter – 5 (Software Development Life Cycle)

5.1 What is SDLC?

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.
SDLC is a process followed for software building within a software organization. SDLC consists of a precise plan that describes how to develop, maintain, replace, and enhance specific software. The life cycle defines a method for improving the quality of software and the all-around development process.
SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer's expectations and fits within the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process.



5.2 SDLCS MODELS

To this day, we have more than 50 recognized SDLC models in use. But None of them is perfect, and each brings its favourable aspects and disadvantages for a specific software development project or a team.

In this article, I've listed the top five most popular SDLC models below.

2. Agile Model

The agile model was mainly designed to adapt to changing requests quickly. The main goal of the Agile model is to facilitate quick project completion. The agile model refers to a group of development processes. These processes have some similar characteristics but also possess certain subtle differences among themselves.

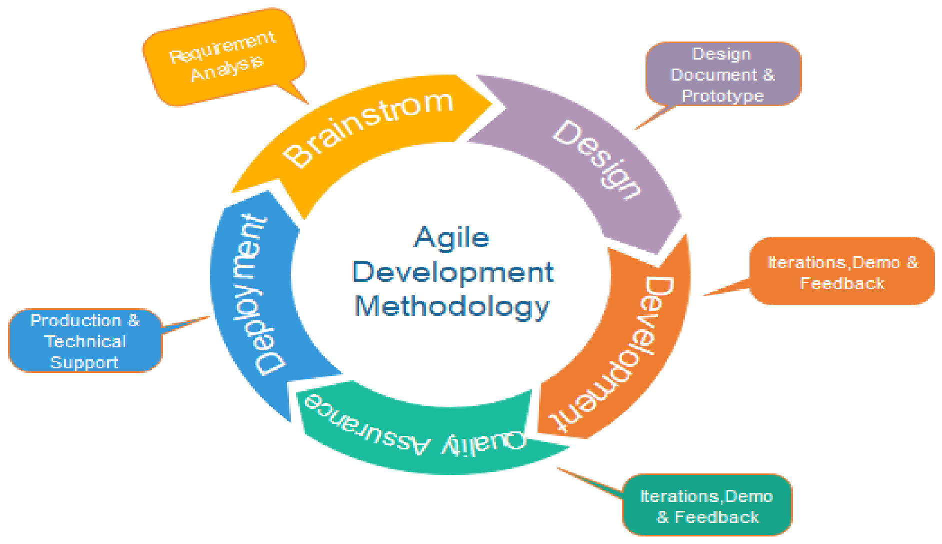

Requirement Analysis
Design Document & Prototype
Brainstrom
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

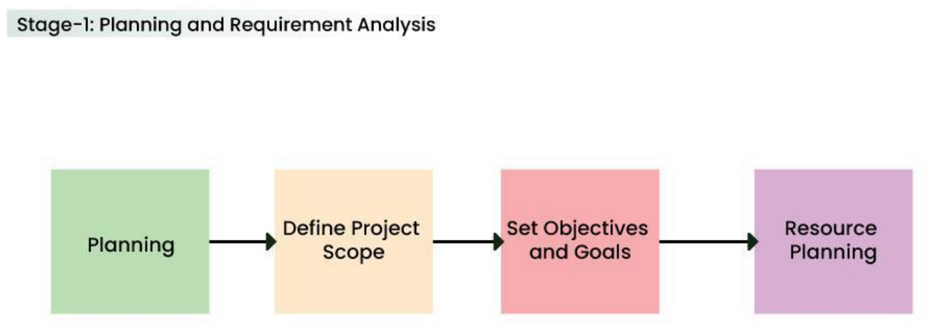
Fig. Agile Model

5.3 PHASES OR STAGES OF SDLC ARE AS FLLOWS:

Stage-1: Planning and Requirement Analysis

Planning is a crucial step in everything, just as in software development. In this same stage, requirement analysis is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.

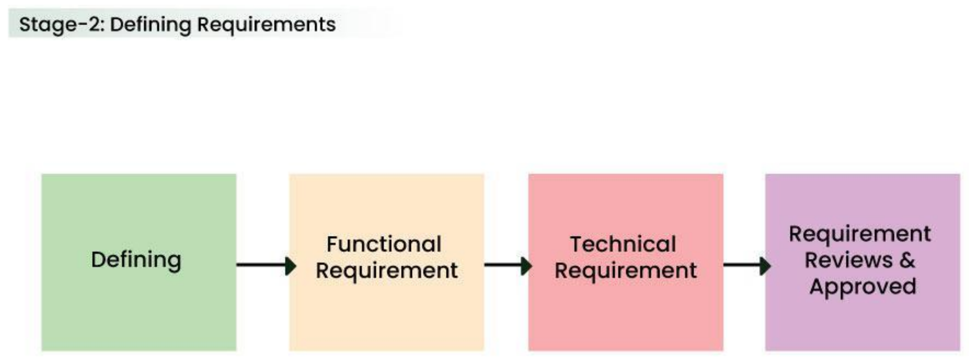
The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.



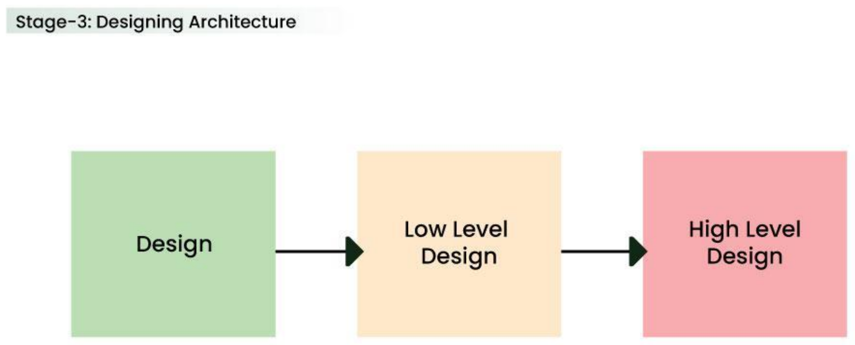
Stage-2: Defining Requirements

In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.

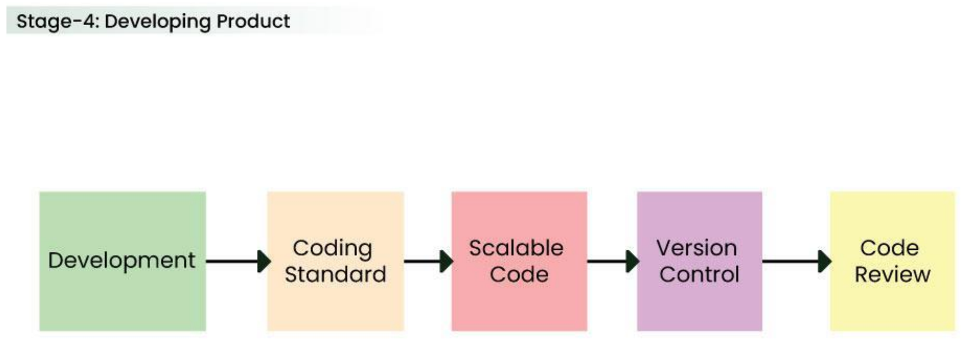
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.


Stage-2: Defining Requirements
Defining
Functional Requirement
Technical Requirement
Requirement Reviews & Approved

Stage-3: Designing Architecture
SRS is a reference for software designers to come up with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).
This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the most practical and logical design is chosen for development.



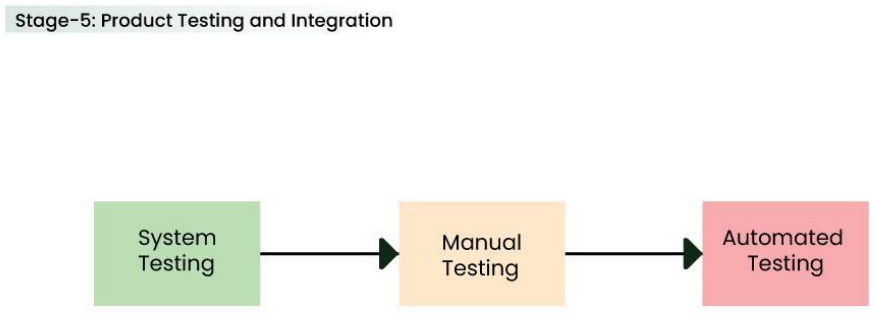
Stage-4: Developing Website(product)
At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.



Stage-5: Product Testing and Integration
After the development of the website or product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at

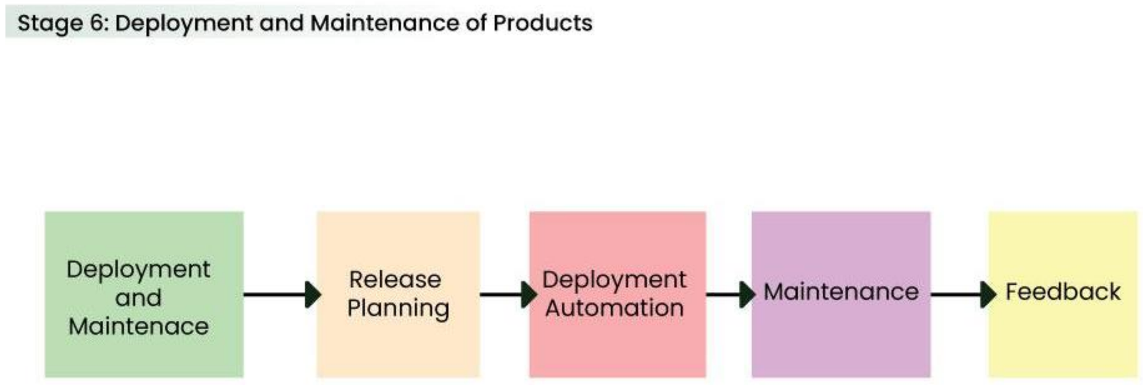
this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

Documentation, Training, and Support: Software documentation is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions, and maintenance. Documentation also provides information about how to use the product. Training in an attempt to improve the current or future employee performance by increasing an employee's ability to work through learning, usually by changing his attitude and developing his skills and understanding.



Stage 6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the product's supervision.


Stage 6: Deployment and Maintenance of Products
Deployment and Maintenace
Release Planning
Deployment Automation
Maintenance
Feedback

Stakeholders, including clients, end-users, project managers, and developers, play crucial roles throughout the SDLC. They contribute to requirements gathering, provide feedback during development, and ensure that the final product aligns with business objectives.

Chapter – 6 (Desigining)

6.1 SYSTEM DESIGN

Architecture:

The Hospital Management System follows a three-tier architecture:

1. **Client Tier:**
   - **Front-End Technologies:** JSP, HTML, CSS, JS.
   - **User Interface (UI):** The client-side interface provides a user-friendly environment for doctors, receptionists, and administrators to interact with the system.
2. **Middle Tier:**
   - **Server-Side Technology:** Servlet.
   - **Business Logic:** Servlets handle HTTP requests and responses, executing business logic and managing the flow of information between the front-end and back-end.
3. **Data Tier:**
   - **Back-End Database:** MySQL.
   - **Data Storage:** The relational database stores information related to patients, doctors, appointments, receptionists, and staff.

Modules:

Admin Module:

- Admin has the authority to manage doctors, receptionists, patients, appointments, and staff.
- Tasks include adding, deleting, and updating doctor, receptionist, patient, and staff records.
- Admin can view lists of doctors, receptionists, and staff.
- Admin can add/view appointments.

Doctor Module:

- Doctors can access appointment details and the patient list.
- They can view their appointments and the list of patients assigned to them.

Receptionist Module:

- Receptionists can add, edit, and view appointments.
- They can manage patient records, including adding, editing, and viewing patient details.

Technology Stack:

- **Front-End:**
    - JSP: Used for dynamic content generation on the server side.
    - HTML, CSS, JS: Standard web technologies for creating the user interface.
- **Server-Side:**
    - Servlet: Handles HTTP requests and responses, implementing business logic.
- **Back-End:**
    - MySQL: Relational database for storing and managing data records.
- **Server:**
    - MySQL Server: Hosts the database and manages data retrieval and storage.

Scalability and Performance:

- The system is designed to handle a growing number of patients, doctors, and staff.
- Techniques such as database indexing, query optimization, and connection pooling are implemented for optimal performance.

Security:

Security measures are implemented to safeguard sensitive data and ensure user privacy

Encryption mechanisms, secure authentication, and authorization controls are in place.

Conclusion:

The Hospital Management System project, designed with a clear three-tier architecture and specific modules for each user role, aims to streamline hospital activities and enhance efficiency in managing patient records, appointments, and staff. The technology stack ensures a robust and scalable solution, while security measures protect against unauthorized access and data breaches.

6.2 DATA DESIGN

Login Table:

Login
username
password

Doctor Table:

Attribute Name
id
fname
lname
gender
mobile
city
email
address
date
qualification

Staff

Attribute Name
fname
lname
gender
mobile
city
email
address
date

Recp

Attribute Name
fname
lname
mobile
date

appointment Table:

Attribute Name
appointment_id
patient_mobile
patient_first_name
patient_last_name
patient_gender
doctor_first_name
doctor_last_name
appointmrnt_date

Patient

Attribute Name
fname
lname
gender
mobile
city
email
age
address
date

admin

Attribute Name
username
email

user details

Attribute Name
fname
lname
mobile
date

6.3DATA INTEGRITY AND CONSTRAINT

Admin Table:

ATTRIBUTE NAME	DATATYPE	CONTRAINTS
username	Varchar(100)	UNIQUE
Password	Varchar(100)	UNIQUE

Doctor Table:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
id	Int	PRIMARY KEY, UNIQUE
fname	Varchar(100)	
lname	Varchar(100)	
gender	Varchar(100)	
mobile	Varchar(100)	
city	Varchar(100)	
email	Varchar(100)	
age	Varchar(100)	
address	Varchar(100)	
date	Varchar(100)	
qualification	Varchar(100)	

login Table:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
username	Varchar(15)	
password	Varchar(15)	

Patient Table:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
fname	Varchar(100)	
lname	Varchar(100)	
gender	Varchar(100)	
city	Varchar(100)	
email	Varchar(100)	
age	Varchar(100)	
address	Varchar(100)	
date	Varchar(100)	
mobile	Varchar(100)	PRIMARY KEY,

Recp Table:

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
fname	Varchar(30)	
lname	Varchar(30)	
mobile	Varchar(255)	PRIMARY KEY

Staff Table

ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
fname	Varchar(100)	
lname	Varchar(100)	
gender	Varchar(100)	
city	Varchar(100)	
email	Varchar(100)	
age	Varchar(100)	
address	Varchar(100)	
date	Varchar(100)	
mobile	Varchar(100)	PRIMARY KEY,

Appointment Table

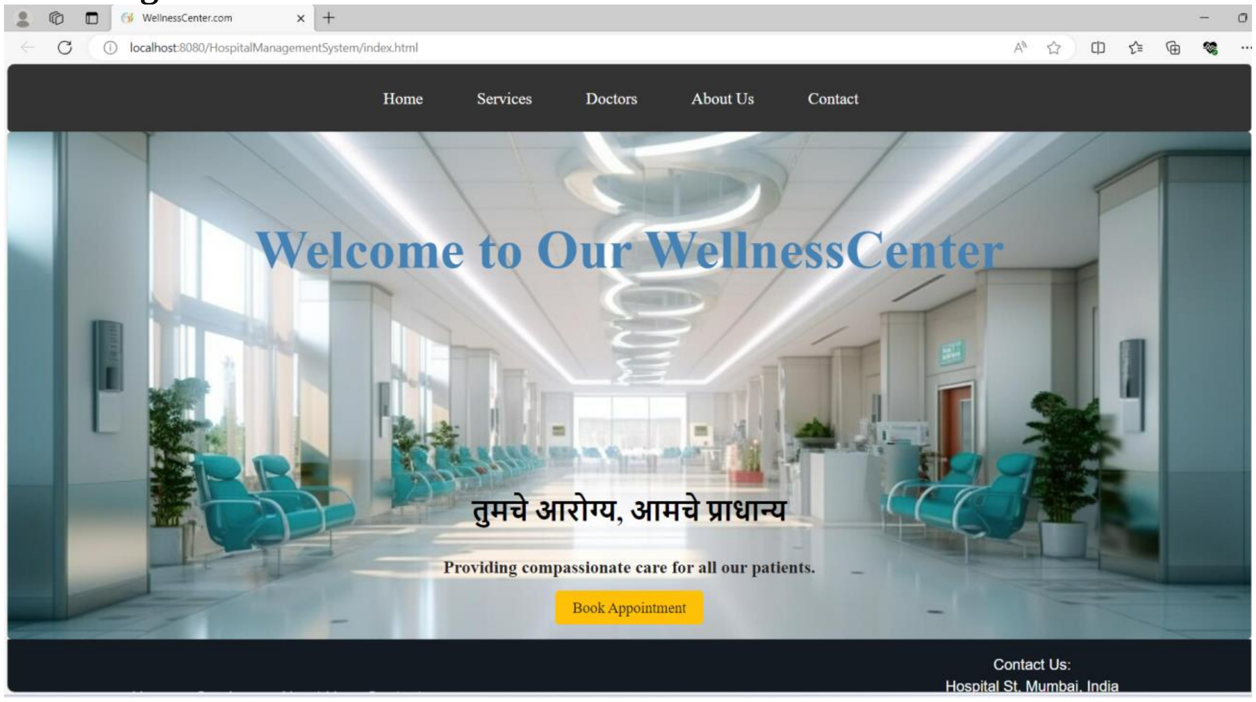
ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
id	Int	PRIMARY KEY, UNIQUE
pmobile	Varchar(100)	FORIGN_KEY
fname	Varchar(100)	
lname	Varchar(100)	
gender	Varchar(100)	
dname	Varchar(100)	
dlname	Varchar(100)	
did	int(100)	FORIGN_KEY
date	Varchar(100)	

Contact Table

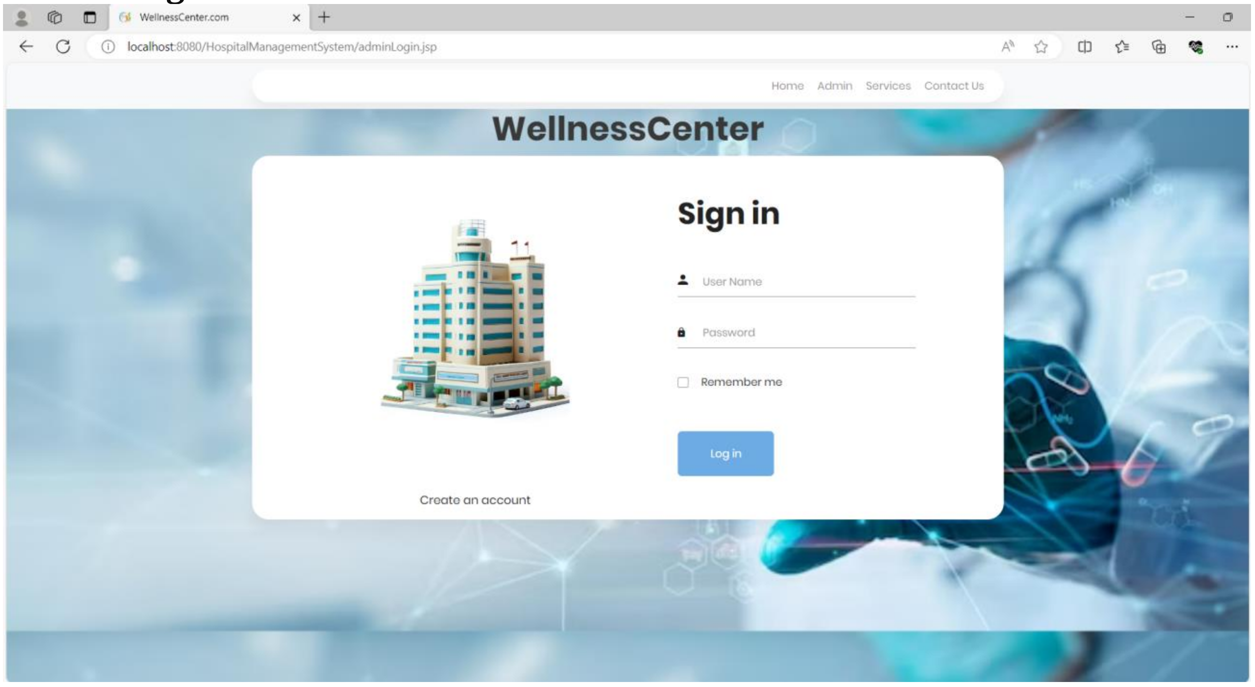
ATTRIBUTE NAME	DATATYPE	CONSTRAINTS
id	int(30)	PAIMARY_KEY
name	Varchar(30)	
email	Varchar(30)	
message	Varchar(255)	
Created_at	Varchar(255)	

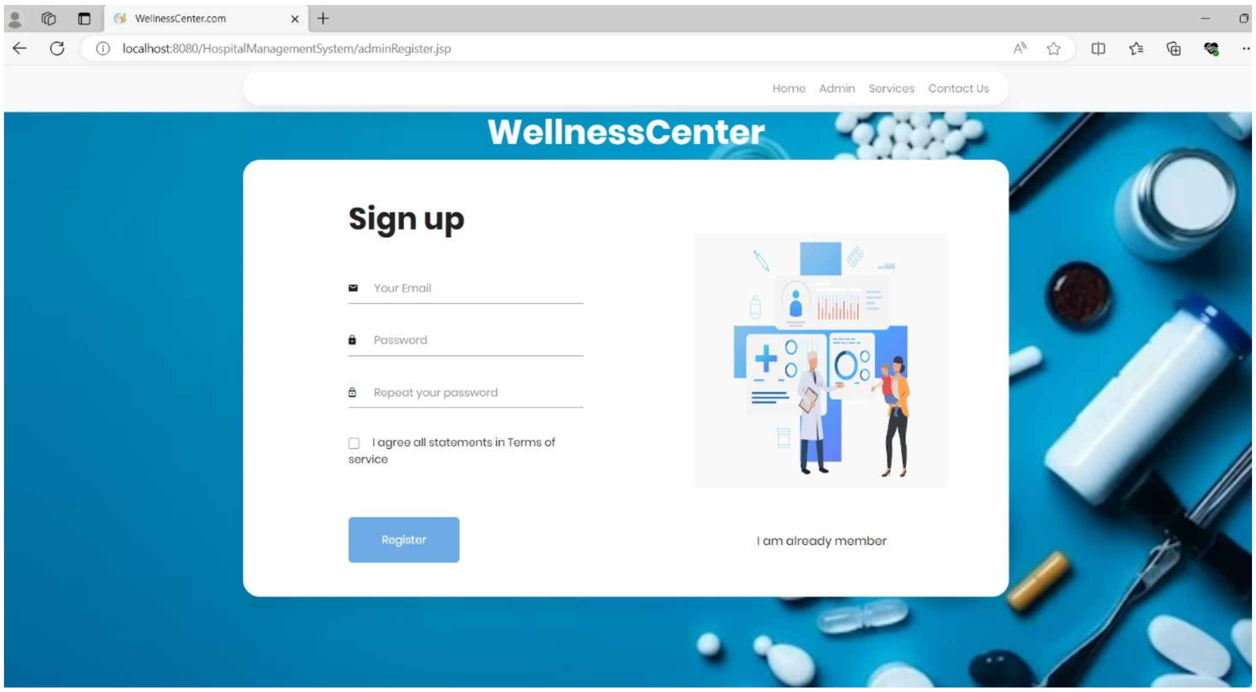
6.4USER INTERFACE AND DESIGN

Main Page:

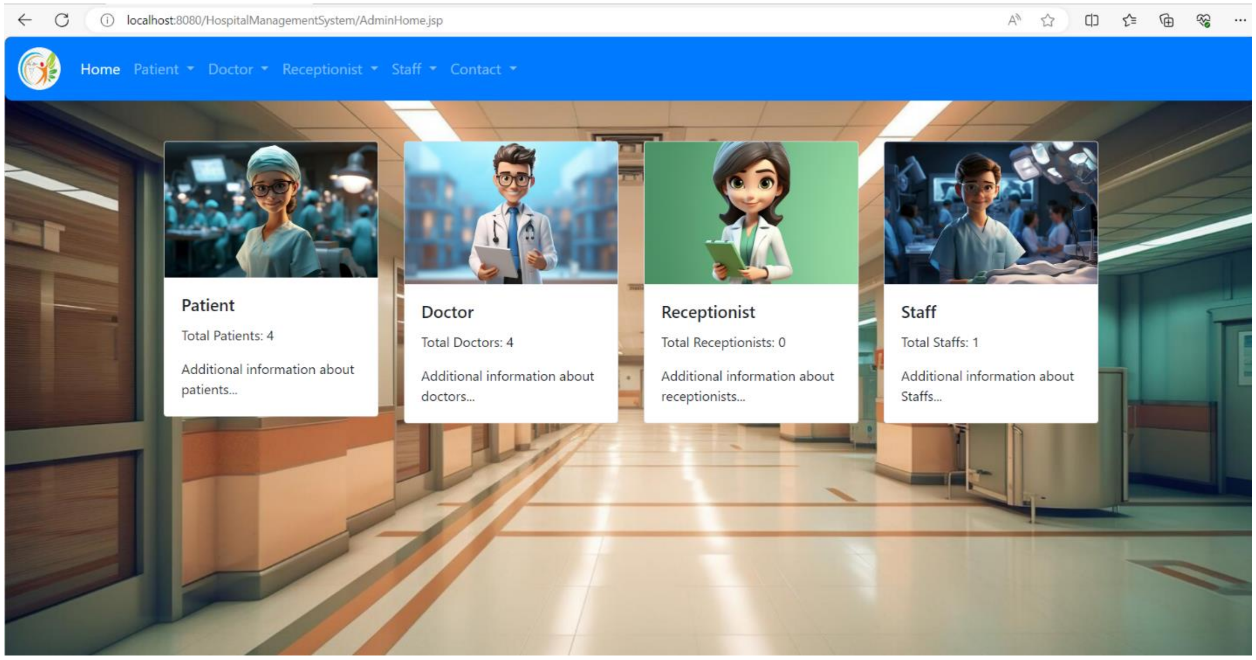

WellnessCenter.com
localhost:8080/HospitalManagementSystem/index.html
Home
Services
Doctors
About Us
Contact
Welcome to Our WellnessCenter
तुमचे आरोग्य, आमचे प्राधान्य
Providing compassionate care for all our patients.
Book Appointment
Contact Us:
Hospital St, Mumbai, India

Admin Login:





AdminSide



Patient

Total Patients: 4

Additional information about patients...

Doctor

Total Doctors: 4

Additional information about doctors...

Receptionist

Total Receptionists: 0

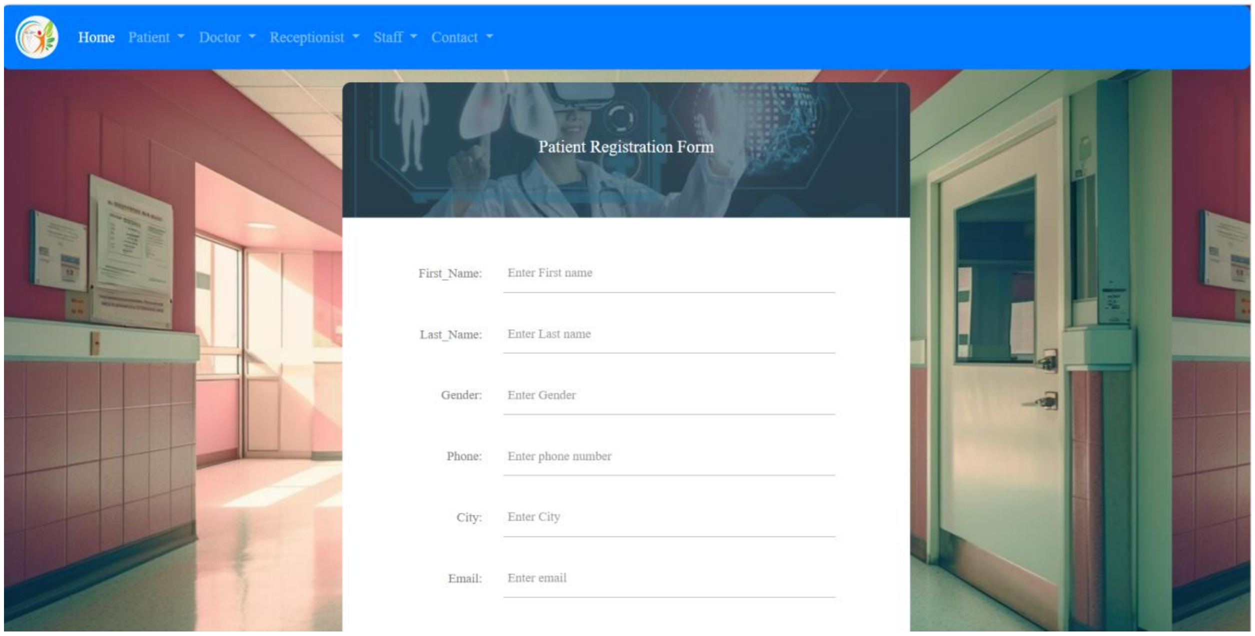
Additional information about receptionists...

Staff

Total Staffs: 1

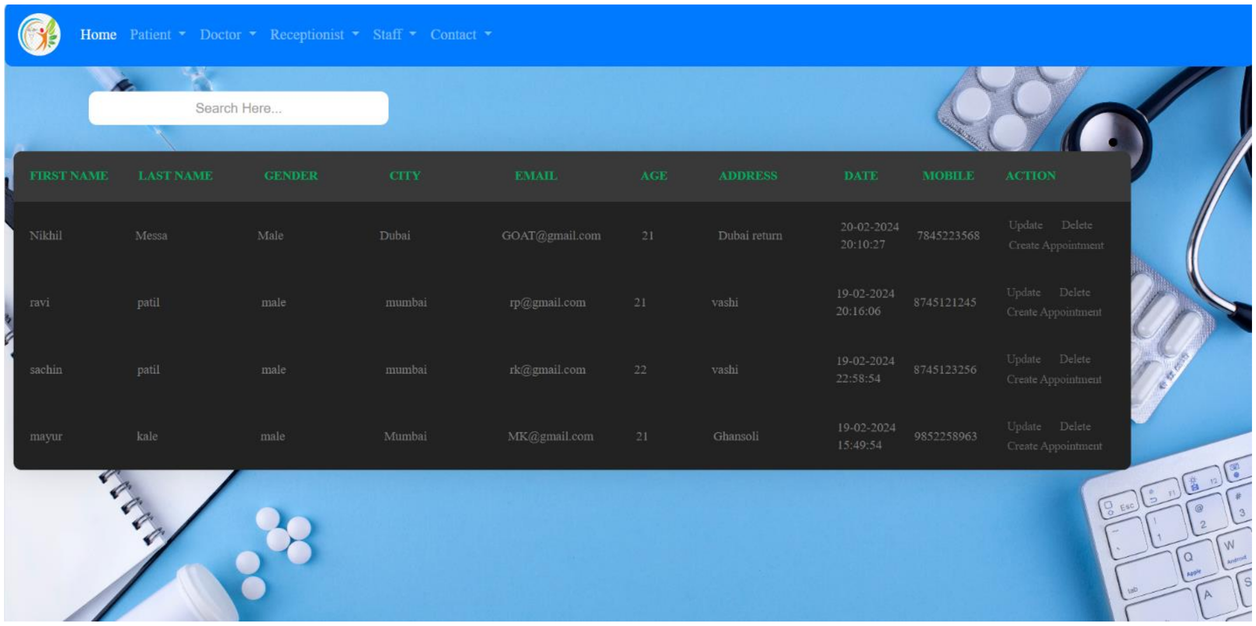
Additional information about Staffs...

AddPatient



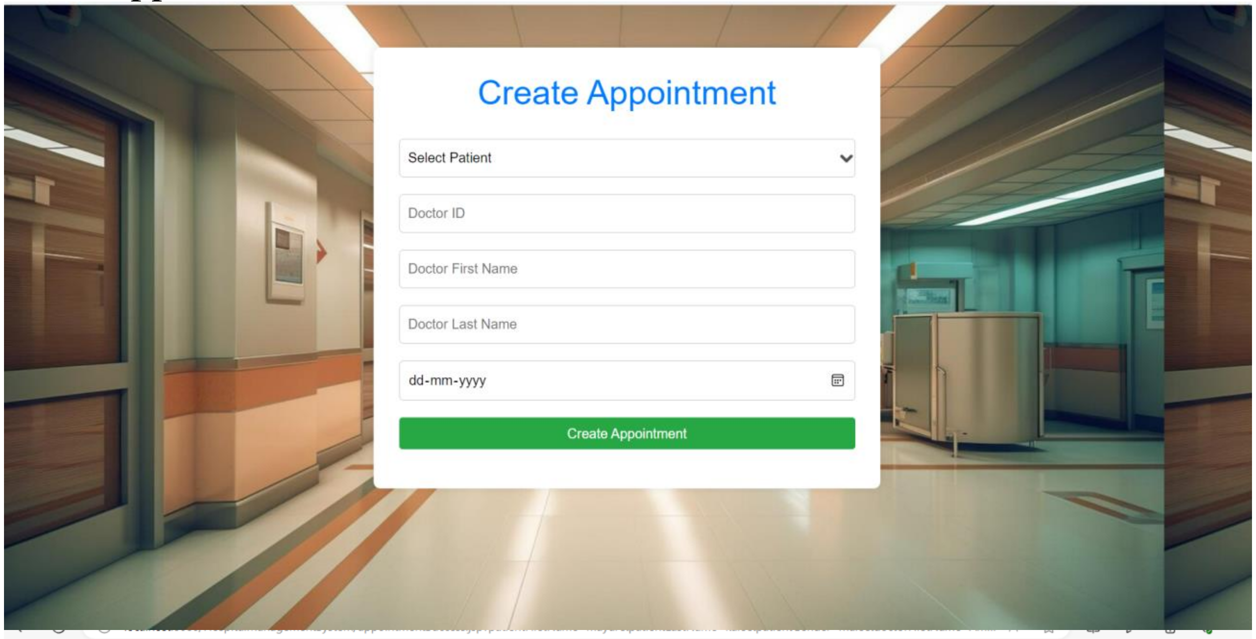
Patient Registration Form

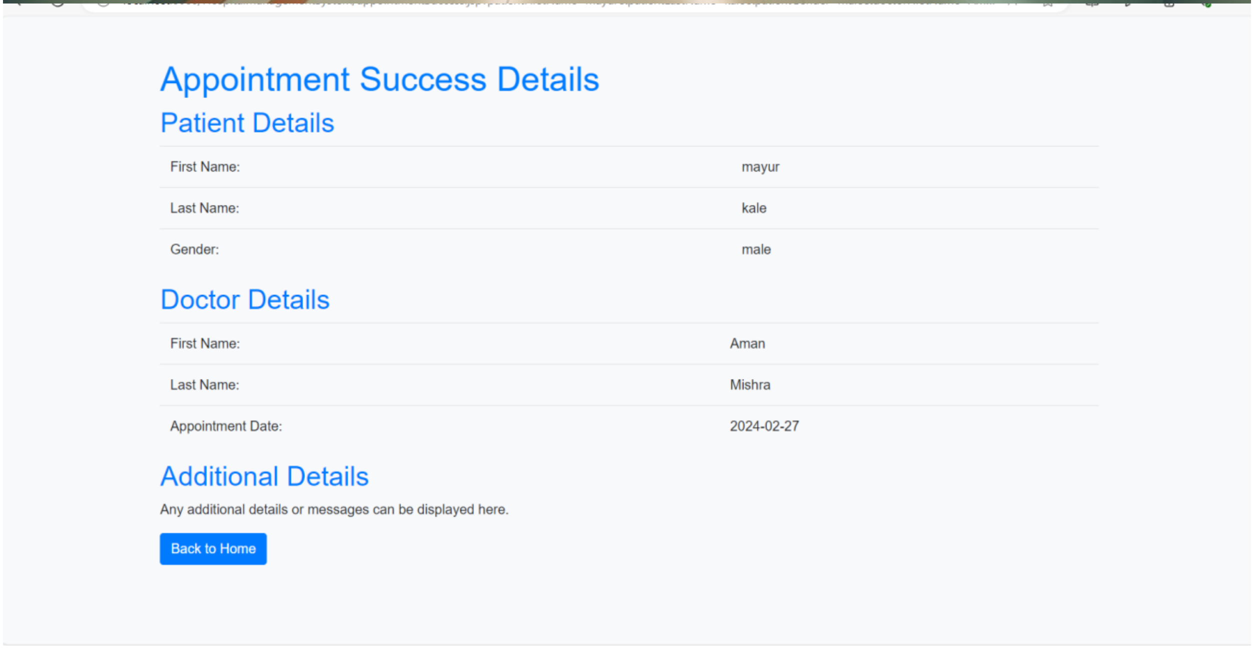
First_Name:	Enter First name
Last_Name:	Enter Last name
Gender:	Enter Gender
Phone:	Enter phone number
City:	Enter City
Email:	Enter email



FIRST NAME	LAST NAME	GENDER	CITY	EMAIL	AGE	ADDRESS	DATE	MOBILE	ACTION
Nikhil	Messa	Male	Dubai	GOAT@gmail.com	21	Dubai return	20-02-2024 20:10:27	7845223568	Update Delete Create Appointment
ravi	patil	male	mumbai	rp@gmail.com	21	vashi	19-02-2024 20:16:06	8745121245	Update Delete Create Appointment
sachin	patil	male	mumbai	rk@gmail.com	22	vashi	19-02-2024 22:58:54	8745123256	Update Delete Create Appointment
mayur	kale	male	Mumbai	MK@gmail.com	21	Ghansoli	19-02-2024 15:49:54	9852258963	Update Delete Create Appointment

Createappointment




Appointment Success Details

Patient Details

First Name:	mayur
Last Name:	kale
Gender:	male

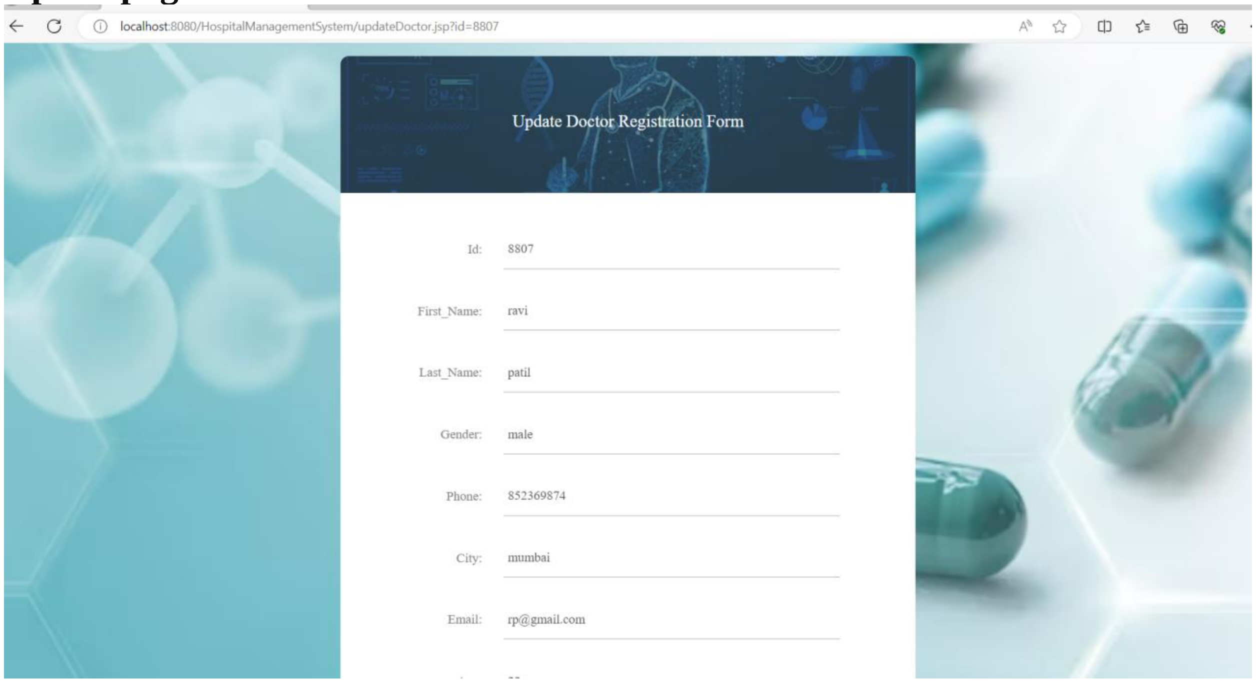
Doctor Details

First Name:	Aman
Last Name:	Mishra
Appointment Date:	2024-02-27

Additional Details

Any additional details or messages can be displayed here.

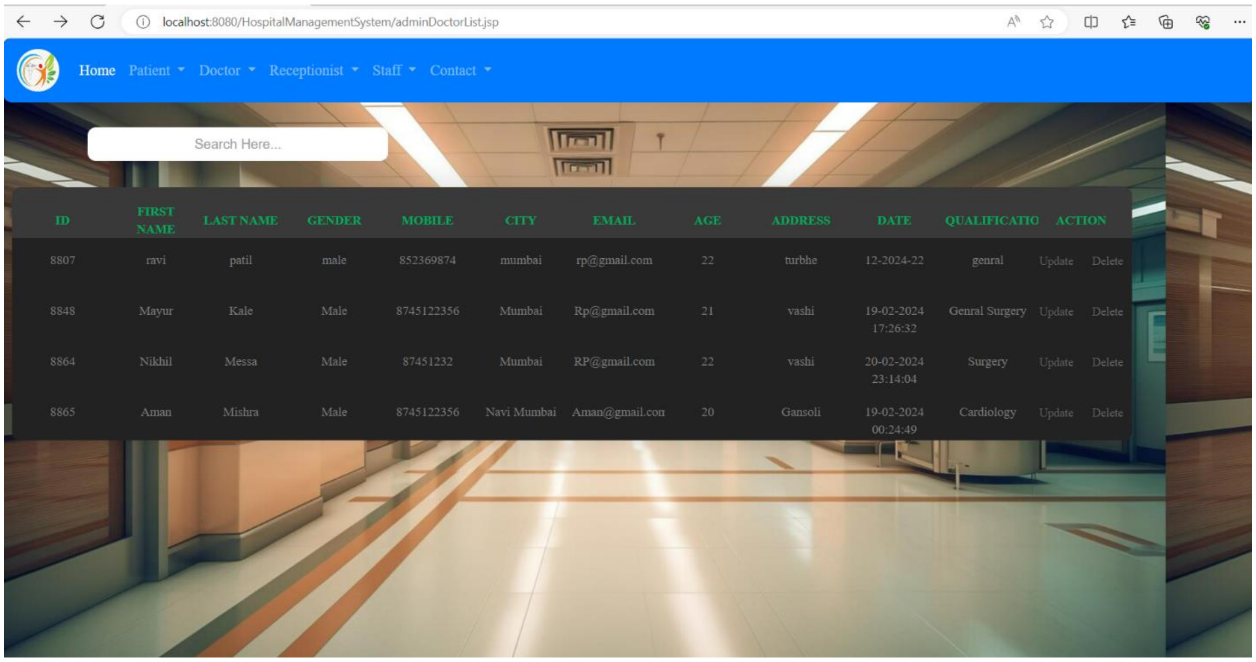
Updatepage



localhost:8080/HospitalManagementSystem/updateDoctor.jsp?id=8807

Update Doctor Registration Form

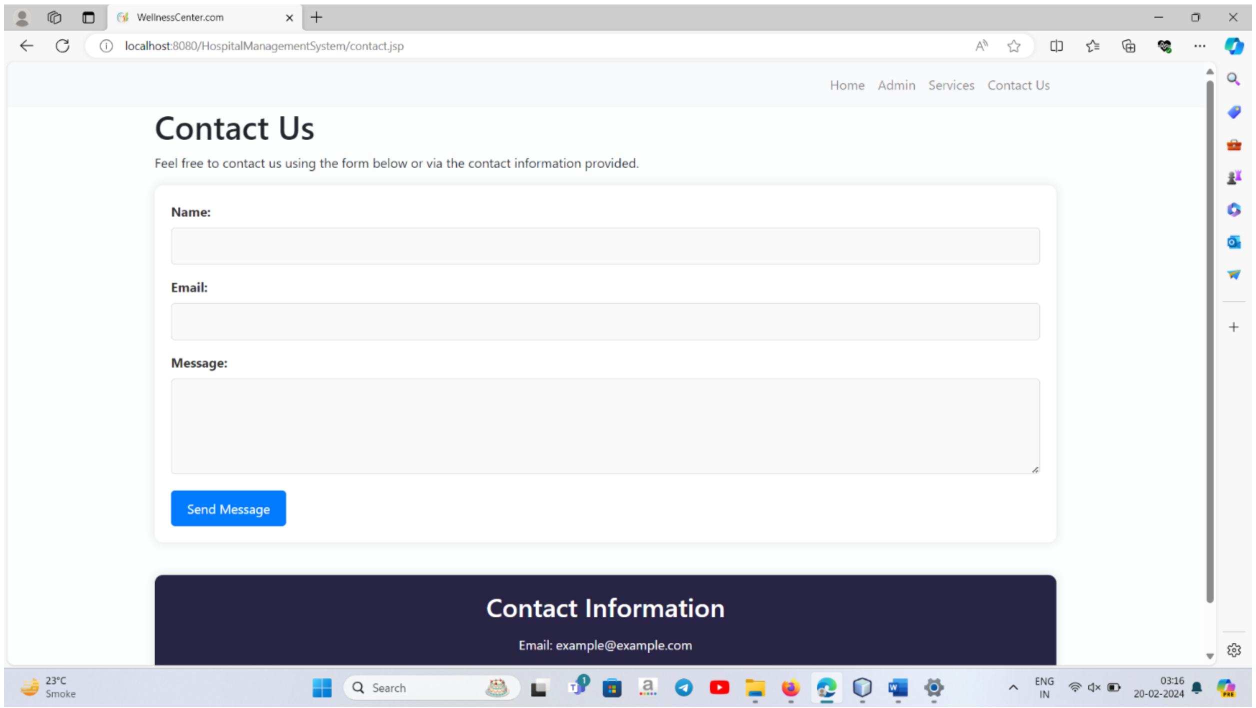
Id:	8807
First_Name:	ravi
Last_Name:	patil
Gender:	male
Phone:	852369874
City:	mumbai
Email:	rp@gmail.com



localhost:8080/HospitalManagementSystem/adminDoctorList.jsp

ID	FIRST NAME	LAST NAME	GENDER	MOBILE	CITY	EMAIL	AGE	ADDRESS	DATE	QUALIFICATIO	ACTION	
8807	ravi	patil	male	852369874	mumbai	rp@gmail.com	22	turbhe	12-2024-22	genral	Update	Delete
8848	Mayur	Kale	Male	8745122356	Mumbai	Rp@gmail.com	21	vashi	19-02-2024 17:26:32	Genral Surgery	Update	Delete
8864	Nikhil	Messa	Male	87451232	Mumbai	RP@gmail.com	22	vashi	20-02-2024 23:14:04	Surgery	Update	Delete
8865	Aman	Mishra	Male	8745122356	Navi Mumbai	Aman@gmail.con	20	Gansoli	19-02-2024 00:24:49	Cardiology	Update	Delete

Contact


WellnessCenter.com
localhost:8080/HospitalManagementSystem/contact.jsp
Home Admin Services Contact Us

Contact Us

Feel free to contact us using the form below or via the contact information provided.

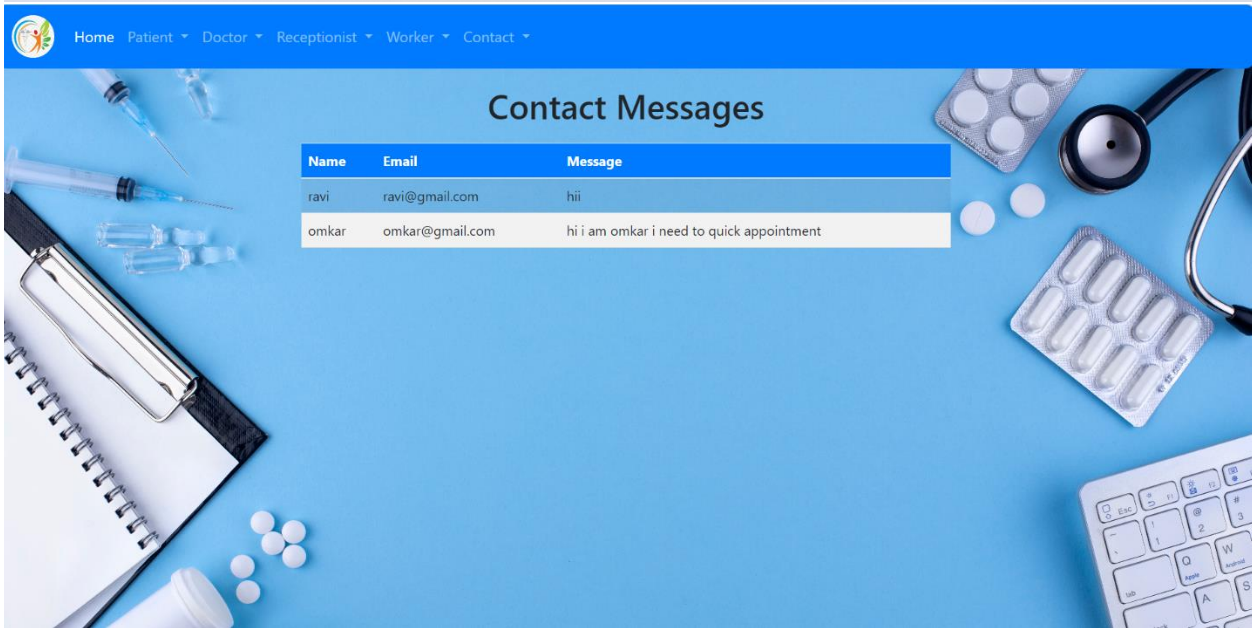
Name:

Email:

Message:

Contact Information

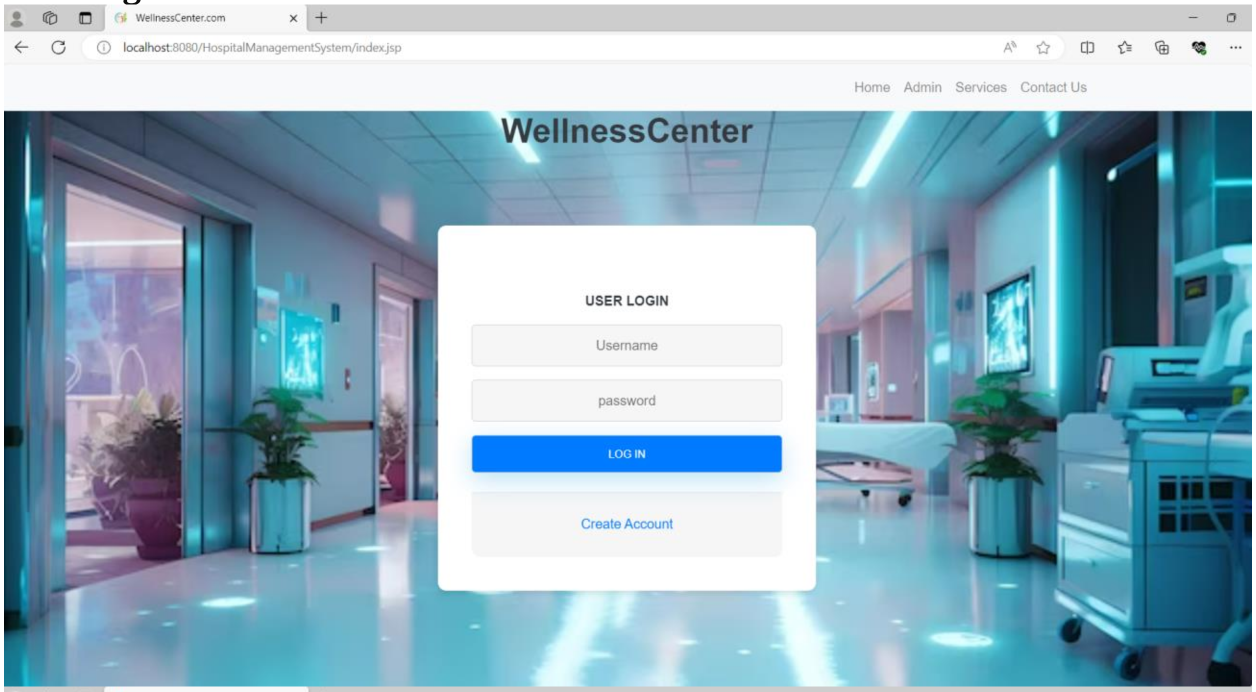
Email: example@example.com

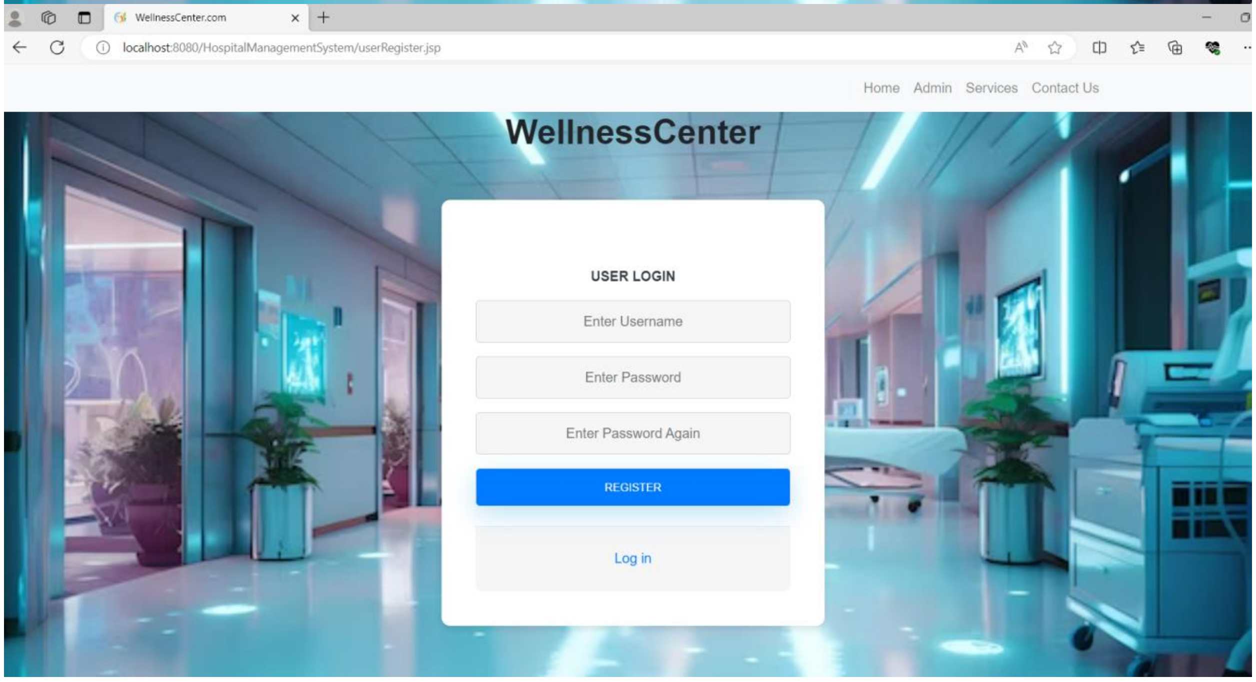


Contact Messages

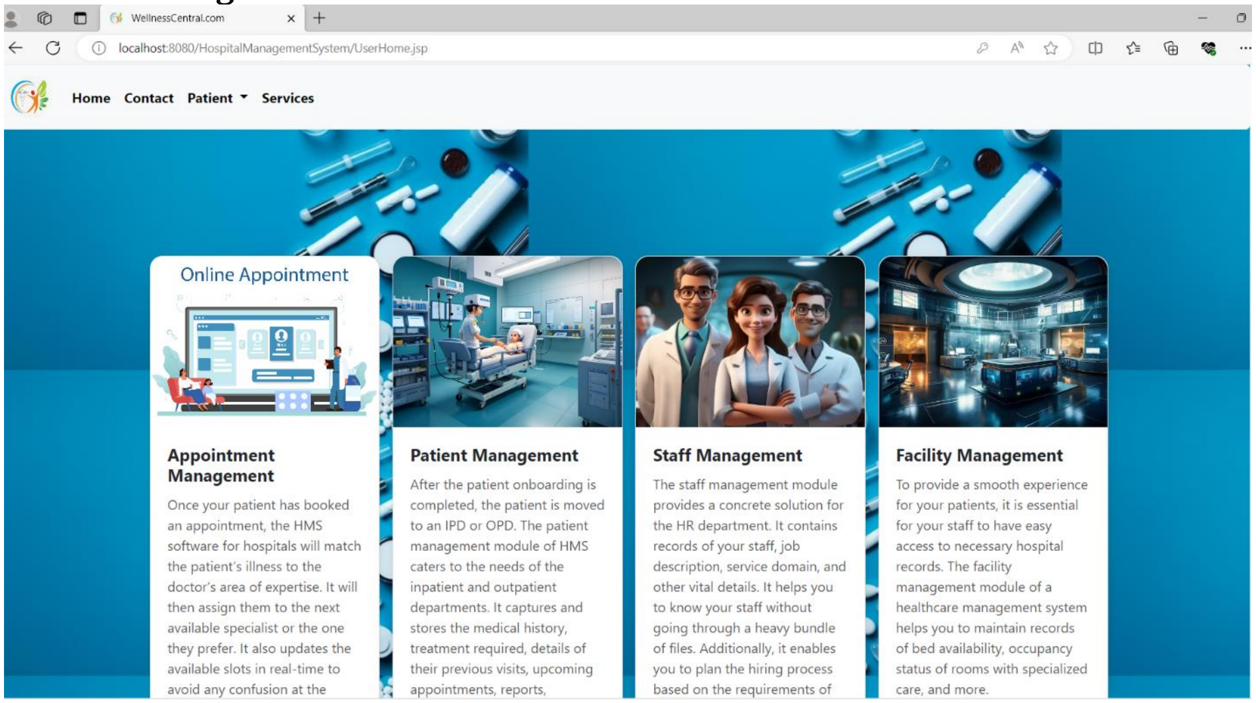
Name	Email	Message
ravi	ravi@gmail.com	hii
omkar	omkar@gmail.com	hi i am omkar i need to quick appointment

UserLogin


WellnessCenter.com
localhost:8080/HospitalManagementSystem/index.jsp
Home
Admin
Services
Contact Us
WellnessCenter
USER LOGIN
Username
password
LOG IN
Create Account


WellnessCenter.com
localhost:8080/HospitalManagementSystem/userRegister.jsp
Home
Admin
Services
Contact Us
WellnessCenter
USER LOGIN
Enter Username
Enter Password
Enter Password Again
REGISTER
Log in

UserHomePage



Appointment Management

Once your patient has booked an appointment, the HMS software for hospitals will match the patient's illness to the doctor's area of expertise. It will then assign them to the next available specialist or the one they prefer. It also updates the available slots in real-time to avoid any confusion at the

Patient Management

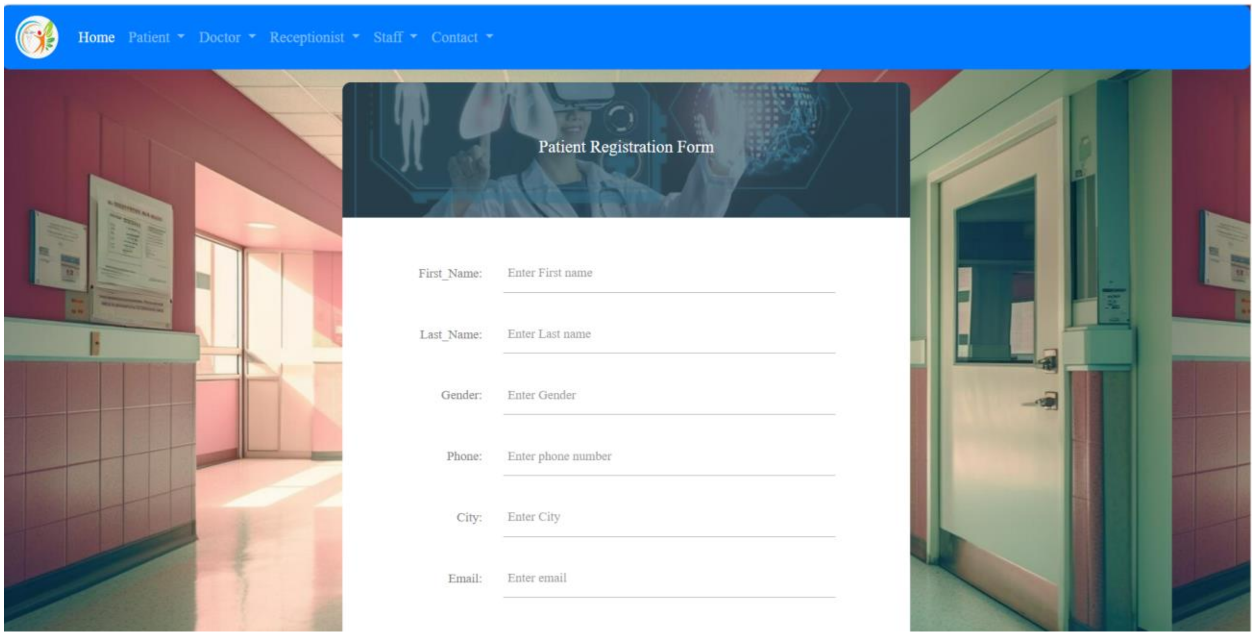
After the patient onboarding is completed, the patient is moved to an IPD or OPD. The patient management module of HMS caters to the needs of the inpatient and outpatient departments. It captures and stores the medical history, treatment required, details of their previous visits, upcoming appointments, reports,

Staff Management

The staff management module provides a concrete solution for the HR department. It contains records of your staff, job description, service domain, and other vital details. It helps you to know your staff without going through a heavy bundle of files. Additionally, it enables you to plan the hiring process based on the requirements of

Facility Management

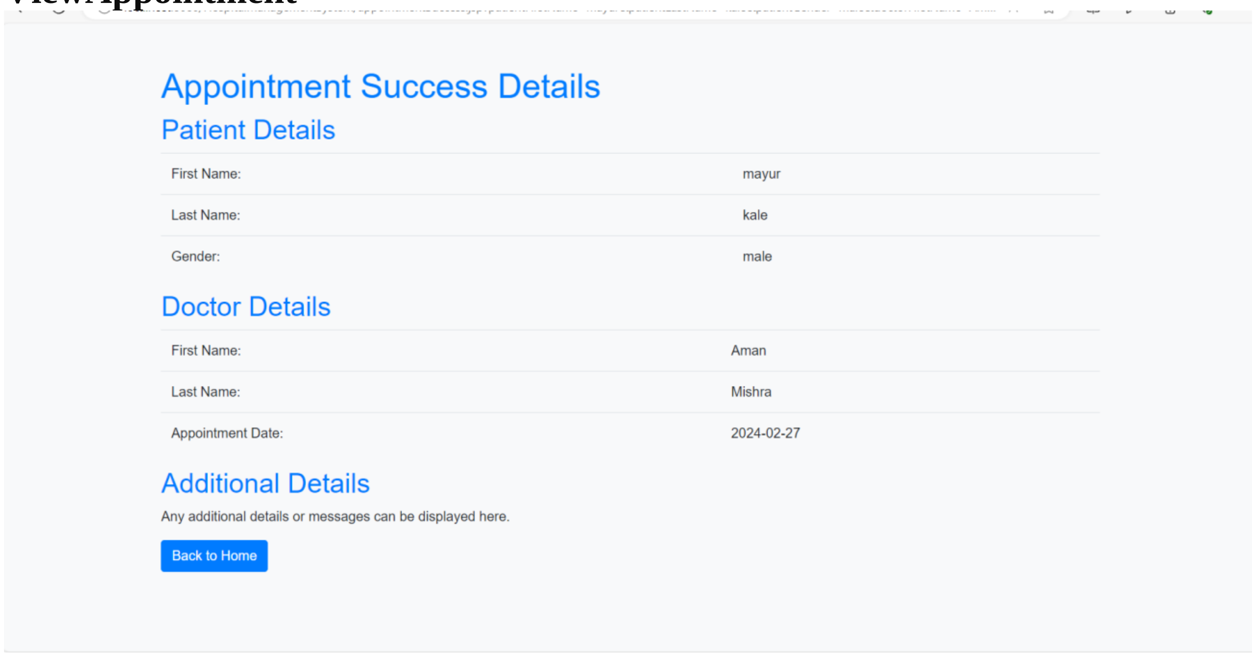
To provide a smooth experience for your patients, it is essential for your staff to have easy access to necessary hospital records. The facility management module of a healthcare management system helps you to maintain records of bed availability, occupancy status of rooms with specialized care, and more.



Patient Registration Form

First_Name:	Enter First name
Last_Name:	Enter Last name
Gender:	Enter Gender
Phone:	Enter phone number
City:	Enter City
Email:	Enter email

ViewAppointment



Appointment Success Details

Patient Details

First Name:	mayur
Last Name:	kale
Gender:	male

Doctor Details

First Name:	Aman
Last Name:	Mishra
Appointment Date:	2024-02-27

Additional Details

Any additional details or messages can be displayed here.

Chapter – 7 (Coding/Implementation)

7.1 CODING

DatabaseConnection.java

```
package Database;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
  // This class can be used to initialize the database connection
public class DatabaseConnection {
    public static Connection initializeDatabase()
        throws SQLException, ClassNotFoundException
    {
         String dbDriver = "com.mysql.jdbc.Driver";
        String dbURL = "jdbc:mysql://localhost:3306/";
        // Database name to access
        String dbName = "hospital";
        String dbUsername = "root";
        String dbPassword = "root";

        Class.forName(dbDriver);
        Connection con =
DriverManager.getConnection(dbURL+dbName,dbUsername,dbPassword);
        return con;
    }
}
```

Home.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>WellnessCenter.com </title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet" href="css/style1.css" type="text/css" />
        <script src="js/script.js"></script>
    </head>
    <body>
        <header>
            <nav>
                <ul>
                    <li><a href="#">Home</a></li>
                    <li><a href="#" id="services-link">Services</a></li>
                    <li><a href="#" id="doctors-link">Doctors</a></li>
                    <li><a href="#">About Us</a></li>
                    <li><a href="contact.jsp">Contact</a></li>
                </ul></nav></header><main>
            <section class="hero">
                <div class="img-hero">
                    <h1>Welcome to Our WellnessCenter</h1>
                    <h2>तुमचे आरोग्य, आमचे प्राधान्य</h2>
                    <p>Providing compassionate care for all our patients.</p>
                    <a href="index.jsp" class="btn">Book Appointment</a>
                </div>
            </section>
      <section class="services">
                <h2>Our Services</h2>
                <div class="service">
                    <img src="img/service1.jpg" alt="Service 1">
                    <h3>Emergency Care</h3>
                    <p>24/7 emergency medical services</p>
                </div>
                <div class="service">
                    <img src="img/service2.jpg" alt="Service 2">
                    <h3>Specialized Treatments</h3>
                    <p>Advanced treatments for complex conditions</p>
                </div>
                <div class="service">
                    <img src="img/service3.jpg" alt="Service 3">
                    <h3>Primary Care</h3>
                    <p>Comprehensive primary healthcare services</p></div>
            </section></main>
        <section class="doctors">
            <h2>आमचे डॉक्टर</h2>
            <div class="doctor-box">
                <img src="img/doctor1.jpg" alt="Doctor 1">
                <h3>डॉ अमन मिश्रा</h3>
                <p>Specialty: Cardiology</p>
            </div>
            <div class="doctor-box">
                <img src="img/doctor2.jpg" alt="Doctor 2">
                <h3>डॉ सनाथ शेट्टी</h3>
                <p>Specialty: Plastic Surgeons</p>
            </div>
            <div class="doctor-box">
                <img src="img/doctor3.jpg" alt="Doctor 3">
                <h3>डॉ मयूर काळे</h3>
                <p>Specialty: General Surgery</p>
            </div>
        </section>
        <footer>
            <div class="container">
                <div class="footer-links">
                    <ul>
                        <li><a href="index.html">Home</a></li>
                        <li><a href="index.htmll">Services</a></li>
                        <li><a href="index.html">About Us</a></li>
                        <li><a href="index.html">Contact</a></li>
                    </ul>
                </div>
                <div class="footer-contact">
                    <p>Contact Us:</p>
                    <p>Hospital St, Mumbai, India</p>
                    <p>Email: info@ourhospital.com</p>
                    <p>Phone: +1234567890</p>
                </div>
            </div>
            <p>&copy; 2024 Our Hospital. All rights reserved.</p>
        </footer>
    </body>
</html>
```

Adminlogin.jsp

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="ISO-8859-1">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet" href="fonts/material-icon/css/material-design-iconic-font.min.css">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" href="css/register.css" type="text/css" />
    </head>
    <body>
         <nav class="navbar navbar-expand-md navbar-light bg-light">
            <div class="container">

                <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav ml-auto">
                        <li class="nav-item">
                            <a class="nav-link" href="index.jsp">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="adminLogin.jsp">Admin</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="index.html" id="services-link">Services</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="contact.jsp">Contact Us</a>
                        </li>  </ul></div></div></nav>  <div>
            <h1>
                <b>WellnessCenter</b>
            </h1>
        </div>
        <!-- Sing in  Form -->
        <form action="<%=request.getContextPath()%>/AdminLogin" method="post">
            <section class="sign-in">
                <div class="container">
                    <div class="signin-content">
                        <div class="signin-image" >
                            <figure><img src="img/adminlogin.png" alt="sing up image" > </figure>
                            <a href="adminRegister.jsp" class="signup-image-link">Create an account</a>
                        </div>

                        <div class="signin-form">
                            <h2 class="form-title">Sign in</h2>
                            <form method="POST" class="register-form" id="login-form">
                                <div class="form-group">
                                <label for="your_name"><i class="zmdi zmdi-account material-icons-
name"></i></label>
                                <input type="text" name="your_name" id="your_name"
placeholder="User Name"/>
                            </div>
                            <div class="form-group">
                                <label for="your_pass"><i class="zmdi zmdi-lock"></i></label>
                                <input type="password" name="your_pass" id="your_pass"
placeholder="Password"/>
                            </div>
                            <div class="form-group">
                                <input type="checkbox" name="remember-me" id="remember-me"
class="agree-term" />
                                <label for="remember-me" class="label-agree-
term"><span><span></span></span>Remember me</label>
                            </div>
                            <div class="form-group form-button">
                                <input type="submit" name="signin" id="signin" class="form-submit"
value="Log in"/>
                            </div>  </form></div></div></div></section>
        </form>
            <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
            <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
            <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
    </body>
</html>
```

Adminregister.jsp

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="ISO-8859-1">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet" href="fonts/material-icon/css/material-design-iconic-font.min.css">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" href="css/register.css" type="text/css" />
    </head>
    <body>
        <nav class="navbar navbar-expand-md navbar-light bg-light">
            <div class="container">

                <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav ml-auto">
                        <li class="nav-item">
                            <a class="nav-link" href="index.jsp">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="adminLogin.jsp">Admin</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="index.html" id="services-link">Services</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="contact.jsp">Contact Us</a>
                        </li>
                    </ul>
                </div></div>    </nav> <div>
            <h1>
                <b>WellnessCenter</b>
            </h1>
        </div>
        <form action="<%=request.getContextPath()%>/AdminRegister" method="post">
            <!-- Sign up form -->
            <section class="signup">
                <div class="container">
                    <div class="signup-content">
                        <div class="signup-form">
```

```
                              <h2 class="form-title">Sign up</h2>
                              <form method="POST" class="register-form" id="register-form">
                                  <div class="form-group">
                                      <label for="email"><i class="zmdi zmdi-email"></i></label>
                                      <input type="email" name="email" id="email" placeholder="Your
Email"/>
                                  </div>
                                  <div class="form-group">
                                      <label for="pass"><i class="zmdi zmdi-lock"></i></label>
                                      <input type="password" name="pass" id="pass"
placeholder="Password"/>
                                  </div>
                                  <div class="form-group">
                                      <label for="re-pass"><i class="zmdi zmdi-lock-outline"></i></label>
                                      <input type="password" name="re_pass" id="re_pass"
placeholder="Repeat your password"/>
                                  </div>
                                  <div class="form-group">
                                      <input type="checkbox" name="agree-term" id="agree-term"
class="agree-term" />
                                      <label for="agree-term" class="label-agree-
term"><span><span></span></span>I agree all statements in  <a href="#" class="term-
service">Terms of service</a></label>
                                  </div>
                                  <div class="form-group form-button">
                                      <input type="submit" name="signup" id="signup" class="form-submit"
value="Register"/>
                                  </div>
                              </form>
                          </div>
                          <div class="signup-image">
                              <figure><img src="img/admin1.jpg" alt="sing up image"></figure>
                              <a href="adminLogin.jsp" class="signup-image-link">I am already
member</a>
                          </div>
                      </div>
                  </div>
              </section>
          </form>
              <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
              <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
              <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

      </body>
```

```
</html>
```

AdminLogin.java

```
package Controller;
import Database.DatabaseConnection;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/AdminLogin")
public class AdminLogin extends HttpServlet {
    private String user;
    private String pass;

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        PrintWriter pw = response.getWriter();
        try {
            String userp = request.getParameter("your_name");
            String passp = request.getParameter("your_pass");
            Connection con = DatabaseConnection.initializeDatabase();

            String s = "select *from adminreg";
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery(s);
            while (rs.next()) {
                user = rs.getString(1);
                pass = rs.getString(2);
            }
            if (userp.equals(user) && passp.equals(pass)) {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Login Successfully..!');");
                pw.println("window.location.href = \"AdminHome.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("AdminHome.jsp");
                //rd.forward(request, response);
            } else {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Username or Password is Incorrect..!');");
                pw.println("window.location.href = \"adminLogin.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
                //rd.include(request, response);
            }
        } catch (Exception e) {}}}}
```

AdminHome.jsp

```
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="Database.DatabaseConnection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" href="css/nav.css" type="text/css" />
    </head>
    <body>
        <%
            Connection con = null;
        %>
        <nav class="navbar navbar-expand-lg navbar-dark bg-primary" >
            <a class="navbar-brand" href="index.html">
                <img src="img/logo1.png" height="50" width="50"
alt="HospitalManagementSystem">
```

```
            </a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
                    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="index.jsp">Home <span class="sr-
only">(current)</span></a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="addpatient.jsp"
id="patientDropdown" role="button"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Patient
                        </a>
                        <div class="dropdown-menu" aria-labelledby="patientDropdown">
                            <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                            <a class="dropdown-item" href="adminPatientList.jsp">Patient List</a>
                            <a class="dropdown-item" href="appointmentSuccess.jsp">Patient List</a>
                        </div>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="addDoctor.jsp"
id="doctorDropdown" role="button"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Doctor
                        </a>
                        <div class="dropdown-menu" aria-labelledby="doctorDropdown">
                            <a class="dropdown-item" href="addDoctor.jsp">Add Doctor</a>
                            <a class="dropdown-item" href="adminDoctorList.jsp">View Doctor</a>
                        </div>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="addRecp.jsp"
id="receptionistDropdown" role="button"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Receptionist
                        </a>
                        <div class="dropdown-menu" aria-labelledby="receptionistDropdown">
                            <a class="dropdown-item" href="addRecp.jsp">Add Receptionist</a>
                            <a class="dropdown-item" href="adminRecpList.jsp">View Receptionist</a>
                        </div>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="addStaff.jsp" id="StaffDropdown"
role="button"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Staff
                        </a>
                        <div class="dropdown-menu" aria-labelledby="StaffDropdown">
                            <a class="dropdown-item" href="addStaff.jsp">Add Staff</a>
                            <a class="dropdown-item" href="adminStaffList.jsp">View Staff</a>
                        </div>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="admincontactview.jsp"
id="StaffDropdown" role="button"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Contact
                        </a>
                        <div class="dropdown-menu" aria-labelledby="StaffDropdown">

                            <a class="dropdown-item" href="admincontactview.jsp">View contact</a>
                        </div></li></ul></div></nav><%
            try {
                con = DatabaseConnection.initializeDatabase();
                Statement st = (Statement) con.createStatement();
                String query = "select count(*) from patient";
                ResultSet rs = st.executeQuery(query);
                while (rs.next()) {
                    int patient = rs.getInt(1);
        %><%
                }con.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        %>
        <%
            try {
                con = DatabaseConnection.initializeDatabase();
                Statement st = (Statement) con.createStatement();
                String query = "select count(*) from doctor";
                ResultSet rs = st.executeQuery(query);
                while (rs.next()) {
                    int doctor = rs.getInt(1);%>
        <%}
                con.close();
```

```
    } catch (Exception e) {
        e.printStackTrace();
    }
%>
<%
    try {
        con = DatabaseConnection.initializeDatabase();
        Statement st = (Statement) con.createStatement();
        String query = "select count(*) from recp";
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            int recp = rs.getInt(1);
%>
<%
        }
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
%>
<%
    try {
        con = DatabaseConnection.initializeDatabase();
        Statement st = (Statement) con.createStatement();
        String query = "select count(*) from Staff";
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            int Staff = rs.getInt(1);
%>
<%
        }
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
%>
<%
    try {
        con = DatabaseConnection.initializeDatabase();
        Statement st = (Statement) con.createStatement();
        // Fetch details for Patient
        String patientQuery = "select count(*) from patient";
        ResultSet patientRs = st.executeQuery(patientQuery);
        int patientCount = 0;
        if (patientRs.next()) {
            patientCount = patientRs.getInt(1);
        }   // Fetch details for Doctor
        String doctorQuery = "select count(*) from doctor";
        ResultSet doctorRs = st.executeQuery(doctorQuery);
        int doctorCount = 0;
        if (doctorRs.next()) {
            doctorCount = doctorRs.getInt(1);
        }
        // Fetch details for Receptionist
        String receptionistQuery = "select count(*) from recp";
        ResultSet receptionistRs = st.executeQuery(receptionistQuery);
        int receptionistCount = 0;
        if (receptionistRs.next()) {
            receptionistCount = receptionistRs.getInt(1);
        }
        // Fetch details for Staff
        String StaffQuery = "select count(*) from Staff";
        ResultSet StaffRs = st.executeQuery(StaffQuery);
        int StaffCount = 0;
        if (StaffRs.next()) {
            StaffCount = StaffRs.getInt(1);
        }
 %>
<div class="container mt-5">
    <div class="row">
        <!-- Box 1: Patient -->
        <div class="col-md-3">
            <div class="card">
                <a href="adminPatientList.jsp">
                    <img class="card-img-top" src="img/patient.png" alt="Patient Image">
                </a>
                <div class="card-body">
                    <h5 class="card-title">Patient</h5>
                    <p class="card-text">Total Patients: <%= patientCount%></p>
                    <p class="card-text">Additional information about patients...</p>
                </div></div></div>
        <!-- Box 2: Doctor -->
        <div class="col-md-3">
            <div class="card">
                <a href="adminDoctorList.jsp">
                    <img class="card-img-top" src="img/doctor1.png" alt="doctor Image">
                </a>
                <div class="card-body">
                    <h5 class="card-title">Doctor</h5>
                    <p class="card-text">Total Doctors: <%= doctorCount%></p>
                    <p class="card-text">Additional information about doctors...</p>
                </div></div></div>
                <!-- Box 3: Receptionist -->
                <div class="col-md-3">
                    <div class="card">
                        <a href="adminRecpList.jsp">
                            <img class="card-img-top" src="img/receptionist.png" alt="Receptionists
Image">
                        </a>
                        <div class="card-body">
                            <h5 class="card-title">Receptionist</h5>
                            <p class="card-text">Total Receptionists: <%= receptionistCount%></p>
                            <p class="card-text">Additional information about receptionists...</p>
                        </div></div></div>
                <!-- Box 4: Staff -->
                <div class="col-md-3">
                    <div class="card">
                        <a href="adminStaffList.jsp">
                            <img class="card-img-top" src="img/worker.png" alt="Staff Image">
                        </a>
                        <div class="card-body">
                            <h5 class="card-title">Staff</h5>
                            <p class="card-text">Total Staffs: <%= StaffCount%></p>
                            <p class="card-text">Additional information about Staffs...</p>
                        </div>   </div></div></div>
            <%
                con.close();
            } catch (Exception e) {
                e.printStackTrace();
            }           %>
    </body>
</html>
```

AddDoctor.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" type="text/css" href="css/adddataform.css">
        <link rel="stylesheet" type="text/css" href="css/adddatafrm1.css">
         <link rel="stylesheet" href="css/nav.css" type="text/css" />
        <style>
            body {
                background-image: url("img/Medical.jpg");
                background-color: #cccccc;
            }
        </style>
    </head>
    <body>
        <nav class="navbar navbar-expand-lg navbar-dark bg-primary" >
    <a class="navbar-brand" href="index.html">
        <img src="img/logo1.png" height="50" width="50" alt="HospitalManagementSystem">
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="AdminHome.jsp">Home <span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addpatient.jsp" id="patientDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Patient
                </a>
                <div class="dropdown-menu" aria-labelledby="patientDropdown">
```

```
                    <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                    <a class="dropdown-item" href="adminPatientList.jsp">Patient List</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addDoctor.jsp" id="doctorDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Doctor
                </a>
                <div class="dropdown-menu" aria-labelledby="doctorDropdown">
                    <a class="dropdown-item" href="addDoctor.jsp">Add Doctor</a>
                    <a class="dropdown-item" href="adminDoctorList.jsp">View Doctor</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addRecp.jsp" id="receptionistDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Receptionist
                </a>
                <div class="dropdown-menu" aria-labelledby="receptionistDropdown">
                    <a class="dropdown-item" href="addRecp.jsp">Add Receptionist</a>
                    <a class="dropdown-item" href="adminRecpList.jsp">View Receptionist</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addStaff.jsp" id="StaffDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Staff
                </a>
                <div class="dropdown-menu" aria-labelledby="StaffDropdown">
                    <a class="dropdown-item" href="addStaff.jsp">Add Staff</a>
                    <a class="dropdown-item" href="adminStaffList.jsp">View Staff</a>
                </div>
            </li>
             <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="admincontactview.jsp"
id="StaffDropdown" role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Contact
                </a>
                <div class="dropdown-menu" aria-labelledby="StaffDropdown">

                    <a class="dropdown-item" href="admincontactview.jsp">View contact</a>
                </div></li></ul></div>
</nav>
        <div class="container-contact100">
            <div class="wrap-contact100">
                <div class="contact100-form-title" style="background-image: url(img/bg-01.jpg);">
                    <span class="contact100-form-title-1">
                        Doctor Registration Form
                    </span>
                </div>
                <form class="contact100-form validate-form"
action="<%=request.getContextPath()%>/AddDoctor" method="post">
                    <div class="wrap-input100 validate-input" data-validate="Uniqe ID is required">
                        <span class="label-input100">Id</span>
                        <input class="input100" type="text" name="id" placeholder="Enter Uniqe ID">
                        <span class="focus-input100"></span>
                    </div>
                    <div class="wrap-input100 validate-input" data-validate="First Name is required">
                        <span class="label-input100">First_Name:</span>
                        <input class="input100" type="text" name="fname" placeholder="Enter First
name">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Last Name is required">
                        <span class="label-input100">Last_Name:</span>
                        <input class="input100" type="text" name="lname" placeholder="Enter Last
name">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate = "gender is required">
                        <span class="label-input100">Gender:</span>
                        <input class="input100" type="text" name="gender" placeholder="Enter
Gender">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Phone is required">
                        <span class="label-input100">Phone:</span>
                        <input class="input100" type="text" name="Mobile" placeholder="Enter phone
number">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="City is required">
                        <span class="label-input100">City:</span>
```

```
          <input class="input100" type="text" name="City" placeholder="Enter City">
          <span class="focus-input100"></span>
        </div>

        <div class="wrap-input100 validate-input" data-validate="Valid email is required:
ex@abc.xyz">
          <span class="label-input100">Email:</span>
          <input class="input100" type="text" name="email" placeholder="Enter email">
          <span class="focus-input100"></span>
        </div>

        <div class="wrap-input100 validate-input" data-validate="Age is required">
          <span class="label-input100">Age:</span>
          <input class="input100" type="text" name="age" placeholder="Enter Age">
          <span class="focus-input100"></span>
        </div>

        <div class="wrap-input100 validate-input" data-validate="Address is required">
          <span class="label-input100">Address</span>
          <input class="input100" type="text" name="address" placeholder="Enter
Address">
          <span class="focus-input100"></span>
        </div>

        <div class="wrap-input100 validate-input" data-validate="qualification is
required">
          <span class="label-input100">Qualification</span>
          <input class="input100" type="text" name="qualification" placeholder="Enter
Qualification">
          <span class="focus-input100"></span>
        </div>

        <div class="container-contact100-form-btn">
          <button class="contact100-form-btn">
            <span>
              Submit
              <i class="fa fa-long-arrow-right m-l-7" aria-hidden="true"></i></span>
          </button></div></form></div></div>
    <div id="dropDownSelect1"></div>
    <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAKFWBqlKAGCeS1rMVoaNlwyay
u0e0YRes"></script>
    <script src="js/map-custom.js"></script>
    <script src="js/main.js"></script>

    <!-- Global site tag (gtag.js) - Google Analytics -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=UA-23581568-
13"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag() {
        dataLayer.push(arguments);
      }
      gtag('js', new Date());
      gtag('config', 'UA-23581568-13');
    </script>  </body></html>
```

AddDoctor.java

```
package Controller;
import Database.DatabaseConnection;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/AddDoctor")
public class AddDoctor extends HttpServlet {

  private int i;

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    PrintWriter pw = response.getWriter();
    try {
      Date todaysDate = new Date();
      DateFormat df2 = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");

      String sid = request.getParameter("id");
      int id = Integer.parseInt(sid);
```

```
            String fname = request.getParameter("fname");
            String lname = request.getParameter("lname");
            String gender = request.getParameter("gender");
            String phone = request.getParameter("Mobile");
            String city = request.getParameter("City");
            String email = request.getParameter("email");
            String age = request.getParameter("age");
            String address = request.getParameter("address");
            String qualification = request.getParameter("qualification");

            String DateAndTime = df2.format(todaysDate);

            Connection con = DatabaseConnection.initializeDatabase();
            PreparedStatement pst = con.prepareStatement("insert into doctor
values(?,?,?,?,?,?,?,?,?,?,?)");
            pst.setInt(1, id);
            pst.setString(5, phone);
            pst.setString(2, fname);
            pst.setString(3, lname);
            pst.setString(4, gender);
            pst.setString(6, city);
            pst.setString(7, email);
            pst.setString(8, age);
            pst.setString(9, address);
            pst.setString(10, DateAndTime);
            pst.setString(11, qualification);

            i = pst.executeUpdate();
            if (i > 0) {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Data Add Successfully..!');");
                pw.println("window.location.href = \"AdminHome.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("AdminHome.jsp");
                //rd.forward(request, response);
            } else {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Failed !!!!,try Again Later!');");
                pw.println("window.location.href = \"addDoctor.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("addDoctor.jsp");
                //rd.forward(request, response);
            }
        } catch (SQLException | ClassNotFoundException ex) {
            Logger.getLogger(AddPatient.class.getName()).log(Level.SEVERE, null, ex);
        }
```

```
    }

}
```

AddPatient.jsp

```
<div class="container-contact100">


        <div class="wrap-contact100">
            <div class="contact100-form-title" style="background-image: url(img/bg-01.jpg);">
                <span class="contact100-form-title-1">
                    Patient Registration Form
                </span>
            </div>

            <form class="contact100-form validate-form"
action="<%=request.getContextPath()%>/AddPatient" method="post">
                <div class="wrap-input100 validate-input" data-validate="First Name is required">
                    <span class="label-input100">First_Name:</span>
                    <input class="input100" type="text" name="fname" placeholder="Enter First
name">
                    <span class="focus-input100"></span>
                </div>
                <div class="wrap-input100 validate-input" data-validate="Last Name is required">
                    <span class="label-input100">Last_Name:</span>
                    <input class="input100" type="text" name="lname" placeholder="Enter Last
name">
                    <span class="focus-input100"></span>
                </div>
                <div class="wrap-input100 validate-input" data-validate = "gender is required">
                    <span class="label-input100">Gender:</span>
                    <input class="input100" type="text" name="gender" placeholder="Enter
Gender">
                    <span class="focus-input100"></span>
                </div>
                <div class="wrap-input100 validate-input" data-validate="Phone is required">
                    <span class="label-input100">Phone:</span>
                    <input class="input100" type="text" name="Mobile" placeholder="Enter phone
number">
                    <span class="focus-input100"></span>
                </div>
                <div class="wrap-input100 validate-input" data-validate="City is required">
                    <span class="label-input100">City:</span>
                    <input class="input100" type="text" name="City" placeholder="Enter City">
                    <span class="focus-input100"></span>
                </div>
```

```
                    <div class="wrap-input100 validate-input" data-validate="Valid email is required: 
ex@abc.xyz">
                        <span class="label-input100">Email:</span>
                        <input class="input100" type="text" name="email" placeholder="Enter email">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Age is required">
                        <span class="label-input100">Age:</span>
                        <input class="input100" type="text" name="age" placeholder="Enter Age">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Address is required">
                        <span class="label-input100">Address</span>
                        <input class="input100" type="text" name="address" placeholder="Enter 
Address">
                        <span class="focus-input100"></span></div>
                    <div class="container-contact100-form-btn">
                        <button class="contact100-form-btn">
                            <span>Submit<i class="fa fa-long-arrow-right m-l-7" aria-hidden="true"></i>
                            </span>  </button>   </div></form></div>  </div>
                        <div id="dropDownSelect1"></div>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAKFWBqlKAGCeS1rMVoaNlwyay
u0e0YRes"></script>
        <script src="js/map-custom.js"></script>
            <script src="js/main.js"></script>
                        <!-- Global site tag (gtag.js) - Google Analytics -->
        <script async src="https://www.googletagmanager.com/gtag/js?id=UA-23581568-
13"></script>
        <script>
            window.dataLayer = window.dataLayer || [];
            function gtag() {
                dataLayer.push(arguments);
            }
            gtag('js', new Date());
            gtag('config', 'UA-23581568-13');
        </script></body></html>
```

AddPatient.java

```
package Controller;

import Database.DatabaseConnection;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;

@WebServlet("/AddPatient")
public class AddPatient extends HttpServlet {

    private int i;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        PrintWriter pw = response.getWriter();
        try {
            Date todaysDate = new Date();
            DateFormat df2 = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
            String fname = request.getParameter("fname");
            String lname = request.getParameter("lname");
            String gender = request.getParameter("gender");
            String phone = request.getParameter("Mobile");
            String city = request.getParameter("City");
            String email = request.getParameter("email");
            String age = request.getParameter("age");
            String address = request.getParameter("address");

            String DateAndTime = df2.format(todaysDate);

            Connection con = DatabaseConnection.initializeDatabase();
            PreparedStatement pst = con.prepareStatement("insert into patient
values(?,?,?,?,?,?,?,?,?)");
            pst.setString(9, phone);
            pst.setString(1, fname);
            pst.setString(2, lname);
            pst.setString(3, gender);
            pst.setString(4, city);
            pst.setString(5, email);
            pst.setString(6, age);
            pst.setString(7, address);
            pst.setString(8, DateAndTime);

            i = pst.executeUpdate();
            if (i > 0) {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Data Added Successfully..!');");
                pw.println("window.location.href = \"UserHome.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("UserHome.jsp");
                //rd.forward(request, response);
            } else {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Incorrect Data..!');");
                pw.println("window.location.href = \"addpatient.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("addpatient.jsp");
                //rd.forward(request, response);
            }
        } catch (SQLException ex) {
            Logger.getLogger(AddPatient.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(AddPatient.class.getName()).log(Level.SEVERE, null, ex);
        }
```

AdminpatientList.jsp

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="Database.DatabaseConnection"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>WellnessCenter.com</title>
    <link rel="icon" type="image/x-icon" href="img/logo1.png">
    <link rel="stylesheet"
          href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <link rel="stylesheet"
          href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <script
            src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
    <script
            src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
    <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
    <script
            src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script
            src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <link rel="stylesheet" type="text/css" href="css/adddataform.css">
    <link rel="stylesheet" type="text/css" href="css/adddatafrm1.css">
    <link rel="stylesheet" type="text/css" href="css/table.css">
    <link rel="stylesheet" href="css/nav.css" type="text/css"/>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <a class="navbar-brand" href="index.html">
        <img src="img/logo1.png" height="50" width="50" alt="HospitalManagementSystem">
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="AdminHome.jsp">Home <span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addpatient.jsp" id="patientDropdown"
role="button"
                   data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Patient
                </a>
                <div class="dropdown-menu" aria-labelledby="patientDropdown">
                    <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                    <a class="dropdown-item" href="adminPatientList.jsp">Patient List</a>
                </div>
            </li>
            <li class="nav-item dropdown">
```

```
                    <a class="nav-link dropdown-toggle" href="addDoctor.jsp"
id="doctorDropdown" role="button"
                       data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        Doctor
                    </a>
                    <div class="dropdown-menu" aria-labelledby="doctorDropdown">
                        <a class="dropdown-item" href="addDoctor.jsp">Add Doctor</a>
                        <a class="dropdown-item" href="adminDoctorList.jsp">View Doctor</a>
                    </div>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="addRecp.jsp"
id="receptionistDropdown" role="button"
                       data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        Receptionist
                    </a>
                    <div class="dropdown-menu" aria-labelledby="receptionistDropdown">
                        <a class="dropdown-item" href="addRecp.jsp">Add Receptionist</a>
                        <a class="dropdown-item" href="adminRecpList.jsp">View Receptionist</a>
                    </div>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="addStaff.jsp" id="StaffDropdown"
role="button"
                       data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        Staff
                    </a>
                    <div class="dropdown-menu" aria-labelledby="StaffDropdown">
                        <a class="dropdown-item" href="addStaff.jsp">Add Staff</a>
                        <a class="dropdown-item" href="adminStaffList.jsp">View Staff</a>
                    </div>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="admincontactview.jsp"
id="StaffDropdown" role="button"
                       data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                        Contact
                    </a>
                    <div class="dropdown-menu" aria-labelledby="StaffDropdown">

                        <a class="dropdown-item" href="admincontactview.jsp">View contact</a>
                    </div>  </li></ul></div></nav><      Connection con = null;%>
<div class="serchbar">
    <form action=" " method="post">
        <input class="search" type="text" name="search" placeholder="Search Here..."/>
    </form>
```

```
</div>

<div class="container-table100">
    <div class="wrap-table100">
        <div class="table100 ver3 m-b-110">
            <div class="table100-head">
                <table>
                    <thead>
                    <tr class="row100 head">
                        <th class="cell100 column1">First Name</th>
                        <th class="cell100 column2">Last Name</th>
                        <th class="cell100 column3">Gender</th>
                        <th class="cell100 column4">City</th>
                        <th class="cell100 column5">Email</th>
                        <th class="cell100 column6">Age</th>
                        <th class="cell100 column7">Address</th>
                        <th class="cell100 column8">Date</th>
                        <th class="cell100 column9">Mobile</th>
                        <th class="cell100 column10">Action</th>
                    </tr></thead></table></div>
            <%
              try {
                    con = DatabaseConnection.initializeDatabase();
                    Statement st = (Statement) con.createStatement();
                    String sql = "";
                    String query = request.getParameter("search");
                    if (query != null) {
                        sql = "select * from patient where fname like '%" + query + "%' or lname like '%"
+ query + "%' ";
                    } else {
                        sql = "select * from patient";
                    }
                    ResultSet rs = st.executeQuery(sql);
                    while (rs.next()) {
                        String mob = rs.getString(9);
            %>
                        <div class="table100-body js-pscroll">
                            <table>
                                <tbody>
                                    <tr class="row100 body">
                                        <td class="cell100 column1"><%=rs.getString(1)%></td>
                                        <td class="cell100 column2"><%=rs.getString(2)%></td>
                                        <td class="cell100 column3"><%=rs.getString(3)%></td>
                                        <td class="cell100 column4"><%=rs.getString(4)%></td>
                                        <td class="cell100 column5"><%=rs.getString(5)%></td>
                                        <td class="cell100 column6"><%=rs.getString(6)%></td>
                                <td class="cell100 column7"><%=rs.getString(7)%></td>
                                <td class="cell100 column8"><%=rs.getString(8)%></td>
                                <td class="cell100 column9"><%=rs.getString(9)%></td>
                                <td class="cell100 column10">
                                    <a href="updatePatient.jsp?mob=<%=mob%>">Update</a>
                                    &nbsp;&nbsp;&nbsp;&nbsp;
                                    <a href="deletePatient.jsp?mob=<%=mob%>">Delete</a>
                                    &nbsp;&nbsp;&nbsp;&nbsp;
                                    <a href="approvePatient.jsp?mob=<%=mob%>">Create
Appointment</a></td>  </tr></tbody></table></div>   <%
                    }
                    con.close();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            %></div></div></div></body></html>
```

AddDoctorList.java

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="Database.DatabaseConnection"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" type="text/css" href="css/adddataform.css">
        <link rel="stylesheet" type="text/css" href="css/adddatafrm1.css">
        <link rel="stylesheet" type="text/css" href="css/table.css">
         <link rel="stylesheet" href="css/nav.css" type="text/css" />
    </head>
    <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary" >
    <a class="navbar-brand" href="index.html">
        <img src="img/logo1.png" height="50" width="50" alt="HospitalManagementSystem">
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="AdminHome.jsp">Home <span class="sr-
only">(current)</span></a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addpatient.jsp" id="patientDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Patient
                </a>
                <div class="dropdown-menu" aria-labelledby="patientDropdown">
                    <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                    <a class="dropdown-item" href="adminPatientList.jsp">Patient List</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addDoctor.jsp" id="doctorDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Doctor
                </a>
                <div class="dropdown-menu" aria-labelledby="doctorDropdown">
                    <a class="dropdown-item" href="addDoctor.jsp">Add Doctor</a>
                    <a class="dropdown-item" href="adminDoctorList.jsp">View Doctor</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addRecp.jsp" id="receptionistDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Receptionist
                </a>
                <div class="dropdown-menu" aria-labelledby="receptionistDropdown">
                    <a class="dropdown-item" href="addRecp.jsp">Add Receptionist</a>
                    <a class="dropdown-item" href="adminRecpList.jsp">View Receptionist</a>
                </div>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="addStaff.jsp" id="StaffDropdown"
role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Staff
                </a>
                <div class="dropdown-menu" aria-labelledby="StaffDropdown">
                    <a class="dropdown-item" href="addStaff.jsp">Add Staff</a>
                    <a class="dropdown-item" href="adminStaffList.jsp">View Staff</a>
                </div>
            </li>
             <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="admincontactview.jsp"
id="StaffDropdown" role="button"
                    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    Contact
                </a>
                <div class="dropdown-menu" aria-labelledby="StaffDropdown">

                    <a class="dropdown-item" href="admincontactview.jsp">View contact</a>
                </div></li>  </ul></div></nav>
        <div class="serchbar">
            <form action=" " method="post">
                <input class="search" type="text" name="search" placeholder="Search Here..."/>
            </form>
        </div>
        <%
            Connection con = null;
        %>
        <div>
            <div class="container-table100">
                <div class="wrap-table100">
                    <div class="table100 ver3 m-b-110">
                        <div class="table100-head">
                            <table>
                    <thead>
                        <tr class="row100 head">
                            <th class="cell100 column0">Id</th>
                            <th class="cell100 column1">First Name</th>
                            <th class="cell100 column2">Last Name</th>
                            <th class="cell100 column3">Gender</th>
                            <th class="cell100 column4">Mobile</th>
                            <th class="cell100 column5">City</th>
                            <th class="cell100 column6">Email</th>
                            <th class="cell100 column7">Age</th>
                            <th class="cell100 column8">Address</th>
                            <th class="cell100 column9">Date</th>
                            <th class="cell100 column10">Qualification</th>
                            <th class="cell100 column11">Action</th>
                        </tr> </thead></table></div>
            <%
                try {
                    con = DatabaseConnection.initializeDatabase();
                    Statement st = (Statement) con.createStatement();
                    String sql = "";
                    String query = request.getParameter("search");
                    if (query != null) {
                        sql = "select * from doctor where fname like '%" + query + "%' or lname
like '%" + query + "%' ";
                    } else {
                        sql = "select * from doctor";
                    }
                    ResultSet rs = st.executeQuery(sql);
                    while (rs.next()) {
                        String id = rs.getString(1);
            %>
            <div class="table100-body js-pscroll">
                <table>
                    <tbody>
                        <tr class="row100 body">
                            <td class="cell100 column0"><%=rs.getInt(1)%></td>
                            <td class="cell100 column1"><%=rs.getString(2)%></td>
                            <td class="cell100 column2"><%=rs.getString(3)%></td>
                            <td class="cell100 column3"><%=rs.getString(4)%></td>
                            <td class="cell100 column4"><%=rs.getString(5)%></td>
                            <td class="cell100 column5"><%=rs.getString(6)%></td>
                            <td class="cell100 column6"><%=rs.getString(7)%></td>
                            <td class="cell100 column7"><%=rs.getString(8)%></td>
                            <td class="cell100 column8"><%=rs.getString(9)%></td>
                            <td class="cell100 column9"><%=rs.getString(10)%></td>
                            <td class="cell100 column10"><%=rs.getString(11)%></td>
```

```
                              <td><a href="updateDoctor.jsp?id=<%= id%>">Update</a>
                                 &nbsp;&nbsp;&nbsp;&nbsp;
                                 <a href="deleteDoctor.jsp?id=<%=id%>">Delete</a>
                              </td>     </td></tr></tbody>
                        <%  } -con.close();} catch (Exception e) {
                              e.printStackTrace();
                           }
                        %>  </div></div></div></div></div>  </body></html>
```

deletePatient.jsp

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="Database.DatabaseConnection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>WellnessCenter.com</title>
      <link rel="icon" type="image/x-icon" href="img/logo1.png">
   </head>
   <body>
      <%
         String mob = request.getParameter("mob");
         Connection con = null;
         Statement stmt = null;
         con = DatabaseConnection.initializeDatabase();
         stmt = (Statement) con.createStatement();
         String query = "delete from  patient " + "where mobile = '"+mob+"'";
         stmt.executeUpdate(query);
         con.close();
         RequestDispatcher rd = request.getRequestDispatcher("AdminHome.jsp");
         rd.forward(request, response);
      %>
   </body>
</html>
```

DeleteDoctor.jsp

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="Database.DatabaseConnection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
```

```
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
    </head>
    <body>
        <%
            String id = request.getParameter("id");
            Connection con = null;
            Statement stmt = null;
            con = DatabaseConnection.initializeDatabase();
            stmt = (Statement) con.createStatement();
            String query = "delete from  doctor " + "where id = '"+id+"'";
            stmt.executeUpdate(query);
            con.close();
            RequestDispatcher rd = request.getRequestDispatcher("AdminHome.jsp");
            rd.forward(request, response);
        %></body></html>
```

updatePatient.jsp

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="Database.DatabaseConnection"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <link rel="stylesheet" type="text/css" href="css/adddataform.css">
        <link rel="stylesheet" type="text/css" href="css/adddatafrm1.css">
    </head>
    <body>
        <%
            String mob = request.getParameter("mob");
            Connection con = DatabaseConnection.initializeDatabase();
            String s = "SELECT * FROM patient WHERE mobile = '"+mob+"' ";
            PreparedStatement pstmt = con.prepareStatement(s);
            ResultSet  rs = pstmt.executeQuery();
            while (rs.next()) {       %>
        <div class="container-contact100">
            <div class="wrap-contact100">
                <div class="contact100-form-title" style="background-image: url(img/bg-01.jpg);">
                    <span class="contact100-form-title-1">
                        Patient Registration Form
                    </span>
                </div>

                <form class="contact100-form validate-form"
action="<%=request.getContextPath()%>/updatePatient" method="post">
                    <div class="wrap-input100 validate-input" data-validate="First Name is required">
                        <span class="label-input100">First_Name:</span>
                        <input class="input100" type="text" value="<%= rs.getString(1)%>"
name="fname" placeholder="Enter First name">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Last Name is required">
                        <span class="label-input100">Last_Name:</span>
                        <input class="input100" type="text"  value="<%= rs.getString(2)%>"
name="lname" placeholder="Enter Last name">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate = "gender is required">
                        <span class="label-input100">Gender:</span>
                        <input class="input100" type="text" value="<%= rs.getString(3)%>"
name="gender" placeholder="Enter Gender">
                        <span class="focus-input100"></span>
                    </div>
                    <div class="wrap-input100 validate-input" data-validate="Phone is required">
                        <span class="label-input100">Phone:</span>
                        <input class="input100" type="text" value="<%= rs.getString(9)%>"
name="Mobile" placeholder="Enter phone number">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="City is required">
                        <span class="label-input100">City:</span>
                        <input class="input100" type="text" value="<%= rs.getString(4)%>"
name="City" placeholder="Enter City">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Valid email is required:
ex@abc.xyz">
                        <span class="label-input100">Email:</span>
                        <input class="input100" type="text" value="<%= rs.getString(5)%>"
name="email" placeholder="Enter email">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Age is required">
                        <span class="label-input100">Age:</span>
                        <input class="input100" type="text" value="<%= rs.getString(6)%>" name="age"
placeholder="Enter Age">
                        <span class="focus-input100"></span>
                    </div>

                    <div class="wrap-input100 validate-input" data-validate="Address is required">
                        <span class="label-input100">Address</span>
                        <input class="input100" type="text" value="<%= rs.getString(7)%>"
name="address" placeholder="Enter Address">
                        <span class="focus-input100"></span>
                    </div>
                    <div class="container-contact100-form-btn">
                        <button class="contact100-form-btn">
                            <span>
                                Submit
                                <i class="fa fa-long-arrow-right m-l-7" aria-hidden="true"></i>
                            </span></button></div></form></div></div><%}  %></body></html>
```

Updatepatient.java

```
package Controller;
import Database.DatabaseConnection;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/updatePatient")
public class updatePatient extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        PrintWriter pw = response.getWriter();
        String fname = request.getParameter("fname");
        String lname = request.getParameter("lname");
        String gender = request.getParameter("gender");
        String phone = request.getParameter("Mobile");
        String city = request.getParameter("City");
        String email = request.getParameter("email");
        String age = request.getParameter("age");
        String address = request.getParameter("address");
     try {       Connection con = DatabaseConnection.initializeDatabase();
            PreparedStatement pst = con.prepareStatement("update patient set fname = ? , lname = ? ,
gender = ? , city = ? , email = ? , age = ? , address = ?  where mobile = '" + phone + "' ");
            pst.setString(1, fname);
            pst.setString(2, lname);
            pst.setString(3, gender);
            pst.setString(4, city);
            pst.setString(5, email);
            pst.setString(6, age);
            pst.setString(7, address);
            int i = pst.executeUpdate();
            if (i > 0) {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Update Successfully..!');");
                pw.println("window.location.href = \"AdminHome.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("AdminHome.jsp");
                //rd.forward(request, response);
            } else {
                pw.println("<script type=\"text/javascript\">");
                pw.println("alert('Failed..! Try Again Later...');");
                pw.println("window.location.href = \"updatePatient.jsp\";");
                pw.println("</script>");
                //RequestDispatcher rd = request.getRequestDispatcher("updatePatient.jsp");
                //rd.forward(request, response);
            }
            con.close();
        } catch (Exception e) {}    }}
```

UserSide:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>WellnessCentral.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
        <link rel="stylesheet"
              href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
        <link rel="stylesheet"
              href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
        <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <style>
        </style>
    </head>
    <body>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a href="#" class="navbar-brand">
                <img src="img/logo1.png" height="50" width="50"
alt="HospitalManagementSystem">
            </a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
```

```
            <span class="navbar-toggler-icon"></span>
          </button>

          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">
              <li class="nav-item active">
                <a class="nav-link" href="index.jsp">Home <span class="sr-
only">(current)</span></a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="contact.jsp">Contact</a>
              </li>
              <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                  Patient
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                  <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                  <a class="dropdown-item" href="listPatient.jsp">Patient List</a>
                  <div class="row mt-4">
                    <div class="col-md-6">
                      <a href="UserAppointmentDetails.jsp" class="btn btn-primary">Check
Appointment Details</a>
                    </div>
                    <div class="col-md-6">
                    </div>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="index.html" id="services-link">Services</a>
                    </li></ul></div></nav>
                  <div class="container mt-5">
                    <div class="row">
                      <div class="col-md-3">
                        <div class="card">
                          <img class="card-img-top" src="img/appoint.jpg" alt="Box 1">
                          <div class="card-body">
                            <h5 class="card-title">Appointment Management </h5>
                            <p class="card-text">Once your patient has booked an
appointment, the HMS software for hospitals will match the patient’s illness to the doctor’s area
of expertise. It will then assign them to the next available specialist or the one they prefer. It also
updates the available slots in real-time to avoid any confusion at the hospital.</p></div></div>
                      </div>
                      <div class="col-md-3">
                        <div class="card">
                          <img class="card-img-top" src="img/userpatient.jpg" alt="Box 2">
```

```
                                        <div class="card-body">
                                            <h5 class="card-title">Patient Management</h5>
                                            <p class="card-text">After the patient onboarding is completed,
the patient is moved to an IPD or OPD. The patient management module of HMS caters to the
needs of the inpatient and outpatient departments. It captures and stores the medical history,
treatment required, details of their previous visits, upcoming appointments, reports, insurance
details, and more.   </p></div></div></div><div class="col-md-3"><div class="card">
                                        <img class="card-img-top" src="img/usercard2.png" alt="Box 3">
                                        <div class="card-body">
                                            <h5 class="card-title">Staff Management</h5>
                                            <p class="card-text">The staff management module provides a
concrete solution for the HR department. It contains records of your staff, job description,
service domain, and other vital details. It helps you to know your staff without going through a
heavy bundle of files. Additionally, it enables you to plan the hiring process based on the
requirements of the hospital.  </p>  </div></div></div><div class="col-md-3"><div
class="card">
                                        <img class="card-img-top" src="img/usercard3.png" alt="Box 4">
                                        <div class="card-body">
                                            <h5 class="card-title">Facility Management</h5>
                                            <p class="card-text">           To provide a smooth experience for
your patients, it is essential for your staff to have easy access to necessary hospital records. The
facility management module of a healthcare management system helps you to maintain records
of bed availability, occupancy status of rooms with specialized care, and more.  </p>
                                        </div></div></div></div></div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
</body></html>
```

UserLogin.jsp

```
<!DOCTYPE html>
<html><head><meta charset="ISO-8859-1">
        <title>WellnessCenter.com</title>
        <link rel="icon" type="image/x-icon" href="img/logo1.png">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
```

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"rel="stylesheet"
id="bootstrap-css"><script
src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<link rel="stylesheet" href="css/style.css" type="text/css" />
<head><body>
        <nav class="navbar navbar-expand-md navbar-light bg-light">
            <div class="container">
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button><div class="collapse navbar-collapse" id="navbarNav">
                    <ul class="navbar-nav ml-auto">
                        <li class="nav-item">
                            <a class="nav-link" href="index.jsp">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="adminLogin.jsp">Admin</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="index.html" id="services-link">Services</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="contact.jsp">Contact Us</a>
                        </li></ul>   </div></div> <div>
            <h1>
                <b>WellnessCenter</b>
            </h1>
        </div>
        <div class="wrapper fadeInDown">
            <div id="formContent">
                <!-- Tabs Titles -->
                <h2>User Login</h2>
                <!-- Icon -->
                <div class="fadeIn first">
                </div>
                <!-- Login Form -->
                <form action="<%=request.getContextPath()%>/UserLogin" method="post">
                    <input type="text" id="Username" class="fadeIn second" name="username"
                           placeholder="Username">
                    <input type="password" id="password"
                           class="fadeIn third" name="password" placeholder="password">
                    <input
                        type="submit" class="fadeIn fourth" value="Log In">
                </form>
                <div id="formFooter">
                    <a class="underlineHover" href="userRegister.jsp">Create Account</a>
                </div>

            </div>
        </div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
    </body></html>
```

UserHome.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>WellnessCentral.com</title>
<link rel="icon" type="image/x-icon" href="img/logo1.png">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css"><script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script
        src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
        <script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link
            href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
            rel="stylesheet" id="bootstrap-css">
        <script
        src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
        <script
        src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <style>       </style></head><body>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a href="#" class="navbar-brand">
                <img src="img/logo1.png" height="50" width="50"
alt="HospitalManagementSystem">
            </a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="index.jsp">Home <span class="sr-
only">(current)</span></a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.jsp">Contact</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Patient
                        </a>
                        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                            <a class="dropdown-item" href="addpatient.jsp">Add Patient</a>
                            <a class="dropdown-item" href="listPatient.jsp">Patient List</a>
                            <div class="row mt-4">
                                <div class="col-md-6">
                                    <a href="UserAppointmentDetails.jsp" class="btn btn-primary">Check
Appointment Details</a>
                                </div>
                                <div class="col-md-6">
                                </div>
                                </li>
                                <li class="nav-item">
                                    <a class="nav-link" href="index.html" id="services-link">Services</a>
                                </li></ul></div></nav>
                            <div class="container mt-5">
                                <div class="row"><div class="col-md-3"><div class="card">
                                            <img class="card-img-top" src="img/appoint.jpg" alt="Box 1">
                                            <div class="card-body">
                                                <h5 class="card-title">Appointment Management </h5>
                                                <p class="card-text">Once your patient has booked an
appointment, the HMS software for hospitals will match the patient’s illness to the doctor’s area
of expertise. It will then assign them to the next available specialist or the one they prefer. It also
updates the available slots in real-time to avoid any confusion at the hospital.</p>
                                            </div></div></div>
                                    <div class="col-md-3">
                                        <div class="card">
                                            <img class="card-img-top" src="img/userpatient.jpg" alt="Box 2">
                                            <div class="card-body">
                                                <h5 class="card-title">Patient Management</h5>
                                        <p class="card-text">After the patient onboarding is completed,
the patient is moved to an IPD or OPD. The patient management module of HMS caters to the
needs of the inpatient and outpatient departments. It captures and stores the medical history,
treatment required, details of their previous visits, upcoming appointments, reports, insurance
details, and more.   </p></div></div></div>
                              <div class="col-md-3">
                                 <div class="card">
                                    <img class="card-img-top" src="img/usercard2.png" alt="Box 3">
                                    <div class="card-body">
                                       <h5 class="card-title">Staff Management</h5>
                                       <p class="card-text">The staff management module provides a
concrete solution for the HR department. It contains records of your staff, job description,
service domain, and other vital details. It helps you to know your staff without going through a
heavy bundle of files. Additionally, it enables you to plan the hiring process based on the
requirements of the hospital.  </p></div></div></div>
                              <div class="col-md-3">
                                 <div class="card">
                                    <img class="card-img-top" src="img/usercard3.png" alt="Box 4">
                                    <div class="card-body">
                                       <h5 class="card-title">Facility Management</h5>
                                       <p class="card-text">          To provide a smooth experience for
your patients, it is essential for your staff to have easy access to necessary hospital records. The
facility management module of a healthcare management system helps you to maintain records
of bed availability, occupancy status of rooms with specialized care, and more.  </p>
                                    </div></div></div></div></div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
                       </body></html>
```

8 TESTING

The testing phase is a critical aspect of ensuring the reliability, security, and functionality of the Hospital Management System. Through a comprehensive testing approach, the system undergoes rigorous evaluation to identify and rectify potential issues, validate functionalities, and fortify its overall performance.

In the realm of user management, meticulous test cases are executed to verify that new users can seamlessly register with valid credentials and that the system appropriately handles scenarios such as existing email addresses, ensuring a smooth onboarding experience.

For patient management functionalities, a battery of test cases is designed to validate the submission of accurate and error-free information. This includes testing patients' ability to submit personal and medical details without encountering issues and ensuring administrators can effectively manage and update patient records. Additionally, appointment scheduling features are thoroughly tested to confirm that users can create appointments accurately and administrators can efficiently manage the scheduling system.

Outputs from testing, such as successful registration confirmations, patient admission acknowledgments, and appointment scheduling confirmations, are meticulously examined. These outputs serve as indicators of the system's responsiveness and its ability to provide clear and reassuring feedback to users and administrators.

The testing approach adopts a combination of manual and automated methods. Manual testing delves into user interfaces, ensuring not only functionality but also usability and visual appeal. Automated testing focuses on critical functionalities, including authentication mechanisms and data validations, streamlining the testing process and facilitating efficient regression testing.

Unit testing involves scrutinizing individual components in isolation, validating their correctness and functionality. Integrated testing assesses the interaction between different components, ensuring seamless collaboration and integration. Security testing is paramount, with unauthorized access attempts and data encryption scrutinized to guarantee the system's robust defense against potential security threats.

By adopting this comprehensive testing strategy, the Hospital Management System aims to deliver a secure, reliable, and user-friendly solution for efficient hospital management. The commitment to thorough testing contributes to the system's stability, resilience, and ability to meet the dynamic needs of users and administrators.

8.3 Testing Approach

The testing approach for the Hospital Management System (HMS) adopts a multi-faceted strategy to ensure reliability, functionality, and security. It encompasses both manual and automated testing methodologies to comprehensively evaluate the system.

Manual Testing:

Manual testing is crucial for assessing the usability and user experience of the HMS. It involves executing test cases to validate various functionalities such as patient registration, appointment scheduling, medical record management, and billing processes.

Manual testers interact with the system's user interface to ensure it is intuitive, responsive, and error-free. They verify that users can navigate through different modules seamlessly and perform tasks efficiently.

Automated Testing:

Automated testing is employed to validate critical functionalities, especially those related to data processing, calculations, and system integrations. Automated test suites are developed using testing frameworks such as Selenium or JUnit to streamline the testing process.

Automated tests focus on areas such as database operations, API integrations, and complex business logic. They help identify potential issues early in the development cycle and facilitate continuous integration and delivery practices.

8.4 Unit Testing

Unit testing is performed to validate individual components or modules of the HMS in isolation. It ensures that each component functions correctly according to its specifications.

Examples of unit tests in the HMS include:

- Testing the patient registration module to verify that new patient records are created accurately.
- Testing the appointment scheduling module to ensure appointments are booked correctly based on availability.
- Testing the billing module to validate the calculation of medical fees and generation of invoices.

8.5 Integrated Testing

Integrated testing evaluates the interaction between different modules or subsystems of the HMS. It ensures that the system functions as intended when components are integrated together.

Examples of integrated tests in the HMS include:

- Testing the end-to-end patient journey from registration to discharge to ensure seamless transitions between modules.
- Testing the integration between the electronic health record (EHR) system and the laboratory management system to verify the flow of medical data.
- Testing the integration between the billing module and the payment gateway to ensure secure and accurate payment processing.

8.6 Security Testing

Security testing is paramount in the HMS to protect sensitive patient data and ensure compliance with healthcare regulations such as HIPAA (Health Insurance Portability and Accountability Act).

Examples of security tests in the HMS include:

- Testing the authentication and authorization mechanisms to ensure only authorized users can access patient records and sensitive information.
- Testing for vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF) to prevent unauthorized access and data breaches.
- Testing data encryption and secure transmission protocols to safeguard patient data during storage and transmission.

8.7 Performance Testing

Performance testing evaluates the responsiveness, scalability, and stability of the HMS under various load conditions.

Examples of performance tests in the HMS include:

- Load testing to assess the system's performance under expected user loads, ensuring it can handle concurrent user interactions without degradation in response time.
- Stress testing to push the system beyond its normal operational capacity and identify performance bottlenecks or failure points.
- Scalability testing to evaluate the system's ability to scale up or down in response to changing demands, ensuring it can accommodate growth without compromising performance.

8.8 Usability Testing

Usability testing focuses on the user interface design and overall user experience of the HMS. It ensures that the system is intuitive, easy to navigate, and meets the needs of its users.

Examples of usability tests in the HMS include:

- User interface evaluation to assess the layout, design, and accessibility of the HMS interface across different devices and screen sizes.

- User feedback analysis to gather input from end-users and stakeholders about their experience with the system and identify areas for improvement.
- Accessibility testing to ensure the HMS complies with accessibility standards and is usable by individuals with disabilities.

8.9 Compliance Testing

Compliance testing ensures that the HMS adheres to regulatory standards and industry best practices, particularly in the healthcare domain.

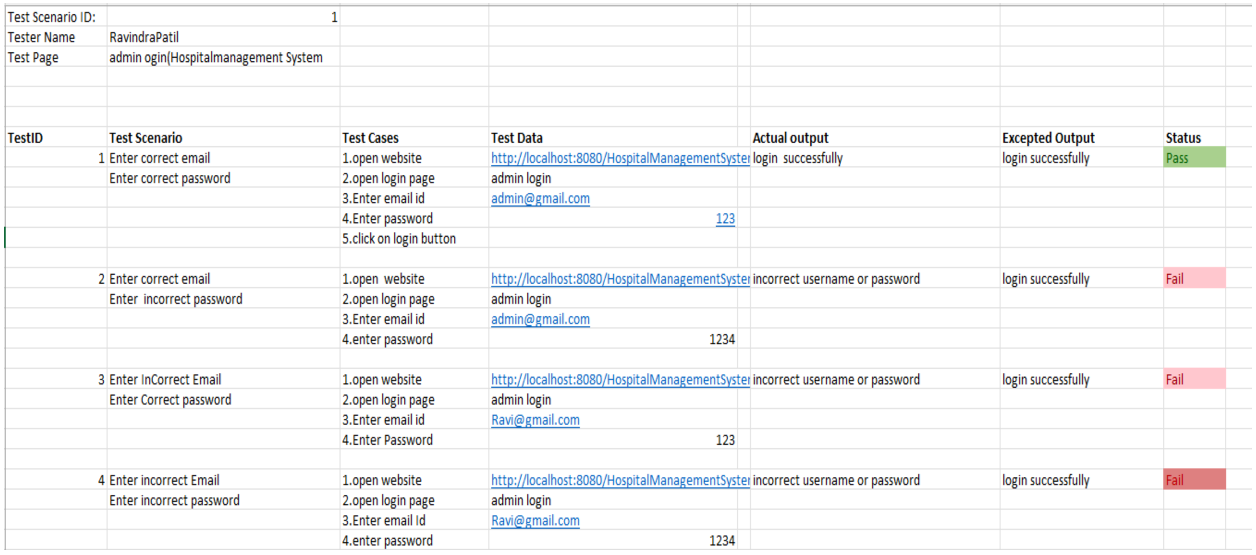
Examples of compliance tests in the HMS include:

- HIPAA compliance testing to verify that the system complies with regulations for protecting patient health information and ensuring confidentiality.
- GDPR compliance testing to ensure compliance with data protection and privacy regulations for European Union residents.
- Healthcare industry standards compliance testing to validate adherence to standards such as HL7 (Health Level Seven) for interoperability and data exchange.

8.10 Conclusion

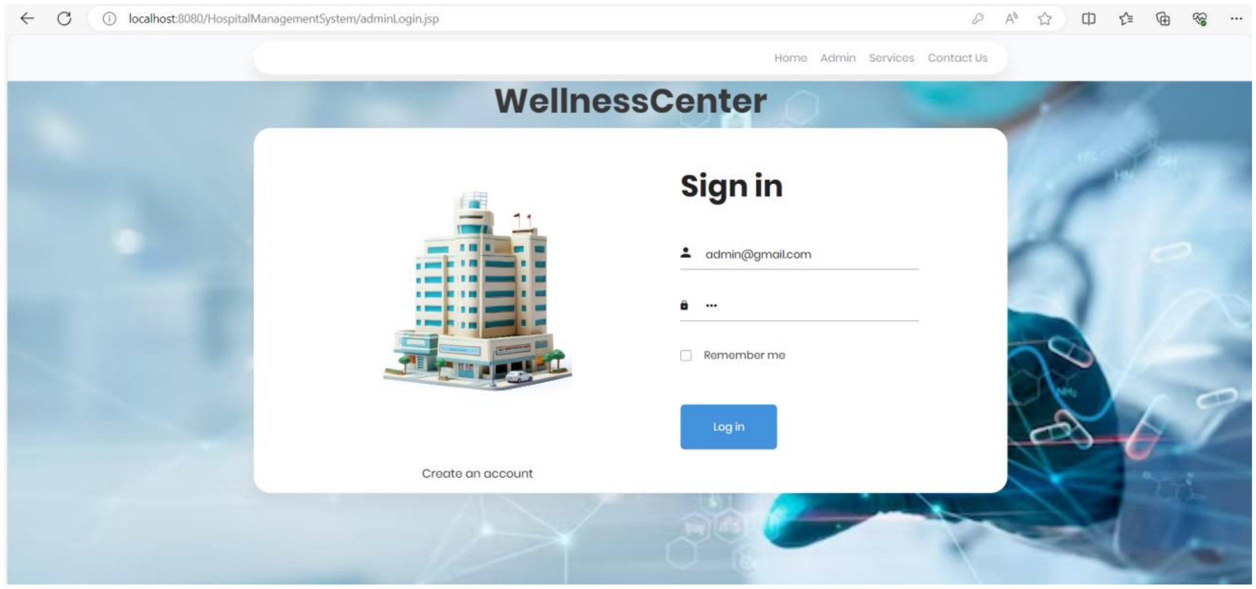
By adopting a comprehensive testing approach encompassing manual and automated methodologies, unit and integrated testing, security assessments, performance evaluations, usability tests, and compliance validations, the Hospital Management System aims to deliver a robust, secure, and user-friendly solution for effective healthcare management. Testing is an ongoing process, with continuous refinement and improvement to ensure the system's reliability, scalability, and compliance with regulatory standards and user expectations

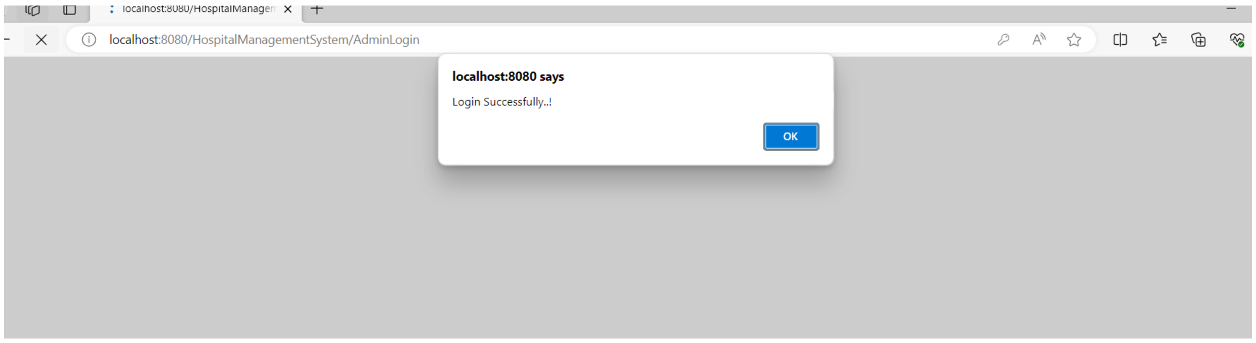
8.1 Test Cases



Test Scenario ID:	1					
Tester Name	RavindraPatil					
Test Page	admin ogin(Hospitalmanagement System					
TestID	Test Scenario	Test Cases	Test Data	Actual output	Excepted Output	Status
1	Enter correct email	1.open website	http://localhost:8080/HospitalManagementSyster	login successfully	login successfully	Pass
	Enter correct password	2.open login page	admin login			
		3.Enter email id	admin@gmail.com			
		4.Enter password	123			
		5.click on login button				
2	Enter correct email	1.open website	http://localhost:8080/HospitalManagementSyster	incorrect username or password	login successfully	Fail
	Enter incorrect password	2.open login page	admin login			
		3.Enter email id	admin@gmail.com			
		4.enter password	1234			
3	Enter InCorrect Email	1.open website	http://localhost:8080/HospitalManagementSyster	incorrect username or password	login successfully	Fail
	Enter Correct password	2.open login page	admin login			
		3.Enter email id	Ravi@gmail.com			
		4.Enter Password	123			
4	Enter incorrect Email	1.open website	http://localhost:8080/HospitalManagementSyster	incorrect username or password	login successfully	Fail
	Enter incorrect password	2.open login page	admin login			
		3.Enter email Id	Ravi@gmail.com			
		4.enter password	1234			

8.2 Testing Outputs

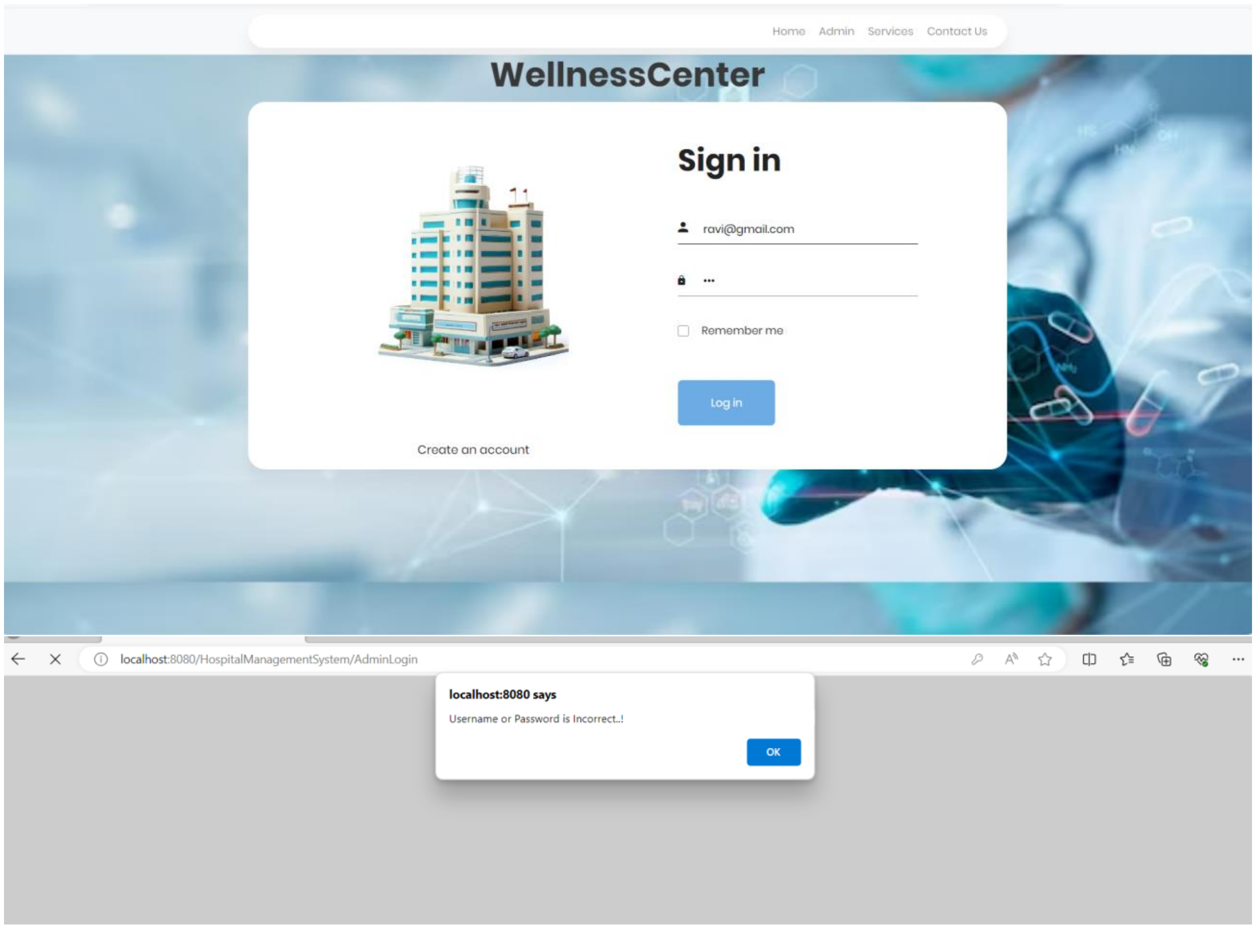

localhost:8080/HospitalManagementSystem/adminLogin.jsp


localhost:8080/HospitalManagementSystem/AdminLogin

localhost:8080 says

Login Successfully..!

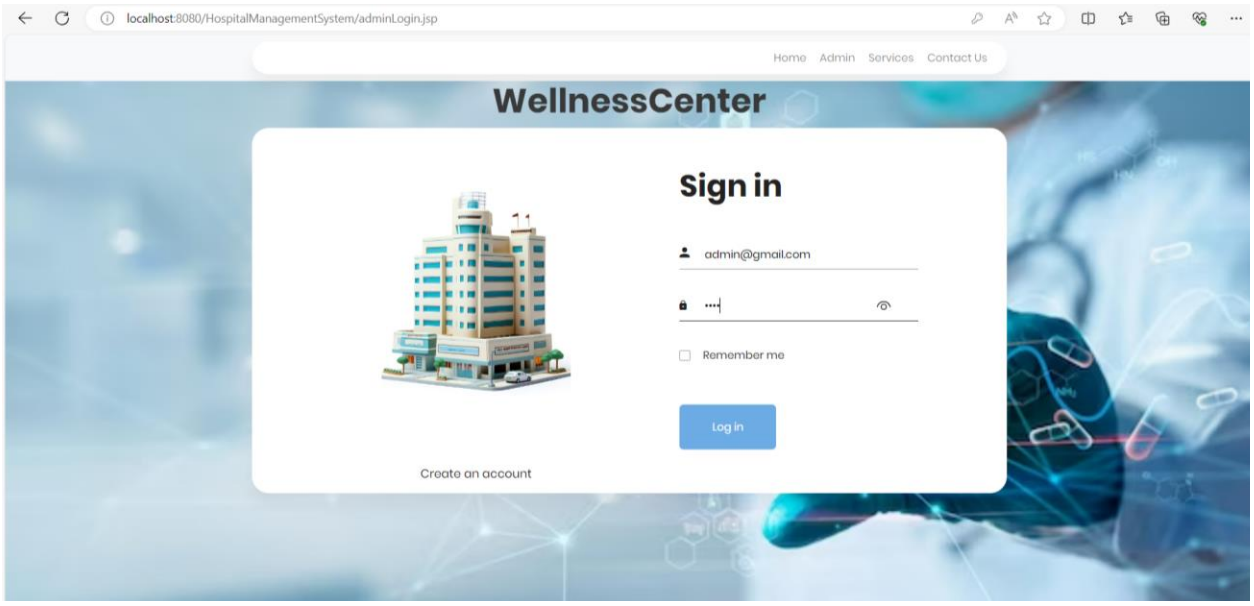
Test2: Incorrect Email and Correct Password

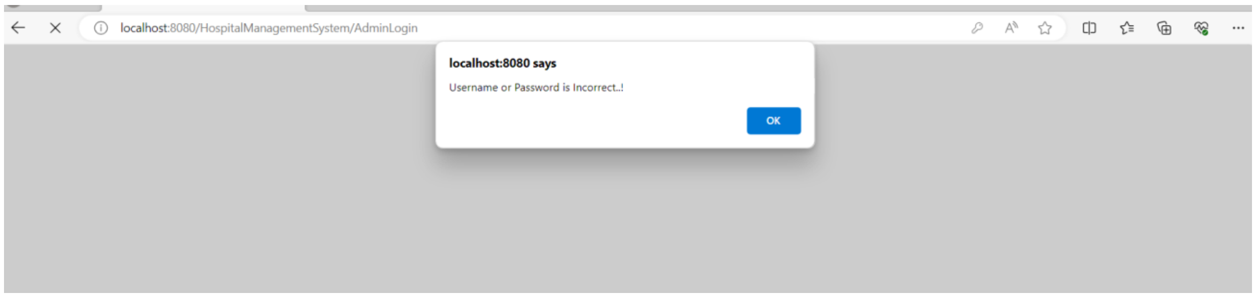


localhost:8080 says

Username or Password is Incorrect..!

Test3: Correct Id and InCorrect Password



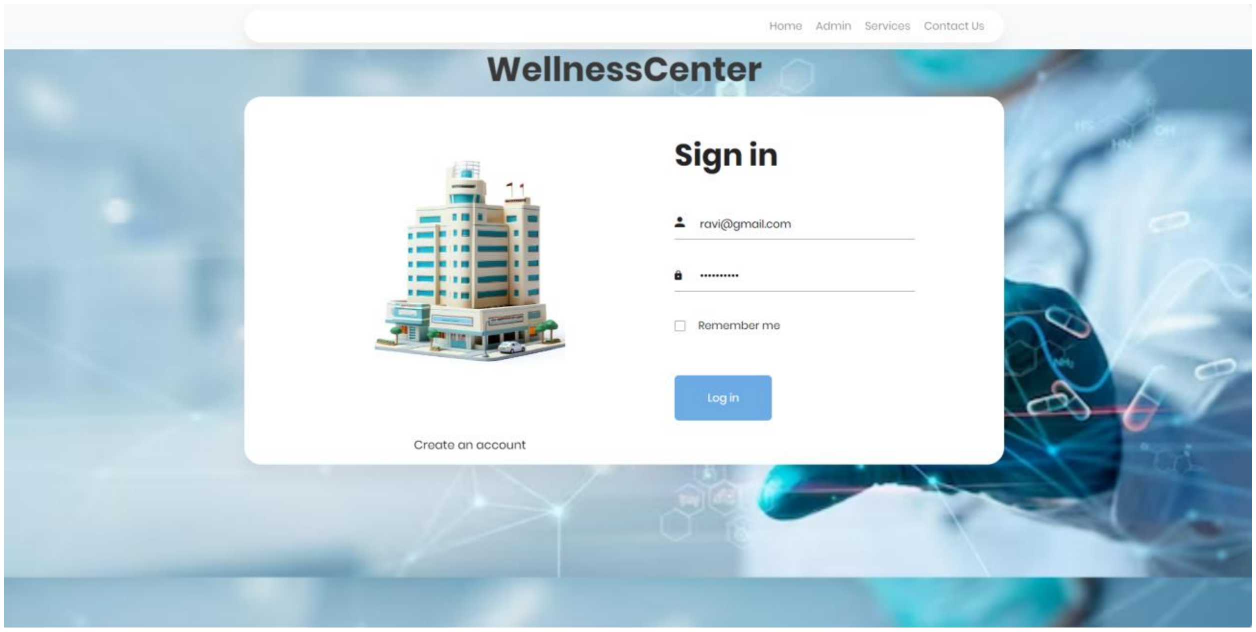

localhost:8080/HospitalManagementSystem/AdminLogin

localhost:8080 says

Username or Password is Incorrect..!

OK

Test 4: InCorrect email and Password


Home Admin Services Contact Us

WellnessCenter

Sign in

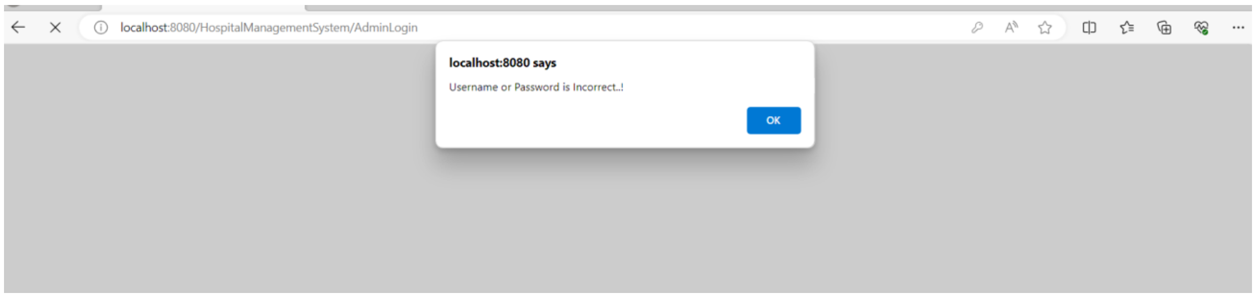
ravi@gmail.com

••••••••••

Remember me

Log in

Create an account


localhost:8080/HospitalManagementSystem/AdminLogin

localhost:8080 says

Username or Password is Incorrect..!

OK

Chapter – 9 Maintenance

In the maintenance phase of the Hospital Management System (HMS), our primary focus is on ensuring the system's reliability, security, and adaptability to meet the evolving needs of healthcare organizations. This phase encompasses a proactive approach aimed at addressing potential issues, implementing enhancements, and promptly responding to user feedback.

We prioritize regular bug fixes to rectify any identified issues promptly. Utilizing a robust bug tracking system, we streamline the resolution process to ensure that reported issues are addressed efficiently, minimizing disruption to hospital operations.

Security remains a top priority, and we conduct routine security audits to safeguard sensitive patient data and ensure compliance with healthcare regulations. Timely updates are implemented to fortify the system's resilience against emerging security threats, providing peace of mind to healthcare providers and patients alike.

User feedback is invaluable, and we actively seek and analyze it to gain insights into system usability and functionality. This feedback-driven approach guides us in identifying areas for improvement, allowing us to prioritize and implement enhancements aligned with user expectations.

Performance optimization is another key aspect of our maintenance strategy. By monitoring system metrics, identifying bottlenecks, and implementing optimizations, we enhance the overall efficiency and responsiveness of the HMS, ensuring a seamless user experience for healthcare professionals and staff.

Comprehensive documentation is consistently updated to reflect modifications and new features, providing users and developers with accurate and relevant information for smooth system maintenance and troubleshooting.

Regular database maintenance tasks, including indexing and data cleanup, contribute to sustained database performance, ensuring efficient data management and retrieval within the HMS.

Compliance checks are conducted regularly to ensure adherence to industry standards and regulations related to hospital management, safeguarding patient privacy and data security.

Continuous training and knowledge transfer sessions are conducted for system administrators and support staff to keep them informed about procedural changes and updates, enabling effective management of the HMS.

9.1 Modifications and Improvements

1. **Bug Fixes and Issue Resolution:** Regular bug fixes are conducted to address any identified issues promptly. A robust bug tracking system is utilized to streamline the resolution process, ensuring that reported issues are prioritized and resolved efficiently.
2. **Security Updates:** The system's resilience against emerging security threats is maintained through routine security audits and timely updates. This helps safeguard sensitive patient data and ensures compliance with healthcare regulations.
3. **User Feedback Analysis:** User feedback is actively sought and analyzed to gain insights into system usability and functionality. This feedback-driven approach guides the identification of areas for improvement, allowing for enhancements aligned with user expectations.
4. **Performance Optimization:** Performance optimization efforts focus on monitoring system metrics, identifying bottlenecks, and implementing optimizations to enhance overall efficiency and responsiveness. This ensures a seamless user experience for healthcare professionals and staff.
5. **Documentation Maintenance:** Comprehensive documentation is consistently updated to reflect modifications and new features. This ensures that users and developers have access to accurate and relevant information, facilitating smooth system maintenance and troubleshooting.
6. **Database Maintenance:** Regular database maintenance tasks, including indexing and data cleanup, contribute to sustained database performance. This ensures efficient data management and retrieval within the HMS.
7. **Compliance Checks:** Compliance checks are conducted regularly to ensure that the system aligns with industry standards and regulations related to hospital management. Any necessary adjustments are made to maintain compliance and data security.
8. **Training and Knowledge Transfer:** Continuous training and knowledge transfer sessions are conducted for system administrators and support staff. This keeps them informed about procedural changes and updates, ensuring effective management of the HMS.
9. **Backup and Recovery Testing:** Rigorous testing of backup and recovery processes is executed to guarantee data integrity and facilitate prompt recovery in case of unforeseen events. This ensures continuity of operations and minimizes downtime.
10. **Scalability Assessments:** Regular assessments are conducted to evaluate the system's scalability and its ability to adapt to evolving requirements. This ensures that the HMS can accommodate growing healthcare needs and technological advancements.
11. **Monitoring and Alerts:** Monitoring tools are implemented to track system health in real-time, with alerts set up for critical events. This enables proactive issue resolution and ensures optimal system performance.

Chapter – 10 (Conclusion)

10.1 Conclusion

In conclusion, the maintenance phase plays a pivotal role in ensuring the hospital management system's long-term success and effectiveness. It involves ongoing efforts to enhance, optimize, and adapt the system to meet evolving needs and challenges in the healthcare industry. By focusing on various aspects such as bug fixes, security updates, user feedback analysis, performance optimization, feature enhancements, documentation updates, database maintenance, compliance checks, training and knowledge transfer, backup and recovery testing, scalability assessment, vendor and technology evaluation, continuous monitoring, user training sessions, and collaboration with stakeholders, hospitals can effectively manage and improve their operations.

Furthermore, the maintenance phase is essential for upholding patient safety, data security, regulatory compliance, and overall system reliability. It ensures that the hospital management system remains responsive, efficient, and user-friendly, enabling healthcare providers to deliver high-quality care and services to patients. Additionally, by proactively addressing issues and incorporating user feedback, hospitals can enhance patient satisfaction, streamline workflows, and optimize resource utilization.

10.2 Future Enhancements

1. **Telemedicine Integration:** Implement telemedicine features to allow remote consultations, enabling patients to receive medical advice and treatment without the need for physical appointments. This enhancement can improve accessibility to healthcare services, especially for patients with mobility issues or those living in remote areas.
2. **Predictive Analytics:** Integrate predictive analytics tools to analyze patient data and predict potential health issues or trends. By leveraging machine learning algorithms, the system can identify patients at risk of certain conditions, allowing healthcare providers to intervene early and provide proactive care.
3. **IoT Integration:** Incorporate Internet of Things (IoT) devices such as wearables and smart medical devices to collect real-time patient data. This data can be used to monitor patient health metrics remotely, provide personalized treatment plans, and track patient progress more accurately.
4. **Enhanced Mobile App:** Improve the hospital management system's mobile application to offer additional functionalities such as appointment scheduling, prescription refills, access to medical records, and communication with healthcare providers. A user-friendly and feature-rich mobile app can enhance patient engagement and satisfaction.
5. **Patient Portal Expansion:** Expand the capabilities of the patient portal to include features such as secure messaging with healthcare providers, online bill payment, access to lab results, and educational resources. Empowering patients with more control over their healthcare experience can improve patient satisfaction and outcomes.
6. **Integration with Electronic Health Records (EHR):** Integrate the hospital management system with EHR systems to streamline data sharing and improve interoperability. This integration enables healthcare providers to access comprehensive patient information

across different healthcare settings, leading to better-informed decision-making and coordinated care.

7. **Virtual Reality (VR) Training:** Implement virtual reality training modules for healthcare professionals to simulate medical procedures, surgeries, and emergency scenarios in a safe and controlled environment. VR training can enhance medical education, improve clinical skills, and reduce the risks associated with traditional training methods.
8. **Blockchain for Medical Records:** Explore blockchain technology to secure and decentralize medical records, ensuring data integrity, privacy, and security. Blockchain-based medical records can provide patients with more control over their data while facilitating secure and transparent sharing of information among healthcare providers.
9. **Automated Prescription Refills:** Develop automated prescription refill features that allow patients to request prescription refills online and receive notifications when their medications are ready for pickup or delivery. This enhancement streamlines the prescription renewal process, improves medication adherence, and reduces administrative burden for healthcare providers.
10. **Advanced Analytics Dashboard:** Create an advanced analytics dashboard for hospital administrators and executives to visualize key performance indicators, financial metrics, patient outcomes, and resource utilization data. The dashboard can facilitate data-driven decision-making, identify areas for improvement, and optimize hospital operations for better efficiency and quality of care.

Chapter – 11 (References)

https://chat.openai.com/

hospital appointment booking system - YouTube

http://www.plantuml.com/plantuml/uml/SyfFKj2rKt3CoKnELR1Io4ZDoSa70000/

https://www.softwaretestinghelp.com/