

Table of Contents

Sl. No.	Topic	Page No.
1	Introduction	1
	i. Overview of the Hospital Management System	2
	ii. Objectives	3
	iii. Technologies Used	5
2	System Requirements	7
	i. Hardware Requirements	7
	ii. Software Requirements	7
3	System Architecture	9
	i. System Design	9
	ii. Module Overview	10
4	Setup and Installation	12
	i. Installing XAMPP	12
	ii. Setting Up the Web Server	12
	iii. Configuring PHP and MySQL	13
	iv. Installing Bootstrap	13
	v. Integrating TCPDF	13

5	Database Design	14
	i. Database Schema	14
	ii. Table Structures	14
	iii. Relationships	16
6	Front-End Development	18
	i. HTML5 and CSS3 for Layout and Styling	18
	ii .Bootstrap for Responsive Design	19
	iii. JavaScript for Dynamic Content	19
7	Back-End Development	20
	i. PHP for Server-Side Scripting	20
	i. Handling User Requests	21
	ii. MySQL for Database Management	22
	i. Data Retrieval and Manipulation	22
8	User Modules	23
	Patient Registration	23
	Appointment Scheduling	23
9	Admin Modules	25
	User Management	25

10	Reporting	27
	Generating Reports with TCPDF	27
11	Security	28
	Input Validation and Sanitization	28
	User Authentication and Authorization	28
12	Maintenance and Support	29
	Contact Support	29
13	Result	30
14	Conclusion	34
	Summary of the Project	34
	Future Enhancements	34
15	References	35

List of Figures:

Sl. No.	Topic	Page No.
---------	-------	----------

3.1	System Architecture	11
5.1	MySQL DB	17
6.1	HTML5	18
6.2	CSS3	18
6.3	Bootstrap	19
6.4	JavaScript	19
7.1	PHP	21
8.1	Patient Login	23
8.2	Patient Dashboard	24
8.3	Patient Booking Appointment	24
9.1	Admin Panel	26
11.1	Security	28
12.1	Maintenance	29
13.1	Patient Registration	30
13.2	Patient Dashboard	30
13.3	Patient Booking Appointment	30
13.4	Doctor Login Page	31
13.5	Doctor Dashboard	31

13.6	Doctor Appointment List	31
13.7	Doctor Prescription List	32
13.8	Patient Appointment History	32
13.9	Patient Prescription	32
13.1.1	Patient Bill	33
13.1.2	Receptionist(Admin)	33
13.1.3	Admin Dashboard	33

<https://github.com/TechnoDk>

Given By:- Deepak Kumar Sah

Email:- <mailto:dkdear11@gmail.com?subject=Report Help>

Mob:-9065034065



1. Introduction

The project Hospital Management System includes appointment booking, registration of patients, doctors, and storing their details into the system. The software has the facility to give a unique ID for every patient and stores the details of every patient and the staff automatically. The project can be entered using a username and password. Users can search for various services available, book appointments, view doctor details, and doctors can view their appointments and patient details using their ID. It is accessible either by an admin or staff. Only they can add data into the database. The data can be retrieved easily.

The Hospital Management System (HMS) is an advanced software solution designed to optimize healthcare operations using modern web technologies. It leverages the capabilities of modern web technologies such as HTML5, CSS3, JavaScript, Bootstrap, PHP, MySQL, and TCPDF. The system aims to provide a comprehensive, efficient, and secure platform for managing hospital activities. The interface is very user-friendly.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals, clinics, etc. The system has three modules: Admin, Doctor, and Patient.

Additionally, the HMS facilitates seamless integration with existing hospital infrastructure, ensuring smooth data migration and interoperability. It supports multi-user access, allowing multiple departments to work concurrently without data conflicts. The system includes robust security features such as data encryption, user authentication, and role-based access control to safeguard sensitive information.

Moreover, the HMS is equipped with reporting and analytics tools that provide valuable insights into hospital performance, patient demographics, and treatment outcomes, aiding in informed decision-making and strategic planning.

This system has three modules.

- Admin
- Doctor
- Patient

i. Overview of the Hospital Management System

The Hospital Management System (HMS) is a comprehensive software solution designed to manage the complex operations of healthcare facilities efficiently. It integrates various functionalities required for the smooth running of a hospital, from patient registration to billing and reporting, using a combination of modern web technologies.

☐ Patient Management

- Registration and Profile Management: Patients can easily register, create profiles, and update their information.
- Appointment Scheduling: Patients can book appointments with doctors, view available time slots, and receive reminders.
- Medical Records: Access to electronic medical records (EMR) with detailed patient history, diagnoses, treatments, and prescriptions.

☐ Doctor Management

- Appointment Management: Doctors can manage their schedules, view upcoming appointments, and minimize patient wait times.
- Electronic Medical Records: Easy access to patient records for accurate diagnosis and treatment.
- Medical Reporting: Generation and review of diagnostic reports, lab results, and prescriptions.
- Patient Communication: Direct communication with patients for follow-ups and consultations.

☐ Admin Management

- User and Role Management: Creating and managing accounts for doctors, nurses, and other staff with role-based access control.
- System Configuration: Setting up hospital departments, managing inventory, and configuring system settings.
- Reporting: Generating comprehensive reports on patient statistics, financial data, and inventory status for better decision-making.

☐ Billing and Payments

- Billing Management: Automated billing processes, reducing errors, and ensuring timely payments.

- Payment Processing: Secure online payment options for patients.
- Receipts and Invoices: Generation of detailed receipts and invoices for transparency.
- Security and Compliance
- Data Encryption: Ensuring the security and privacy of sensitive patient data.
- Access Control: Role-based access to ensure only authorized personnel can access specific information.
- Compliance: Adherence to healthcare regulations and standards.

ii. Objectives of the Hospital Management System

The Hospital Management System (HMS) is designed with several key objectives in mind to enhance the overall functionality and efficiency of healthcare institutions. These objectives include:

1. Automation of Administrative Tasks:
 - Reduce manual workload by automating routine administrative processes such as patient registration, appointment scheduling, billing, and inventory management.
 - Minimize human errors and improve accuracy in administrative operations.
2. Improvement of Patient Care:
 - Provide healthcare professionals with quick and easy access to comprehensive patient records, enabling accurate diagnosis and effective treatment.
 - Enhance patient experience by reducing wait times through efficient appointment scheduling and management.
3. Enhanced Data Management and Accessibility:
 - Centralize all patient and hospital data in a secure database, allowing authorized personnel to access and update information as needed.
 - Ensure that all medical records, billing information, and inventory data are easily retrievable and up-to-date.
4. Efficient Resource Utilization:
 - Optimize the use of hospital resources, including medical supplies, equipment, and staff, through effective inventory management and scheduling.
 - Prevent shortages and ensure that necessary resources are always available when needed.
5. Secure and Compliant Data Handling:

- Implement robust data encryption and access control mechanisms to protect sensitive patient information.
 - Ensure compliance with healthcare regulations and standards, maintaining the confidentiality, integrity, and availability of data.
6. Streamlined Communication and Collaboration:
- Facilitate better communication between patients and healthcare providers through integrated communication tools.
 - Enhance collaboration among doctors, nurses, and administrative staff to improve overall hospital efficiency.
7. Comprehensive Reporting and Analytics:
- Generate detailed reports on various aspects of hospital operations, including patient statistics, financial performance, and inventory status.
 - Provide administrators with actionable insights for informed decision-making and strategic planning.
8. Scalability and Flexibility:
- Design the system to be scalable, accommodating the growth of healthcare institutions and the increasing volume of data.
 - Ensure flexibility to adapt to the specific needs and workflows of different hospitals and clinics.
9. Cost Reduction:
- Lower operational costs by reducing manual labor and improving the efficiency of administrative processes.
 - Optimize resource allocation and minimize waste through effective inventory and resource management.
10. Enhanced Patient Engagement:
- Provide patients with user-friendly tools to manage their appointments, access medical records, and handle billing and payments.
 - Improve patient satisfaction by offering a transparent and streamlined healthcare experience.

iii. Technologies Used in the Hospital Management System

The Hospital Management System (HMS) leverages a variety of modern web technologies to ensure robust functionality, scalability, and user-friendly interfaces. The key technologies used in the development of the HMS include:

1. HTML5 and CSS3
 - o HTML5: Used for structuring the web pages and presenting content in a meaningful way. It provides the backbone of the user interface, ensuring semantic and accessible web content.
 - o CSS3: Utilized for styling the web pages, making the interface visually appealing and responsive across different devices. CSS3 helps in creating a consistent look and feel throughout the application.
2. JavaScript
 - o JavaScript is used to create dynamic and interactive content on the web pages. It enhances user experience by enabling real-time updates, form validation, and interactive elements such as modals, dropdowns, and notifications.
3. Bootstrap
 - o Bootstrap is a popular front-end framework that simplifies the design and development of responsive web applications. It provides pre-designed components, grid systems, and utilities, allowing for a consistent and mobilefriendly layout.
4. PHP
 - o PHP is a server-side scripting language used to handle backend operations. It processes user requests, manages sessions, interacts with the database, and generates dynamic content. PHP ensures the efficient handling of server-side logic and data processing.
5. MySQL
 - o MySQL is a relational database management system (RDBMS) used to store and manage the application's data. It handles the creation, retrieval, updating, and deletion (CRUD) operations on the database. MySQL ensures data integrity, security, and efficient query processing.
6. XAMPP
 - o XAMPP is a cross-platform web server solution stack package that includes Apache, MySQL, PHP, and Perl. It provides a local development environment for building and testing the HMS before deploying it to a production server. XAMPP simplifies the setup and configuration of the development environment.
7. TCPDF
 - o TCPDF is an open-source PHP library used for generating PDF documents. It allows the creation of professional and customizable PDF reports, such as patient records, billing invoices, and medical reports. TCPDF ensures that documents can be easily generated, formatted, and printed directly from the application.

8. JavaScript Libraries and Frameworks
 - o jQuery: A fast and concise JavaScript library that simplifies event handling, AJAX interactions, and DOM manipulation. It enhances the interactivity and responsiveness of the web pages.
 - o AJAX: Used for asynchronous data loading and updating without reloading the entire web page. AJAX enhances user experience by enabling seamless data exchange between the client and server.

TechnoDK

2. System Requirements & Architecture

i. Hardware Requirements

For Server:

- Processor: Quad-core processor (Intel i5 or equivalent) or higher
- RAM: Minimum 8 GB (16 GB recommended for larger installations)
- Storage: Minimum 250 GB of SSD storage (500 GB recommended)
- Network: High-speed internet connection with a stable network environment
- Backup Device: External hard drive or cloud storage solution for regular backups

For Client Machines:

- Processor: Dual-core processor (Intel i3 or equivalent) or higher
- RAM: Minimum 4 GB (8 GB recommended)
- Storage: Minimum 100 GB of storage
- Display: 1024 x 768 resolution or higher
- Network: High-speed internet connection or reliable local network connection ii .

Software Requirements

For Server:

- Operating System: Windows Server, Linux (Ubuntu, CentOS, or similar distributions), or macOS
- Web Server: Apache (included in XAMPP)
- Database Server: MySQL (included in XAMPP)
- Scripting Language: PHP (included in XAMPP)
- Development Environment: XAMPP or similar LAMP/WAMP/MAMP stack
- PDF Generation Library: TCPDF

For Client Machines:

- Operating System: Windows 10/11, Linux, or macOS

- Web Browser: Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari
- PDF Reader: Adobe Acrobat Reader or any compatible PDF reader.

□ Additional Requirements Development

Tools:

- IDE/Code Editor: Visual Studio Code, Sublime Text, PHPStorm, or similar code editors
- Version Control System: Git (with platforms like GitHub, GitLab, or Bitbucket)
- Package Manager: Composer (for PHP dependency management)

Security Requirements:

- SSL Certificate: For secure data transmission (HTTPS)
- Firewall: To protect against unauthorized access and threats
- Antivirus/Anti-malware Software: To ensure the server and client machines are protected from malicious software

Optional:

- Email Server: For sending notifications and alerts (e.g., appointment reminders, billing invoices)
- Backup Solution: Automated backup software for regular data backups to external drives or cloud storage.

□ Architecture

The Hospital Management System (HMS) is designed with a multi-tier architecture to ensure scalability, maintainability, and efficiency. The architecture comprises the following layers:

1. Presentation Layer:

- Components: HTML5, CSS3, JavaScript, Bootstrap
- Function: This layer is responsible for the user interface and user experience. It handles the presentation of data and user interactions, ensuring a responsive and intuitive design.

2. Application Layer:

- Components: PHP, JavaScript (for client-side logic)
- Function: This layer processes the business logic of the application. It handles user requests, processes data, interacts with the database, and returns the appropriate responses to the presentation layer.

3. Data Layer:

- Components: MySQL Database
- Function: This layer manages the storage, retrieval, and manipulation of data. It ensures data integrity, security, and efficient query processing.

4. PDF Generation Layer:

- Components: TCPDF
- Function: This layer is responsible for generating PDF documents, such as medical reports, billing invoices, and patient records, in a professional and customizable format.

i . System Design

The system design focuses on creating a modular, scalable, and maintainable structure. The main components and their interactions are outlined as follows:

1. User Interface:

- HTML5/CSS3: Defines the structure and styling of web pages.
- Bootstrap: Ensures responsive and consistent design.

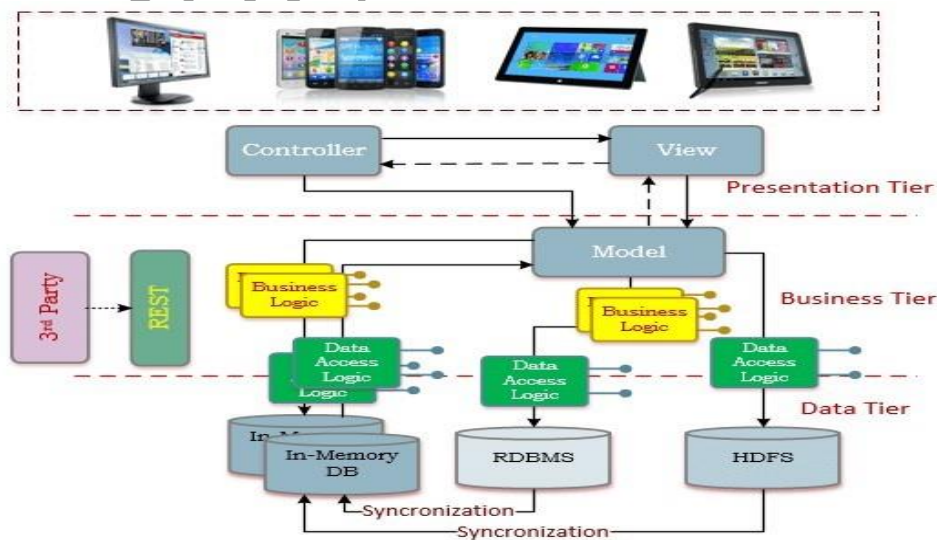
- o JavaScript/jQuery: Enhances interactivity and dynamic content updates.
- 2. Business Logic:
 - o PHP Scripts: Handle user authentication, session management, data processing, and business rules implementation.
 - o JavaScript: Manages client-side logic, form validations, and asynchronous data loading (AJAX).
- 3. Database Management:
 - o MySQL Database: Stores all application data, including patient records, appointments, billing information, and inventory data.
 - o SQL Queries: Facilitate data retrieval, insertion, update, and deletion operations.
- 4. PDF Generation: o TCPDF Library: Generates PDFs for reports, invoices, and medical records.
- 5. Web Server:
 - o XAMPP (Apache Server): Hosts the application, processes PHP scripts, and serves web pages to users.
- 6. Security:
 - o SSL/TLS: Ensures secure data transmission.
 - o Access Control: Role-based access to restrict user permissions.
 - o Data Encryption: Protects sensitive data in transit and at rest.

ii . Module Overview

The HMS is divided into several key modules, each responsible for specific functionalities:

1. User Management Module:
 - o Admin: Manages user accounts, roles, and permissions.
 - o Doctor: Accesses patient records, schedules, and communications.
 - o Patient: Manages profile, appointments, and medical history.
2. Patient Management Module:
 - o Registration: Patient onboarding and profile creation.

- Appointment Scheduling: Booking and managing appointments with doctors.
 - Medical Records: Storing and retrieving patient medical history and treatment details.
3. Doctor Management Module:
- Schedule Management: Viewing and managing appointment schedules.
 - Medical Reporting: Generating and reviewing diagnostic reports.
 - Patient Communication: Facilitating follow-ups and consultations.
4. Billing and Payment Module:
- Billing: Automated generation of billing invoices.
 - Payment Processing: Handling online payments and receipts.
5. Inventory Management Module:
- Stock Management: Tracking medical supplies and equipment.
 - Order Management: Managing orders for new supplies.
6. Reporting Module:
- Admin Reports: Generating detailed reports on hospital operations, patient statistics, and financial data.
 - Medical Reports: Creating medical reports and diagnostic summaries for patients.



Fig(2.1):- System Architecture

3. Setup and Installation

i. Installing XAMPP:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists of Apache HTTP Server, MariaDB (MySQL), and interpreters for scripts written in PHP and Perl.

Steps:

- Download XAMPP:
 - Visit the XAMPP download page.
 - Download the version compatible with your operating system (Windows, Linux, or macOS).
- Install XAMPP:
 - Run the downloaded installer.
 - Follow the on-screen instructions to complete the installation.
 - Choose the components you need (Apache, MySQL, PHP, phpMyAdmin).
- Start XAMPP:
 - Launch the XAMPP Control Panel.
 - Start the Apache and MySQL services.

ii. Setting Up the Web Server:

Steps:

- Navigate to XAMPP Directory:
 - On Windows: D:\xampp\htdocs
 - On macOS/Linux: /Applications/XAMPP/htdocs
- Create Project Directory:
 - Create a new directory for your Hospital Management System project.
- Place Project Files:
 - Copy all your HTML, CSS, JavaScript, PHP files, and folders into the hospital management directory.

iii. Configuring PHP and MySQL:

Steps:

- Configure PHP:
 - Open the XAMPP Control Panel.
 - Click on the Config button next to Apache and select php.ini. ◦ Make any necessary configurations (e.g., changing max_execution_time, memory_limit, etc.). ◦ Save and close the file.
- Set Up MySQL Database:
 - Open phpMyAdmin by navigating to <http://localhost/phpmyadmin> in your web browser. ◦ Create a new database for your project, e.g. 'myhmsdb'.
 - Import the file 'myhmsdb.sql' inside your newly created database and click ok

iv. Installing Bootstrap:

Steps:

- Download Bootstrap:
 - Visit the [Bootstrap website](#).
 - Download the latest version of Bootstrap.
- Include Bootstrap in Your Project:
 - Extract the downloaded files. ◦ Place the bootstrap.min.css and bootstrap.min.js files into your project directory (e.g., css and js folders).

v. Integrating TCPDF:

Steps:

- Download TCPDF:
 - Visit the [TCPDF website](#). ◦ Download the latest version of TCPDF.

4 .Technologies Used

□ Database Design

Designing a robust and efficient database schema is crucial for managing the complex data and relationships within a hospital management system. Below is an overview of the database schema, including table structures and relationships.

i.Database Schema

The database schema for the Hospital Management System will include the following main tables:

1. Users
2. Patients
3. Doctors
4. Appointments
5. MedicalRecords
6. Billing
7. Departments

ii.Table Structures

1.Users

This table stores information about all users of the system, including admins, doctors, and other staff.

```
sql
CREATE TABLE Users (
  user_id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  role ENUM('admin', 'doctor', 'staff') NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Patients

```
sql
CREATE TABLE Patients (
  patient_id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  date_of_birth DATE NOT NULL,
  gender ENUM('male', 'female', 'other') NOT NULL,
  contact_number VARCHAR(15),
  email VARCHAR(100) UNIQUE,
  address TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

3. Doctors

This table stores information about doctors.

```
sql
CREATE TABLE Doctors (
  doctor_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  specialization VARCHAR(100) NOT NULL,
  contact_number VARCHAR(15),
  email VARCHAR(100) UNIQUE,
  department_id INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);
```

4. Appointments

This table stores appointment details.

```
sql
CREATE TABLE Appointments (
  appointment_id INT AUTO_INCREMENT PRIMARY KEY,
  patient_id INT NOT NULL,
  doctor_id INT NOT NULL,
  appointment_date DATETIME NOT NULL,
  status ENUM('scheduled', 'completed', 'cancelled') DEFAULT 'scheduled',
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
  FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

5. MedicalRecords

This table stores medical records of patients.

```
sql
CREATE TABLE MedicalRecords (
  record_id INT AUTO_INCREMENT PRIMARY KEY,
  patient_id INT NOT NULL,
  doctor_id INT NOT NULL,
  visit_date DATETIME NOT NULL,
  diagnosis TEXT,
  treatment TEXT,
  prescription TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
  FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

6. Billing

This table stores billing information

```
sql
CREATE TABLE Billing (
  bill_id INT AUTO_INCREMENT PRIMARY KEY,
  patient_id INT NOT NULL,
  amount DECIMAL(10, 2) NOT NULL,
  payment_status ENUM('paid', 'pending') DEFAULT 'pending',
  billing_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (patient_id) REFERENCES Patients(patient_id)
);
```

8. Departments

This table stores information about hospital departments.

```
sql
CREATE TABLE Departments (
  department_id INT AUTO_INCREMENT PRIMARY KEY,
  department_name VARCHAR(100) NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

iii. Relationships

The relationships between the tables are established using foreign keys. Below is a summary of the key relationships:

1. Users and Doctors:

- o Each doctor is associated with a user account for authentication and authorization purposes.
- o Relationship: One-to-One (Users to Doctors via user_id)

2. Patients and Appointments:

- o Each appointment is linked to a specific patient. o Relationship: One-to-Many (Patients to Appointments via patient_id)

3. Doctors and Appointments:

- o Each appointment is linked to a specific doctor. o Relationship: One-to-Many (Doctors to Appointments via doctor_id)

4. Patients and MedicalRecords:

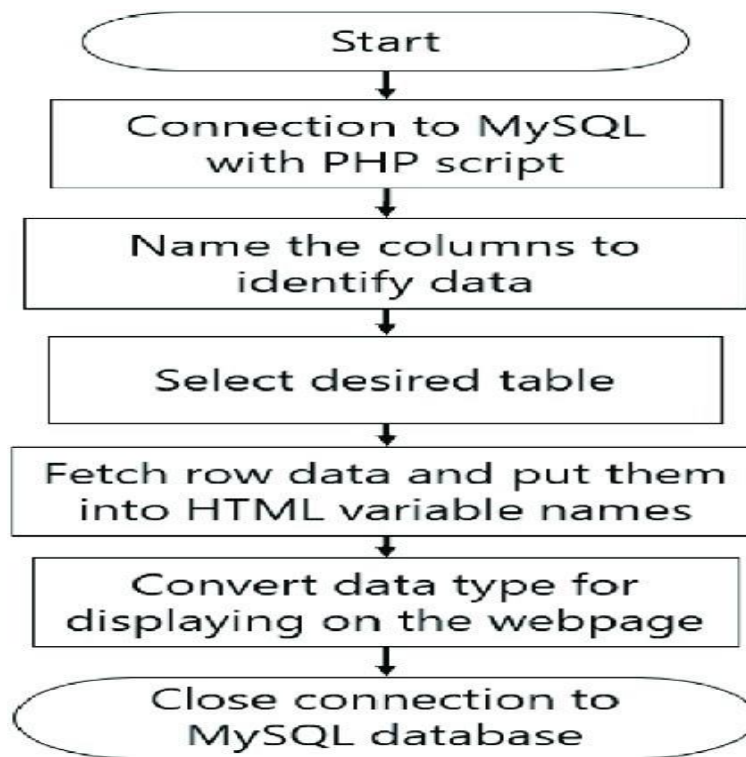
- o Each medical record is linked to a specific patient. o Relationship: One-to-Many (Patients to MedicalRecords via patient_id)

5. Patients and Billing:

- o Each billing record is linked to a specific patient. o Relationship: One-to-Many (Patients to Billing via patient_id)

6. Doctors and Departments:

- o Each doctor belongs to a specific department. o Relationship: Many-to-One (Doctors to Departments via department_id)



FlowChart (4.1):-MySQL DB

□ Front-End Development

i. HTML5

HTML stands for Hyper-text-Markup Language. It is the standard markup language that describes the structure of a web page. It consists of a series of elements that tell the browser how to display the content and these elements are represented by tags. The browser doesn't display the tags on the screen but uses the HTML tags to render the content of the web page. HTML is the root of any web-based application which is easy to learn and use, therefore it is not always considered as an important language.



Fig (4.2): - HTML5

i.1.CSS3

Cascading Style Sheets (CSS) are used for designing and presenting the HTML page, describes how HTML elements are to be displayed on different devices, and directly controls the layout of site on multiple screen sizes. It manipulates the text colors, fonts, background colors, layout, etc. The latest flexbox and grid concept in css has made it easier to layout the document. CSS can be embedded in HTML in three ways, i.e. inline css, internal css and external css. To add css internally <style> tag is used.



Fig (4.3):- CSS3

ii . Bootstrap

Bootstrap is a collection of free and open-source tools for building responsive websites and web applications. An immense framework for HTML, CSS, and JavaScript for developing responsive websites. It solves many problems we used to have, one of which is a browser compatibility problem.



Fig (4.4):-Bootstrap

iii . JavaScript

JavaScript (JS) is a structured language that is lightweight, translated or has recently been integrated into functional tasks in the first grade. Js is popular and well known for writing Web pages; many non-browser sites also use it, as Node.js, Apache-CouchDB, and Adobe Acrobat. It is a stylebased, elementary, series, language based, supported object based, essential, and dictionary (e.g. functional programs). Toda JavaScript latest level is ECMAScript. ECMAScript 5.1 is supported by all modern browsers and at least ECMAScript

3 is supported by older browsers. With the help of JavaScript one can manipulate Dom from manage static forms to dynamic, the language is so vast and widely used that it is not only supported by all browsers but also helps debugging it.



Fig (4.5):- JavaScript

☐ Back-End Development

i. PHP for Server-Side Scripting

PHP (Hypertext Preprocessor) is a widely-used open-source scripting language especially suited for web development. It can be embedded into HTML and is executed on the server, generating dynamic page content.

☐ Server-Side Scripting:

- PHP scripts are executed on the server, and the output is sent to the client's browser as plain HTML.
- This allows for dynamic content generation, such as retrieving data from a database and displaying it on a web page.

☐ Simple and Easy to Learn:

- PHP is known for its simple syntax, which is similar to C, Java, and Perl, making it easy to learn for those familiar with these languages.
- It is designed to be embedded directly into HTML, simplifying the development process.

☐ Cross-Platform:

- PHP is cross-platform and can run on various operating systems like Windows, Linux, Unix, and macOS.
 - It supports a wide range of web servers, including Apache, Nginx, and IIS.
- Database Integration:
- PHP supports numerous databases, including MySQL, PostgreSQL, SQLite, and MongoDB.
 - It provides built-in functions and extensions (like PDO) for interacting with databases, performing CRUD operations, and managing connections.
- Extensive Library Support:
- PHP has a rich set of built-in functions and a vast repository of extensions and libraries.
 - These libraries cover a wide range of functionalities, from basic string manipulation to complex data encryption.
- Community and Documentation:
- PHP has a large and active community, offering extensive documentation, tutorials, and forums for



Fig (4.6) :-PHP

i.i. Handling User Requests

In a web application, handling user requests is a crucial part of back-end development. This involves processing input data from users, performing server-side operations, and returning appropriate responses. Here are the key aspects:

1. Receiving Requests:

- GET Requests: Used to retrieve data from the server. Commonly used for fetching resources, e.g., displaying a list of patients or fetching details of a specific appointment.
- POST Requests: Used to send data to the server to create or update resources, e.g., registering a new patient, scheduling an appointment, or updating medical records.

2. Processing Requests:

- **Input Validation:** Ensuring that the data received from the user is valid and secure. This includes checking for required fields, proper formats, and sanitizing inputs to prevent security vulnerabilities like SQL injection.
- **Business Logic:** Executing the application's core functionalities based on the request. This might involve calculations, data manipulation, or decisionmaking processes.
- **Database Interactions:** Performing CRUD (Create, Read, Update, Delete) operations on the database to store or retrieve the necessary data. This involves writing SQL queries and using PHP's MySQL functions or PDO (PHP Data Objects) to interact with the database.

3. Sending Responses:

- **HTML/CSS/JavaScript:** Rendering views or templates to send as a response to the client. This might involve embedding dynamic data within HTML pages.
-

Redirects: Redirecting users to different pages based on the outcome of the request, e.g., redirecting to a login page after successful registration.

- o
- o JSON/XML: Returning data in a structured format, especially for API endpoints.

JSON is commonly used due to its simplicity and ease of use with JavaScript.

ii. MySQL for Database Management

MySQL is a popular relational database management system (RDBMS) used for storing and managing data. It is known for its reliability, performance, and ease of use, making it a common choice for web applications, including the Hospital Management System.

☐ Data Retrieval and Manipulation

1. Data Retrieval:

- **SELECT Queries:** The SELECT statement is used to fetch data from one or more tables. It allows you to specify the columns you want to retrieve and apply filters using the WHERE clause.
- **Joins:** Joins are used to combine rows from two or more tables based on a related column between them. Common types of joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
- **Aggregation Functions:** Functions like COUNT(), SUM(), AVG(), MIN(), and MAX() are used to perform calculations on a set of values and return a single value.

2. Data Manipulation:

- **INSERT:** The INSERT statement is used to add new rows to a table. It allows you to specify the columns and values for the new row.
- **UPDATE:** The UPDATE statement is used to modify existing rows in a table. It lets you change the values of one or more columns for rows that meet specific criteria.
- **DELETE:** The DELETE statement is used to remove rows from a table. You can specify which rows to delete using the WHERE clause.
- **Transactions:** Transactions are a sequence of SQL statements that are executed as a single unit. They ensure data integrity by allowing you to commit (save) or roll back (undo) changes based on whether all operations succeed or any fail.

5. User & Admin Modules

i. Patient Registration:

- Purpose: The patient registration module captures and stores basic patient information such as name, age, gender, contact details, and insurance information.
- Features:
 - Input forms for capturing patient details. ◦ Validation of entered data to ensure accuracy.
 - Generation of a unique patient ID for future reference. ◦ Storage of data in the database for easy retrieval and updates.

Fig (5.1):- Patient Login

ii. Appointment Scheduling:

- Purpose: This module allows patients to schedule, reschedule, or cancel appointments with doctors.
- Features:

- o Viewing available time slots for specific doctors.

Booking appointments based on availability. o Sending notifications or reminders to patients and doctors about upcoming appointments.

- o Updating appointment status (e.g., confirmed, canceled, rescheduled).

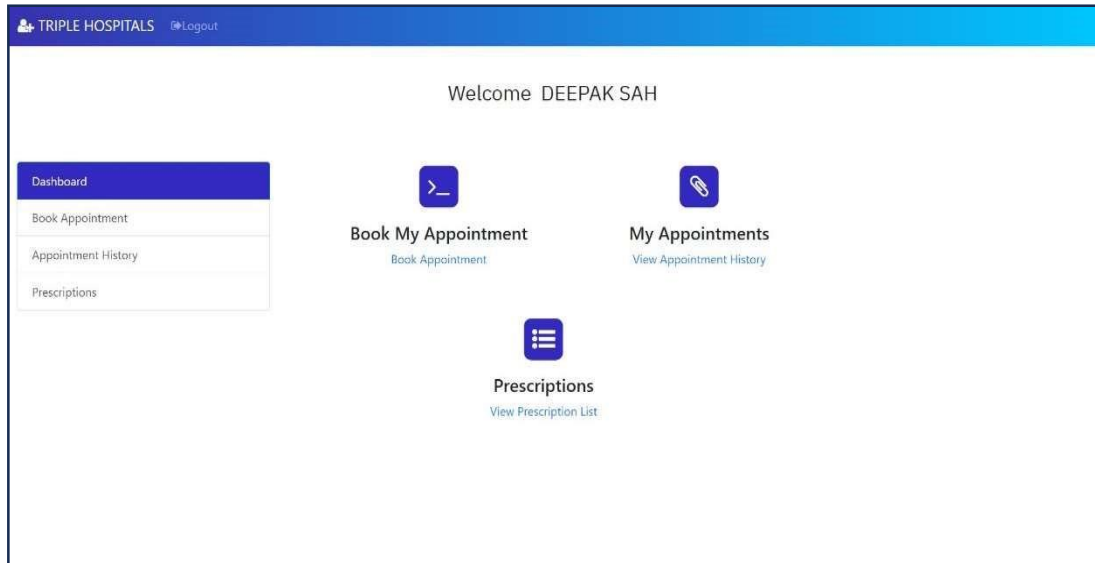
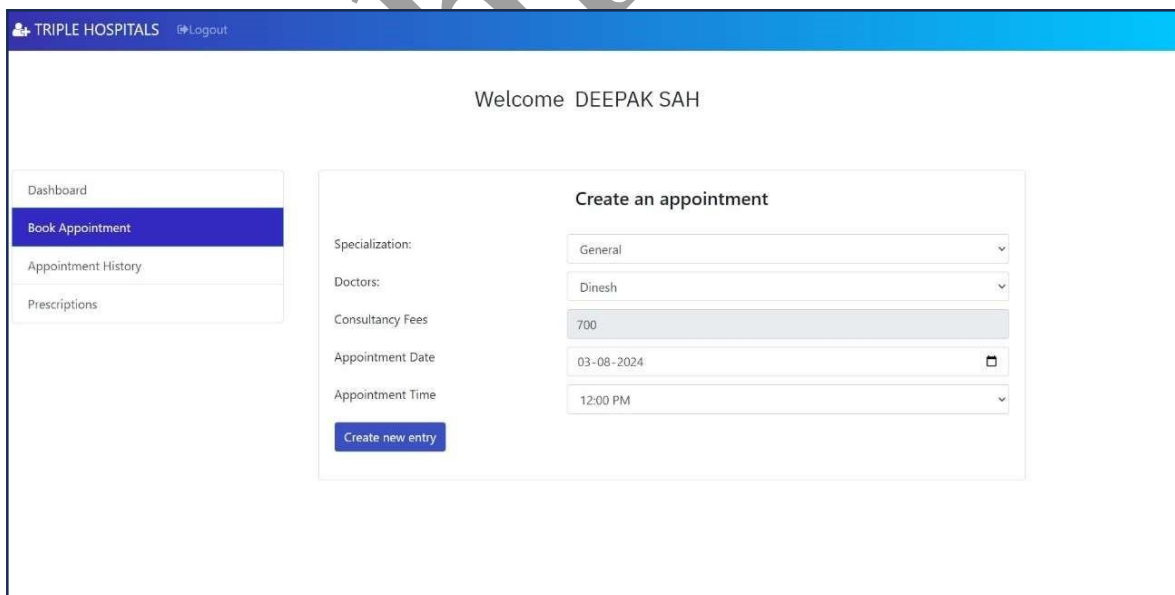


Fig (5.2):- Patient Dashboard



Fig(5.3) :- Patient Appointment

□ Admin Modules

i. User Management:

The User Management module is designed to handle the creation, updating, and deletion of user accounts within the Hospital Management System. It ensures that only authorized personnel have access to specific functionalities based on their roles.

Features:

2. User Creation:

- o Adding New Users: Admins can add new users to the system by inputting their details, such as name, role, contact information, and login credentials.
- o Role Assignment: Assigning roles to new users based on their job functions (e.g., doctor, nurse, receptionist, admin).

3. User Updates:

- o Profile Updates: Admins can update user information such as contact details, job position, and department.
- o Role Changes: Modifying user roles as needed, allowing for changes in job functions or promotions.

4. User Deletion:

- o Removing Users: Admins can deactivate or delete user accounts that are no longer needed, ensuring the system remains secure and up-to-date.
- o Audit Trail: Keeping a log of user deletions for security and compliance purposes.

5. Access Control:

- o Role-Based Access Control (RBAC): Implementing RBAC to ensure that users can only access functionalities relevant to their roles, minimizing security risks and maintaining data integrity.
- o Permissions Management: Defining and managing permissions for different roles to control what actions users can perform within the system.

6. Password Management:

- o Password Policies: Enforcing strong password policies to enhance security, such as requiring a mix of characters and regular password changes.

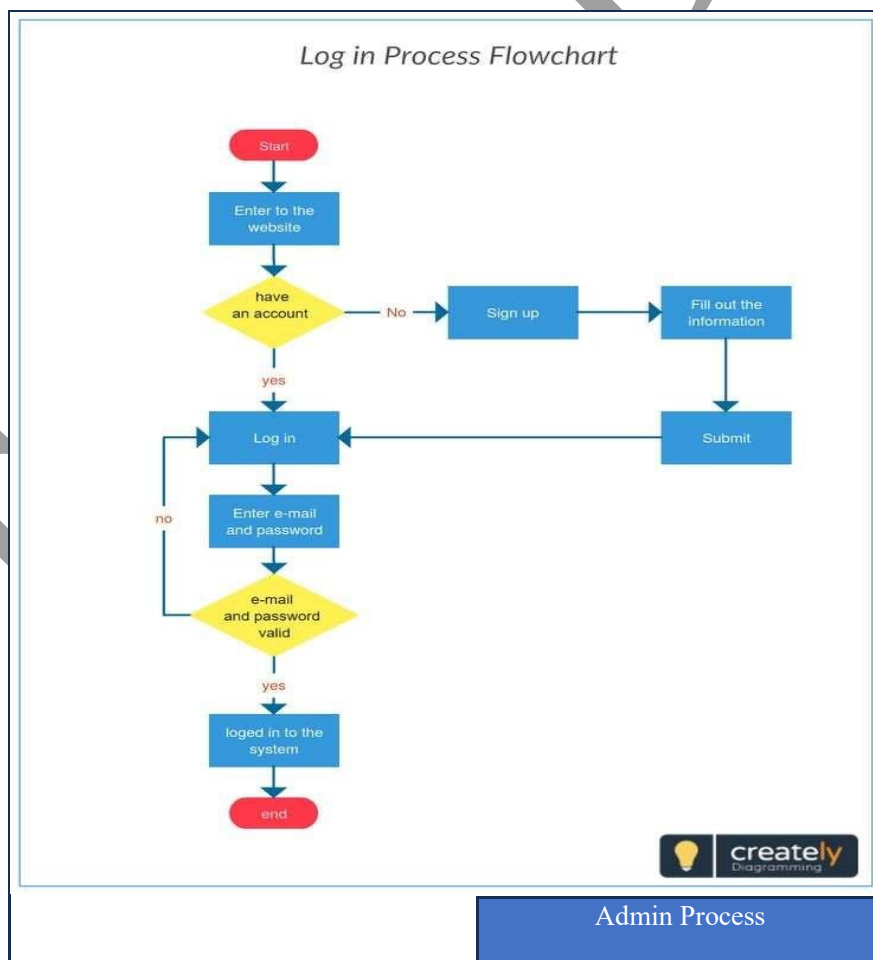
Password Resets: Admins can reset passwords for users who have forgotten their login credentials or whose accounts have been compromised.

7. User Activity Monitoring:

- o Activity Logs: Tracking user activities within the system to monitor for any unauthorized actions or potential security breaches.
- o Usage Reports: Generating reports on user activity to analyze system usage patterns and identify areas for improvement.

8. User Support:

- o Helpdesk Integration: Providing support for users who encounter issues with their accounts, such as login problems or access restrictions.
- o Training and Onboarding: Offering training and onboarding sessions for new users to ensure they understand how to use the system effectively.



FlowChart(5.4):- Admin Panel

6. Reporting

□ Generating Reports with TCPDF

Generating reports is a crucial function in a Hospital Management System, providing essential insights and documentation for various operational and administrative needs. TCPDF is a popular PHP library used to create PDF documents directly from PHP scripts, facilitating the generation of detailed and professional reports.

Features of TCPDF:

1. Customizable Layouts:

- Design Flexibility: Allows customization of report layouts, including headers, footers, fonts, colors, and styles to match the hospital's branding and reporting standards.
- Page Formatting: Supports different page formats and orientations (portrait and landscape), enabling the creation of diverse report types.

2. Complex Data Representation:

- Tables and Lists: Facilitates the creation of complex tables and lists to organize and display data systematically. This is particularly useful for detailed financial reports, patient records, and inventory lists.
- Multi-Page Reports: Handles multi-page documents efficiently, ensuring that long reports are well-structured and readable.

3. Automated Report Generation:

- Scheduled Reports: Enables the automation of report generation at scheduled intervals (e.g., daily, weekly, monthly), ensuring timely availability of critical information.

4. Security and Access Control:

- Password Protection: Allows setting passwords for PDF documents to restrict unauthorized access and protect sensitive information.
- Permissions: Configures permissions to control actions like printing, copying, and modifying the reports.

- Patient Reports: Generating detailed patient reports, including medical history, treatment plans, and test results, for consultations and record-keeping.

7. Security& Maintenance Support

☐ Input Validation and Sanitization:

Ensuring the security of a Hospital Management System involves rigorous input validation and sanitization to protect against malicious data. Input validation is the process of checking incoming data to ensure it meets predefined criteria before processing. This is crucial for preventing security threats such as SQL injection, crosssite scripting (XSS), and other types of attacks that exploit unvalidated inputs.

Sanitization involves cleaning the input data by removing or escaping any potentially harmful characters. This step is essential for ensuring that even if malicious data is entered, it cannot cause harm when processed or stored. Effective input validation and sanitization safeguard the system by ensuring that all data entering the system is clean, expected, and secure, thereby maintaining data integrity and preventing security breaches.

☐ User Authentication and Authorization:

User authentication and authorization are critical components of a secure Hospital Management System. Authentication is the process of verifying the identity of a user attempting to access the system. This typically involves the use of credentials such as usernames and passwords. Advanced systems may also employ multi-factor authentication (MFA), which requires additional verification steps, such as a code sent to a user's mobile device or biometric verification.

Authorization, on the other hand, determines what authenticated users are allowed to do within the system. It involves setting permissions and access controls based on user roles. For instance, doctors, nurses, and administrative staff will have different access levels and permissions, ensuring that users can only access the data and functionalities relevant to their roles. This prevents unauthorized access to sensitive patient information and critical system functions.

Together, input validation and sanitization, along with robust user authentication and authorization mechanisms, form the backbone of a secure Hospital Management System.



Fig(7.1):-Security

□ Contact Support:

In a Hospital Management System, contact support is a crucial aspect of ensuring that users receive timely assistance with any issues or questions they encounter. This feature is designed to provide a streamlined process for users to report problems, seek help, and get their concerns addressed efficiently.

Aspects of Contact Support:

1. Support Channels:

- o Email: Users can send detailed support requests or issues via email to a dedicated support address. This method is suitable for handling complex problems that require detailed explanations and documentation.
- o Phone Support: Offering a phone support line allows users to receive immediate assistance for urgent issues. This is ideal for troubleshooting realtime problems and providing hands-on guidance.

2. Knowledge Base:

- o Documentation: Providing a comprehensive knowledge base with articles, FAQs, and troubleshooting guides helps users find solutions to common problems independently, reducing the need for direct support.
- o Self-Service: Encouraging self-service options through an organized and searchable knowledge base empowers users to resolve minor issues on their own.

3. Feedback and Improvement:

- o User Feedback: Collecting feedback from users about their support experience helps identify areas for improvement and ensures that the support process continually evolves to meet user needs.



Fig(7.2):-Maintenance

8. Result

The screenshot shows the 'Register as Patient' form on the TRIPLE HOSPITAL website. The header includes the hospital logo, name, and navigation links (HOME, ABOUT US, CONTACT). The form has tabs for 'Patient', 'Doctor', and 'Receptionist', with 'Patient' selected. The registration fields include: First Name *, Last Name *, Your Email *, Your Phone *, Password *, and Confirm Password *. There are radio buttons for 'Male' (selected) and 'Female', and a link for 'Already have an account?'. A 'Register' button is at the bottom right.

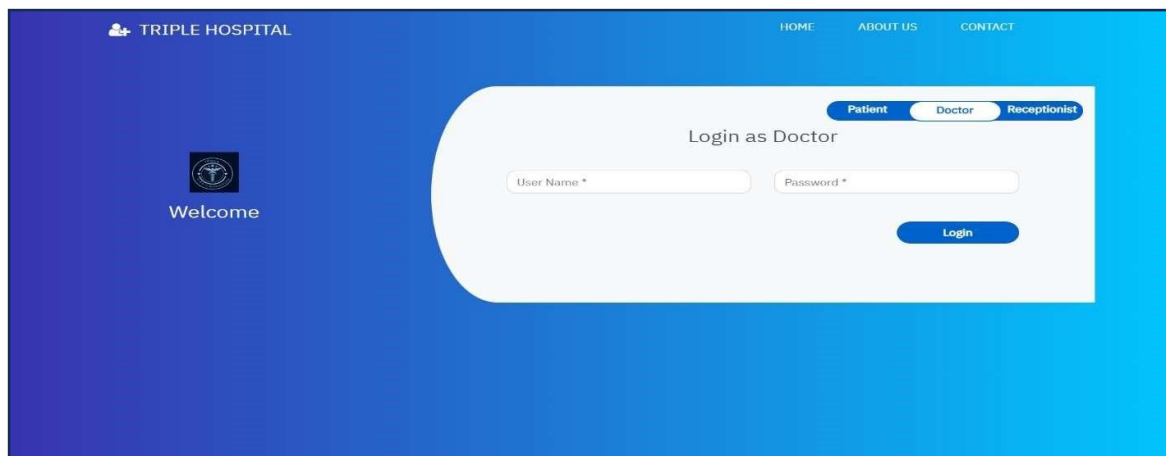
Fig (8.1) :- Patient Registration

The screenshot shows the Patient Dashboard for a user named DEEPAK SAH. The header includes the hospital name and a 'Logout' link. A sidebar menu on the left contains 'Dashboard', 'Book Appointment', 'Appointment History', and 'Prescriptions'. The main area features three large cards: 'Book My Appointment' (with a calendar icon and 'Book Appointment' link), 'My Appointments' (with a link icon and 'View Appointment History' link), and 'Prescriptions' (with a list icon and 'View Prescription List' link).

Fig(8.2) :- Patient Dashboard

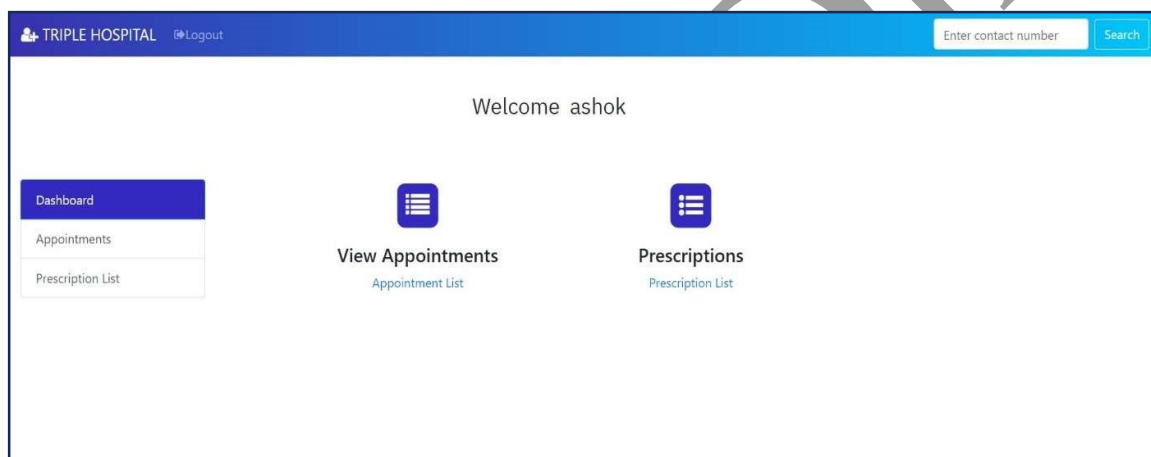
The screenshot shows the 'Create an appointment' form on the TRIPLE HOSPITALS website. The header includes the hospital name and a 'Logout' link. The user is logged in as DEEPAK SAH. A sidebar menu on the left contains 'Dashboard', 'Book Appointment' (highlighted), 'Appointment History', and 'Prescriptions'. The main area has a 'Create an appointment' form with the following fields: Specialization (dropdown menu showing 'General'), Doctors (dropdown menu showing 'Dinesh'), Consultancy Fees (text input showing '700'), Appointment Date (calendar icon showing '03-08-2024'), and Appointment Time (dropdown menu showing '12:00 PM'). A 'Create new entry' button is at the bottom left of the form.

Fig(8.3) :- Patient Booking Appointment



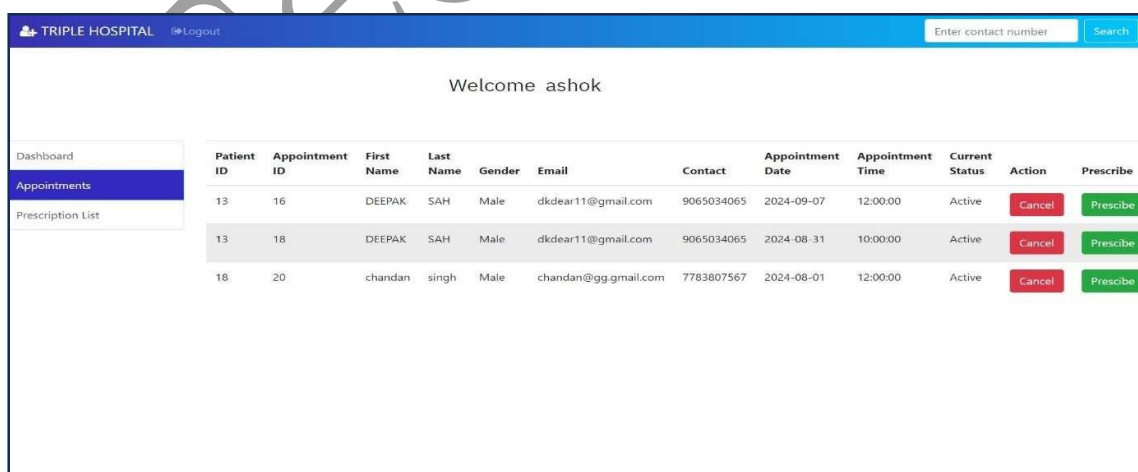
The image shows the 'Login as Doctor' page of the Triple Hospital system. It features a blue header with the hospital name and navigation links. A sidebar on the left contains a 'Welcome' message and a medical icon. The main content area has a login form with tabs for 'Patient', 'Doctor', and 'Receptionist'. The 'Doctor' tab is selected, showing fields for 'User Name *' and 'Password *', and a 'Login' button.

Fig (8.4) :-Doctor Login Page



The image shows the Doctor Dashboard for a user named 'ashok'. It includes a blue header with the hospital name, a 'Logout' link, and a search bar. The dashboard is divided into three main sections: a left sidebar with 'Dashboard', 'Appointments', and 'Prescription List'; a central area with 'View Appointments' and 'Prescriptions' buttons; and a right area with a search bar.

Fig(8.5):- Doctor Dashboard



The image shows the 'Appointments' section of the Doctor Dashboard. It displays a table of appointments with columns for Patient ID, Appointment ID, First Name, Last Name, Gender, Email, Contact, Appointment Date, Appointment Time, Current Status, Action, and Prescribe. The table contains three rows of data.

Patient ID	Appointment ID	First Name	Last Name	Gender	Email	Contact	Appointment Date	Appointment Time	Current Status	Action	Prescribe
13	16	DEEPAK	SAH	Male	dkdear11@gmail.com	9065034065	2024-09-07	12:00:00	Active	Cancel	Prescribe
13	18	DEEPAK	SAH	Male	dkdear11@gmail.com	9065034065	2024-08-31	10:00:00	Active	Cancel	Prescribe
18	20	chandani	singh	Male	chandani@gg@gmail.com	7783807567	2024-08-01	12:00:00	Active	Cancel	Prescribe

Fig(8.6):-Doctor Appointment List

TRIPLE HOSPITAL
Logout

Welcome ashok

Dashboard
Appointments
Prescription List

Patient ID	First Name	Last Name	Appointment ID	Appointment Date	Appointment Time	Disease	Allergy	Prescribe
13	DEEPAK	SAH	16	2024-09-07	12:00:00	Normal Cough	Cold Water	Cold water is not drink
13	DEEPAK	SAH	18	2024-08-31	10:00:00	Normal Fever	Take rest for two days	xyz medicine twice a day

Fig(8.7) :- Doctor Prescription List

TRIPLE HOSPITALS
Logout

Welcome DEEPAK SAH

Dashboard
Book Appointment
Appointment History
Prescriptions

Doctor Name	Consultancy Fees	Appointment Date	Appointment Time	Current Status	Action
ashok	500	2024-09-07	12:00:00	Active	<button>Cancel</button>
Ganesh	550	2024-09-08	10:00:00	Active	<button>Cancel</button>
ashok	500	2024-08-31	10:00:00	Active	<button>Cancel</button>
arun	600	2024-08-01	12:00:00	Active	<button>Cancel</button>
Dinesh	700	2024-08-03	12:00:00	Active	<button>Cancel</button>

Fig(8.8):- Patient Appointment History

TRIPLE HOSPITALS
Logout

Welcome DEEPAK SAH

Dashboard
Book Appointment
Appointment History
Prescriptions

Doctor Name	Appointment ID	Appointment Date	Appointment Time	Diseases	Allergies	Prescriptions	Bill Payment
ashok	16	2024-09-07	12:00:00	Normal Cough	Cold Water	Cold water is not drink	<button>Pay Bill</button>
ashok	18	2024-08-31	10:00:00	Normal Fever	Take rest for two days	xyz medicine twice a day	<button>Pay Bill</button>
arun	23	2024-08-01	12:00:00	Cough	Cold drink	xyz Medicine take	<button>Pay Bill</button>

Fig(8.9) :- Patient Prescription

The screenshot shows a web application interface for generating a patient bill. At the top, there is a header bar with a menu icon, the text 'Generate Bill', and a toolbar with icons for zooming and printing. The main content area is titled 'TRIPLE HOSPITALS' and 'Bill'. It contains a list of patient and appointment details.

TRIPLE HOSPITALS	
Bill	
Patient ID :	13
Appointment ID :	16
Patient Name :	DEEPAK SAH
Doctor Name :	ashok
Appointment Date :	2024-09-07
Appointment Time :	12:00:00
Disease :	Normal Cough
Allergies :	Cold Water
Prescription :	Cold water is not drink
Fees Paid :	500

Fig(8.1.1):- Patient Bill

The screenshot shows the login page of the Triple Hospital system. The page has a blue gradient background. On the left, there is a logo and the text 'Welcome'. On the right, there is a login form titled 'Login as Admin'. The form includes tabs for 'Patient', 'Doctor', and 'Receptionist'. Below the tabs, there are input fields for 'User Name *' and 'Password *', and a 'Login' button.

TRIPLE HOSPITAL

HOME ABOUT US CONTACT

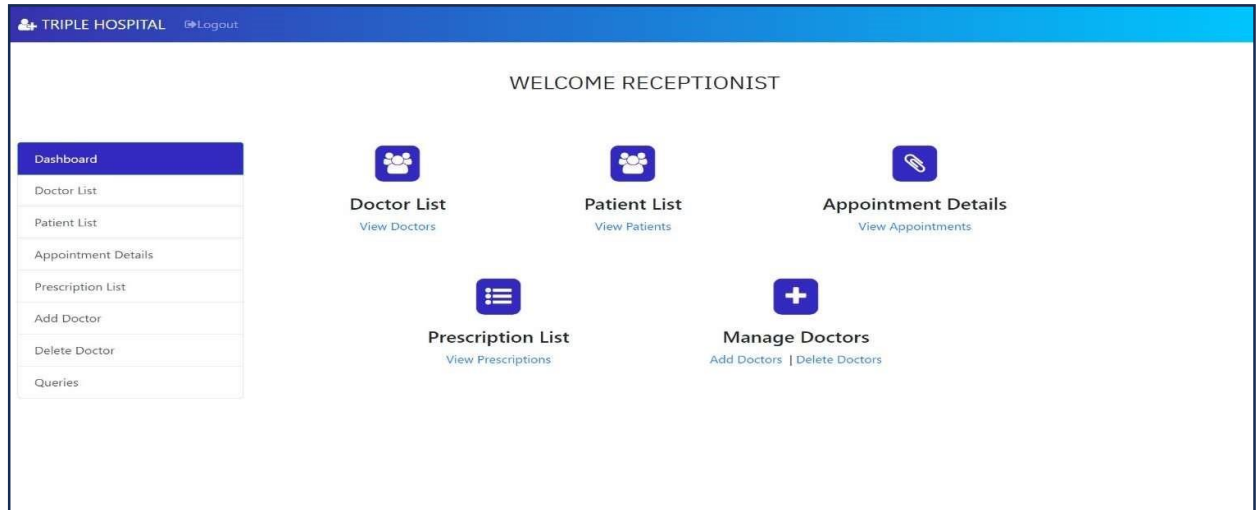
Patient Doctor Receptionist

Login as Admin

User Name * Password *

Login

Fig(8.1.2) :-Receptionist(Admin)



Fig(8.1.3) :-Admin Dashboard

9. Conclusion

□ Summary of the Project:

The Hospital Management System project aims to streamline and modernize hospital operations through an integrated software solution. By leveraging technologies such as PHP, MySQL, HTML5, CSS3, JavaScript, and TCPDF, the system effectively manages various aspects of hospital administration. Key modules include patient registration, appointment scheduling, electronic medical records (EMR), billing, and inventory management. Additionally, the system supports administrative functions such as user management, role-based access control, and comprehensive reporting. The implementation of robust security measures ensures data protection and integrity, while the contact support module provides timely assistance to users. Overall, the project enhances operational efficiency, improves patient care, and ensures a secure and user-friendly environment for hospital management.

□ Future Enhancements:

As the Hospital Management System evolves, several potential enhancements could be considered to further improve its functionality and user experience. Integrating advanced analytics and artificial intelligence (AI) could provide deeper insights into patient data and operational performance, enabling more informed decision-making. Enhancing mobile access through dedicated apps or responsive design improvements could increase user convenience and accessibility. Expanding interoperability with other healthcare systems and electronic health record (EHR) platforms would facilitate better data exchange and collaboration. Additionally, incorporating telemedicine capabilities could support remote consultations and broaden service delivery. Continual updates and feature additions based on user feedback and technological advancements will ensure the system remains at the forefront of healthcare management solutions.



10.RefeRences

<https://www.php.net/docs.php> <https://dev.mysql.com/doc/>

https://www.w3schools.com/html/html5_intro.asp

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

<https://www.apachefriends.org/index.html>

<https://tcpdf.org/docs/>

<https://www.himss.org/>

<https://www.master-care.com.au/patient-administration-system/>

<https://github.com/TechnoDk>

<https://github.com/TechnoDk?tab=repositories>

<https://github.com/TechnoDk?tab=projects>



Given By:-

Deepak Kumar sah

Mail:-dkdear11@gmail.com