HOSPITAL MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

SAMEER D	**220701242**
SHAGHABETH HUSSAIN M	**220701256**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2023 – 24

BONAFIDE CERTIFICATE

Certified that this project report **" HOSPITAL MANAGEMENT SYSTEM "** is the bonafide work of

" SAMEER D (220701242) , SHAGHABETH HUSSAIN M (220701256) "

who carried out the project work under my supervision**.**

Submitted for the Practical Examination held on_____

SIGNATURE

MRS. K. MAHESMEENA

Assistant Professor,

Computer Science and Engineering,

Rajalakshmi Engineering College,

(Autonomous),

Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Hospital Management System (HMS) project, developed using PHP and MySQL, aims to digitize the administrative and medical processes of a hospital, thereby increasing efficiency, accuracy, and ease of access to vital information. This system categorizes users into three primary roles: Admin, Doctor, and Patient. Each role is provided with a distinct set of functionalities tailored to their needs and responsibilities within the hospital ecosystem.

For patients, the HMS facilitates the booking of appointments by selecting a doctor based on specialization, date, and time, while also displaying the associated fees. Patients can view their appointment history, prescription details, and have the ability to cancel appointments if needed. This not only simplifies the patient experience but also reduces administrative workload.

Doctors, on the other hand, can manage their appointments by approving or rejecting patient requests. Once an appointment is approved, doctors can record details such as diagnosis, prescribed medications, and any allergies. This data is then accessible to the patients, ensuring clear communication and effective treatment plans. Doctors can also review their appointment and prescription histories, which aids in better patient management and follow-up care.

The Admin panel serves as the backbone of the HMS, providing comprehensive control over the system. Admins can manage doctor accounts, view all patient and doctor records, and oversee appointments and prescriptions. This role includes functionalities for searching and filtering records, ensuring that information retrieval is both quick and efficient. The admin can also handle contact queries, thereby maintaining an open channel for communication within the hospital.

The HMS project integrates a clean and intuitive user interface, developed using the Bootstrap framework along with custom CSS, ensuring a seamless user experience across different devices. The backend, powered by PHP and SQL, guarantees robust data management and secure operations.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION

In the rapidly evolving field of healthcare, the efficient management of hospital operations is paramount. The Hospital Management System (HMS) is a web-based application developed to streamline and digitize the myriad administrative and clinical tasks that occur within a hospital setting. Utilizing technologies such as PHP for server-side scripting and MySQL for database management, this system ensures that critical information is both readily accessible and securely managed. The HMS is designed to cater to three primary user roles—Admin, Doctor, and Patient—each with specific functionalities to address their unique needs. For patients, it simplifies the process of booking appointments, viewing medical histories, and accessing prescriptions, while doctors can efficiently manage their schedules, record patient interactions, and prescribe medications. Admins are provided with comprehensive control over the system, enabling them to manage user accounts, oversee operations, and ensure the smooth functioning of the hospital.

The HMS not only aims to enhance the efficiency of hospital operations but also to improve the quality of patient care. By automating routine tasks and providing a user-friendly interface, the system reduces the administrative burden on hospital staff and minimizes the likelihood of errors. Patients benefit from a more organized and transparent healthcare experience, with easy access to their medical records and appointment details. Doctors can better manage their workload and focus on providing high-quality care. The system's robust backend, powered by SQL, ensures reliable data storage and retrieval, while the responsive frontend, developed using Bootstrap and custom CSS, guarantees an optimal user experience across various devices. Overall, the Hospital Management System represents a significant step towards the modernization of healthcare facilities, leveraging technology to foster a more efficient, patient-centered approach to healthcare delivery.

1.2 OBJECTIVES

- To automate and streamline hospital management processes.
- To maintain a comprehensive database of patient and doctor records.
- To enable easy appointment booking and management.
- To facilitate efficient prescription and billing processes.
- To provide secure access and control to admin, doctors, and patients.

1.3 MODULES

- **Admin Module**: This module provides administrators with full control over the system, including managing doctor and patient accounts, viewing and filtering appointments, prescriptions, and handling contact queries.

- **Doctor Module**: Doctors use this module to manage their appointments, approve or reject patient requests, and record diagnosis details and prescriptions, which are then accessible to patients.

- **Patient Module**: Patients can book appointments, view their medical history and prescriptions, and manage their personal information through this module.

2. SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION

The Hospital Management System (HMS) project employs a range of software technologies to create a robust, efficient, and user-friendly web application. The system is primarily developed using PHP, a popular server-side scripting language that facilitates dynamic content generation and seamless interaction with databases. PHP is chosen for its flexibility, ease of use, and strong community support, making it ideal for developing complex web applications like the HMS. The backend database is managed using MySQL, a reliable and widely-used relational database management system (RDBMS). MySQL ensures efficient data storage, retrieval, and management, which is crucial for handling the large volumes of data generated by hospital operations.

2.2 LANGUAGES

2.2.1 SQL

Structured Query Language (SQL) is employed for database operations within the HMS. SQL is a standardized language used to manage relational databases and perform various operations on the data, such as querying, updating, and managing database schema creation and modifications. The use of SQL ensures that data is efficiently stored, organized, and accessed, which is critical for maintaining accurate and up-to-date hospital records.

2.2.2 PHP

PHP (Hypertext Preprocessor) is the core programming language used for the server-side scripting of the HMS. PHP's ability to embed within HTML, its compatibility with various databases, and its extensive set of built-in functions make it a powerful tool for developing web applications. In the HMS, PHP handles the logic behind user interactions, data processing, and communication with the MySQL database, ensuring that the system is responsive and interactive.

3. REQUIREMENTS AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION

The Hospital Management System (HMS) aims to streamline hospital operations and improve patient care through a web-based application. The requirements for the HMS can be categorized into functional and non-functional requirements.

3.1.1 Functional Requirements:

1. **User Authentication and Authorization**:
   - Admin, Doctor, and Patient login and registration functionalities.
   - Role-based access control to ensure that users can only access functionalities relevant to their roles.
2. **Appointment Management**:
   - Patients can book, view, and cancel appointments.
   - Doctors can approve or reject appointment requests.
   - Admin can oversee all appointments.
3. **Patient Management**:
   - Patients can manage personal information, view medical history, and access prescriptions.
   - Admin can view and manage patient records.
4. **Doctor Management**:
   - Doctors can manage their profiles, view patient details, and record diagnosis and prescriptions.
   - Admin can add, update, or remove doctor accounts.
5. **Prescription Management**:
   - Doctors can create and update prescriptions for patients.
   - Patients can view their prescriptions.
6. **Search and Filter**:
   - Admin can search and filter records for patients, doctors, and appointments.
7. **Contact Management**:
   - Admin can view and manage contact queries from users.

3.1.2 Non-Functional Requirements:

1. **Usability**:
    - The system should have an intuitive user interface for easy navigation.
    - The system should be accessible on various devices through a responsive design.
2. **Performance**:
    - The system should handle concurrent users efficiently.
    - The system should have quick response times for database queries.
3. **Security**:
    - Data encryption for sensitive information.
    - Secure authentication mechanisms to prevent unauthorized access.
4. **Scalability**:
    - The system should be scalable to handle a growing number of users and data.
5. **Reliability**:
    - The system should ensure data integrity and availability.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 Hardware Requirements:

- **Server**:
    - Processor: Quad-Core CPU
    - RAM: 8 GB or higher
    - Storage: 500 GB SSD or higher
    - Network: High-speed internet connection
- **Client Devices**:
    - Any device capable of running a modern web browser (desktop, laptop, tablet, or smartphone)
    - Screen resolution: 1024x768 or higher

3.2.2 Software Requirements:

- **Server**:
    - Operating System: Linux (preferred), Windows Server, or macOS
    - Web Server: Apache HTTP Server or Nginx
    - Database Server: MySQL or MariaDB
    - Scripting Language: PHP 7.4 or higher
- **Client Devices**:

- Web Browser: Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari

3.3 ARCHITECTURE DIAGRAM

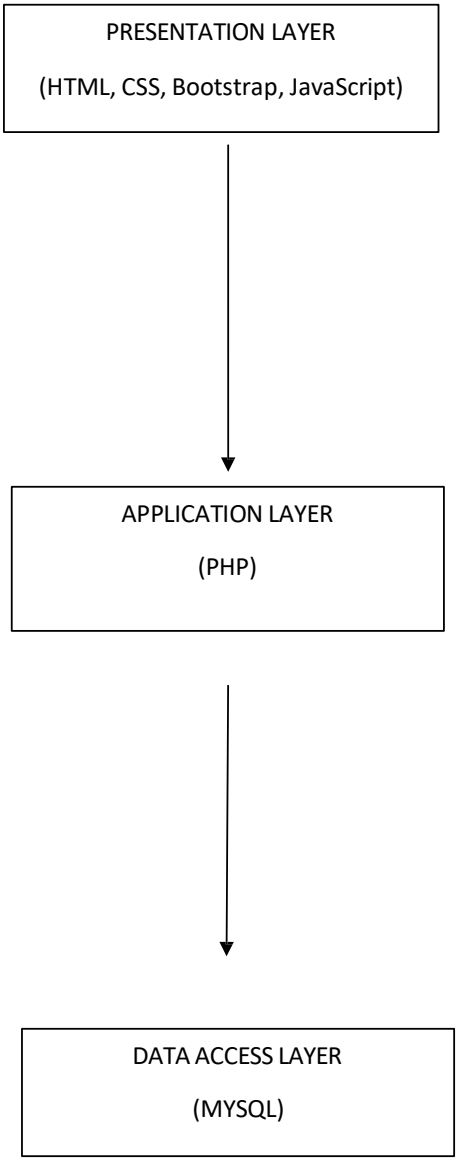

Fig 3.3.1 – Architecture Diagram

The HMS architecture follows a multi-tier architecture model, typically consisting of the following layers:

1. **Presentation Layer**:
    - Responsible for the user interface.
    - Technologies: HTML, CSS, Bootstrap, JavaScript

2. **Application Layer**:
    - Handles the business logic of the application.
    - Technologies: PHP

3. **Data Layer**:
    - Manages data storage and retrieval.
    - Technologies: MySQL

3.4 ER DIAGRAM

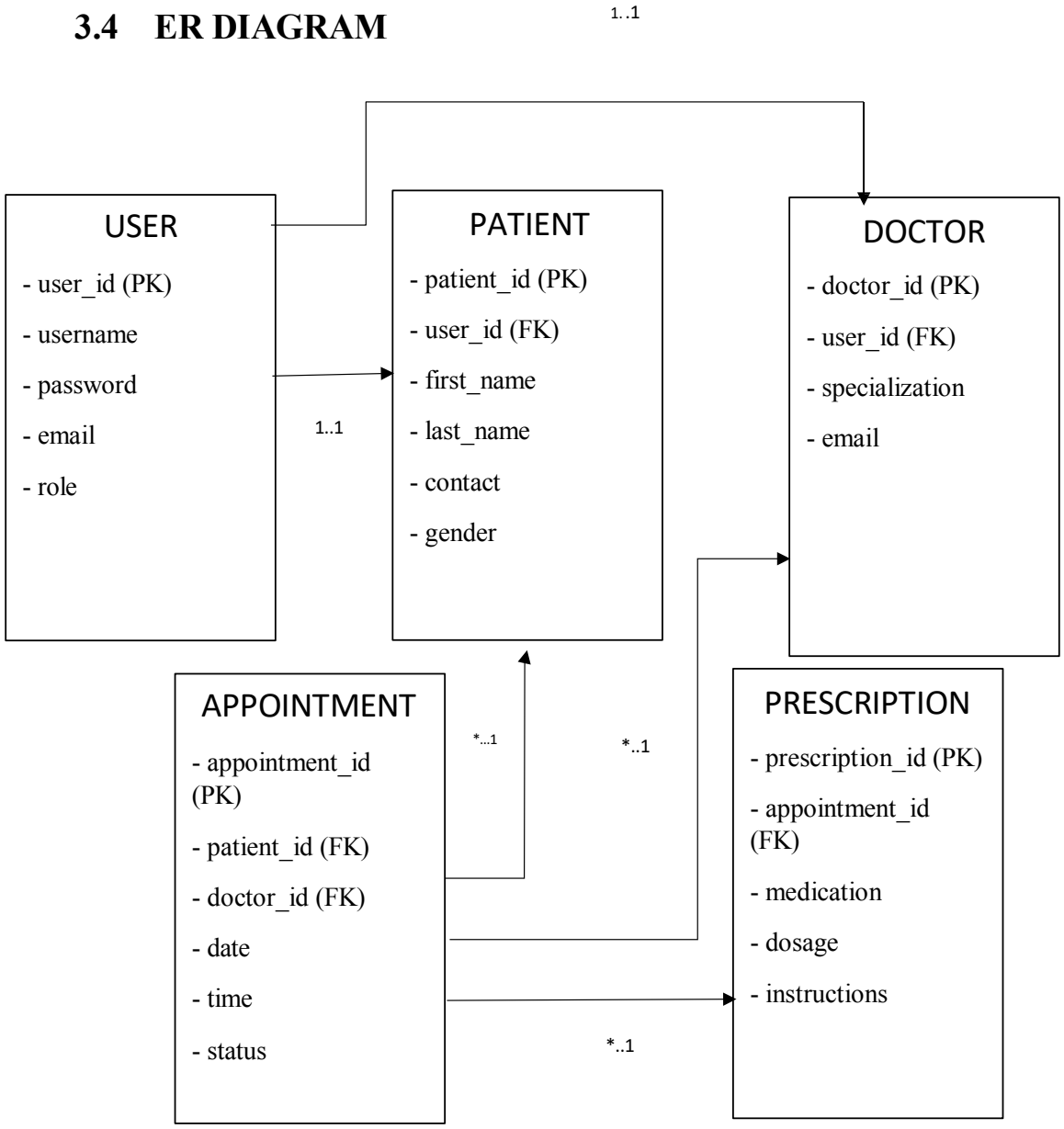

Fig 3.4.1 – ER Diagram

4. PROGRAM CODE

```
<html>
<head>
    <title>HMS</title>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.png"
/>
<link rel="stylesheet" type="text/css" href="style1.css">
<link
href="https://fonts.googleapis.com/css?family=IBM+Plex+Sans&display=swap"
rel="stylesheet">
<!-- <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css" integrity="sha384-
HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l9a69psu"
crossorigin="anonymous"> -->

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.
css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
<link
href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<style >
     .form-control {
    border-radius: 0.75rem;
}
</style>

<script>
    var check = function() {
  if (document.getElementById('password').value ==
    document.getElementById('cpassword').value) {
    document.getElementById('message').style.color = '#5dd05d';
    document.getElementById('message').innerHTML = 'Matched';
  } else {
    document.getElementById('message').style.color = '#f55252';
    document.getElementById('message').innerHTML = 'Password fields doesnot
match';
  }
}

function alphaOnly(event) {
```

```
  var key = event.keyCode;
  return ((key >= 65 && key <= 90) || key == 8 || key == 32);
};

function checklen()
{
    var pass1 = document.getElementById("password");
    if(pass1.value.length<6){
        alert("Password must be at least 6 characters long. Try again!");
        return false;
  }
}

</script>

</head>

<!------ Include the above in your HEAD tag---------- >
<body>
<nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav" >
    <div class="container">

      <a class="navbar-brand js-scroll-trigger" href="#" style="margin-top:
10px;margin-left:-65px;font-family: 'IBM Plex Sans', sans-serif;"><h4><i
class="fa fa-hospital-o" aria-hidden="true"></i>  HOSPITAL MANAGEMENT
SYSTEM</h4></a>
      <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item" style="margin-right: 40px;">
            <a class="nav-link js-scroll-trigger" href="index.php"
style="color: white;font-family: 'IBM Plex Sans', sans-
serif;"><h6>HOME</h6></a>
          </li>


          <li class="nav-item">
            <a class="nav-link js-scroll-trigger" href="contact.html"
style="color: white;font-family: 'IBM Plex Sans', sans-
serif;"><h6>CONTACT</h6></a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

```
<div class="container register" style="font-family: 'IBM Plex Sans', sans-
serif;">
                <div class="row">
                    <div class="col-md-3 register-left" style="margin-top:
10%;right: 5%">
                        <img
 src="https://image.ibb.co/n7oTvU/logo_white.png"  alt=""/>
                        <h3>Welcome</h3>

                    </div>
                    <div class="col-md-9 register-right" style="margin-top:
40px;left: 80px;">
                        <ul class="nav nav-tabs nav-justified" id="myTab"
role="tablist" style="width: 40%;">
                            <li class="nav-item">
                                <a class="nav-link active" id="home-tab"
data-toggle="tab" href="#home" role="tab" aria-controls="home" aria-
selected="true">Patient</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" id="profile-tab" data-
toggle="tab" href="#profile" role="tab" aria-controls="profile" aria-
selected="false">Doctor</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link" id="profile-tab" data-
toggle="tab" href="#admin" role="tab" aria-controls="admin" aria-
selected="false">Admin</a>
                            </li>
                        </ul>
                        <div class="tab-content" id="myTabContent">
                            <div class="tab-pane fade show active"
id="home" role="tabpanel" aria-labelledby="home-tab">
                                <h3 class="register-heading">Register as
Patient</h3>
                                <form method="post" action="func2.php">
                                <div class="row register-form">

                                    <div class="col-md-6">
                                        <div class="form-group">
                                            <input type="text" class="form-
control"  placeholder="First Name *" name="fname"  onkeydown="return
alphaOnly(event);" required/>
                                        </div>
                                        <div class="form-group">
```

```
                                        <input type="email"
class="form-control" placeholder="Your Email *" name="email"  />
                                    </div>
                                    <div class="form-group">
                                        <input type="password"
class="form-control" placeholder="Password *" id="password" name="password"
onkeyup='check();' required/>
                                    </div>

                                    <div class="form-group">
                                        <div class="maxl">
                                            <label class="radio
inline">

                                                <input type="radio"
name="gender" value="Male" checked>

                                                <span> Male </span>
                                            </label>
                                            <label class="radio
inline">

                                                <input type="radio"
name="gender" value="Female">

                                                <span>Female </span>
                                            </label>
                                        </div>
                                        <a href="index1.php">Already
have an account? Login Now</a>
                                    </div>
                                </div>

                                <div class="col-md-6">
                                    <div class="form-group">
                                        <input type="text" class="form-
control" placeholder="Last Name *" name="lname" onkeydown="return
alphaOnly(event);" required/>
                                    </div>

                                    <div class="form-group">
                                        <input type="tel"
minlength="10" maxlength="10" name="contact" class="form-control"
placeholder="Contact *"  />
                                    </div>
                                    <div class="form-group">
                                        <input type="password"
class="form-control"  id="cpassword" placeholder="Confirm Password *"
name="cpassword"  onkeyup='check();' required/><span id='message'></span>
                                    </div>
```

```
                                    <input type="submit"
class="btnRegister" name="patsub1" onclick="return checklen();"
value="Register"/>
                                </div>

                            </div>
                        </form>
                        </div>


                        <div class="tab-pane fade show" id="profile"
role="tabpanel" aria-labelledby="profile-tab">
                            <h3  class="register-heading">Login as
Doctor</h3>
                            <form method="post" action="func1.php">
                            <div class="row register-form">
                                <div class="col-md-6">
                                    <div class="form-group">
                                        <input type="text" class="form-
control" placeholder="User Name *" name="username3" onkeydown="return
alphaOnly(event);" required/>
                                    </div>
                                </div>
                                <div class="col-md-6">
                                    <div class="form-group">
                                        <input type="password"
class="form-control" placeholder="Password *" name="password3" required/>
                                    </div>

                                    <input type="submit"
class="btnRegister" name="docsub1" value="Login"/>
                                </div>
                            </div>
                        </form>
                        </div>


                        <div class="tab-pane fade show" id="admin"
role="tabpanel" aria-labelledby="profile-tab">
                            <h3  class="register-heading">Login as
Admin</h3>
                            <form method="post" action="func3.php">
                            <div class="row register-form">
                                <div class="col-md-6">
                                    <div class="form-group">
                                        <input type="text" class="form-
control" placeholder="User Name *" name="username1" onkeydown="return
alphaOnly(event);" required/>
                                    </div>
```

```
                                    </div>
                                    <div class="col-md-6">
                                        <div class="form-group">
                                            <input type="password"
class="form-control" placeholder="Password *" name="password2" required/>
                                        </div>

                                        <input type="submit"
class="btnRegister" name="adsub" value="Login"/>
                                    </div>
                                </div>
                            </form>
                            </div>
                        </div>

                    </div>
                </div>

            </div>
    </body>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min
.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js
" integrity="sha384-
aJ21OjlMXNL5UyIl/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd"
crossorigin="anonymous"></script>
    </html>
```

5. RESULTS AND DISCUSSION

5.1 Functionality of the Project

The Hospital Management System (HMS) project is designed to streamline hospital operations by managing patient records, doctor appointments, and prescriptions through a web-based application. The primary functionalities include:

1. **User Authentication and Authorization**:

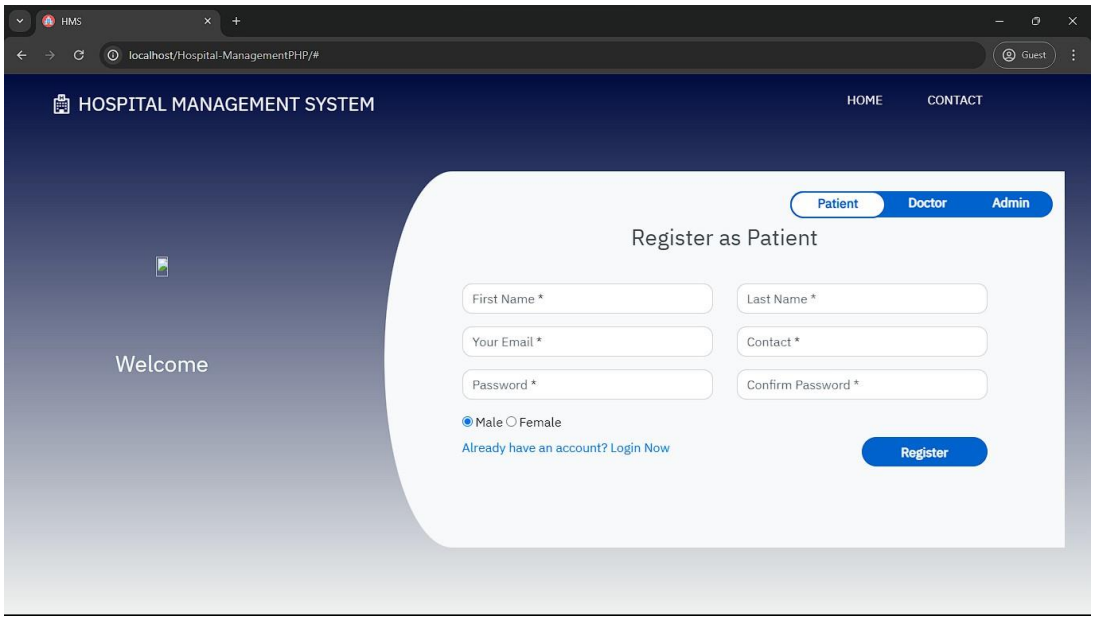

Fig 5.1.1 – Patient Login

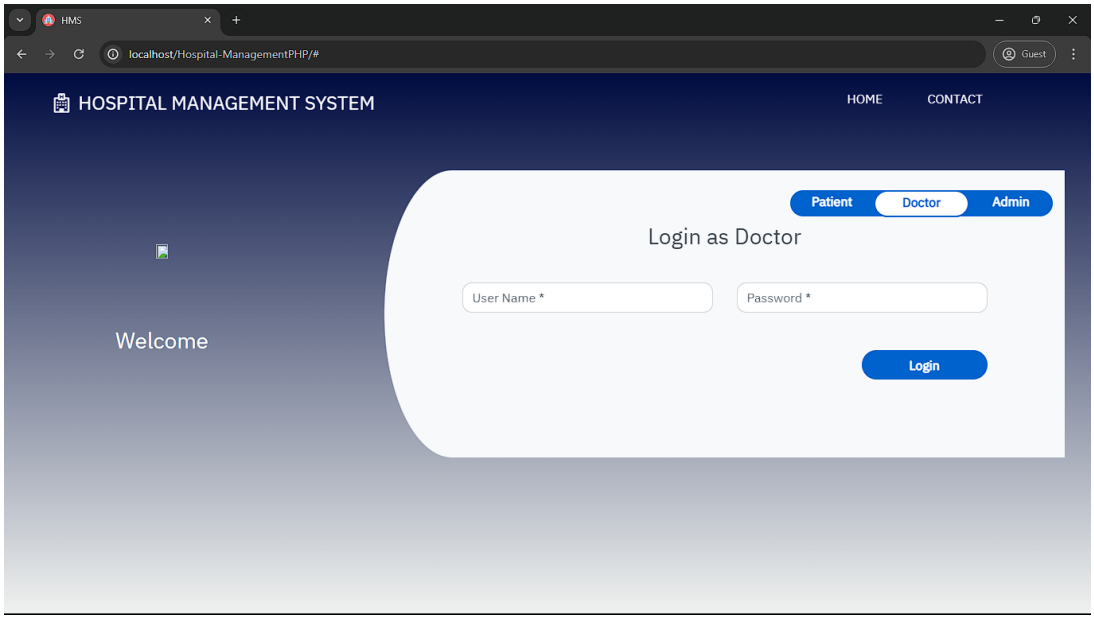

Fig 5.1.2 – Doctor Login

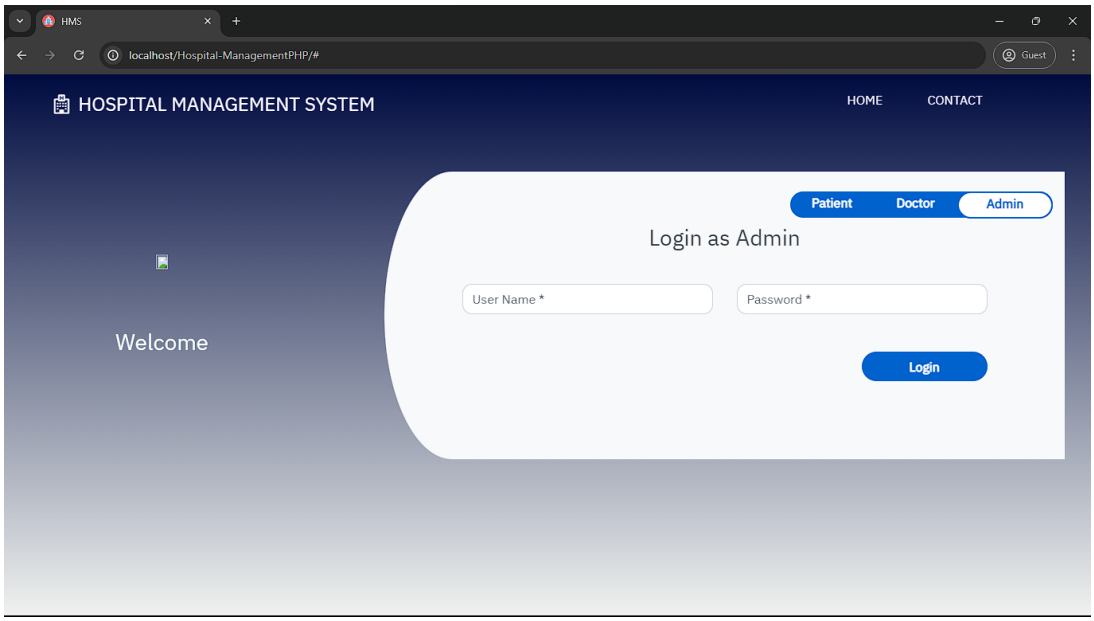

Fig 5.1.3 – Admin Login

Secure login and registration for Admin, Doctor, and Patient roles, ensuring role-based access to different features.

2. **Appointment Management**:

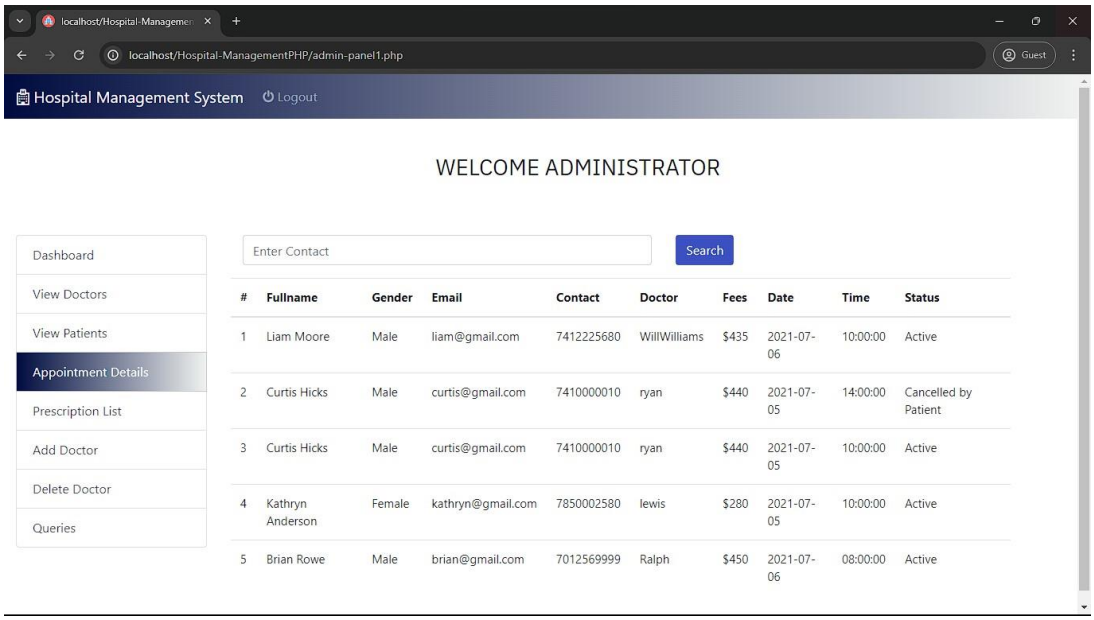

Fig 5.1.4 – Appointment management

- Patients can book, view, and cancel appointments with doctors.
- Doctors can approve or reject appointment requests and manage their schedules.
- Admins can oversee all appointments, ensuring smooth operation.

3. **Patient Management**:

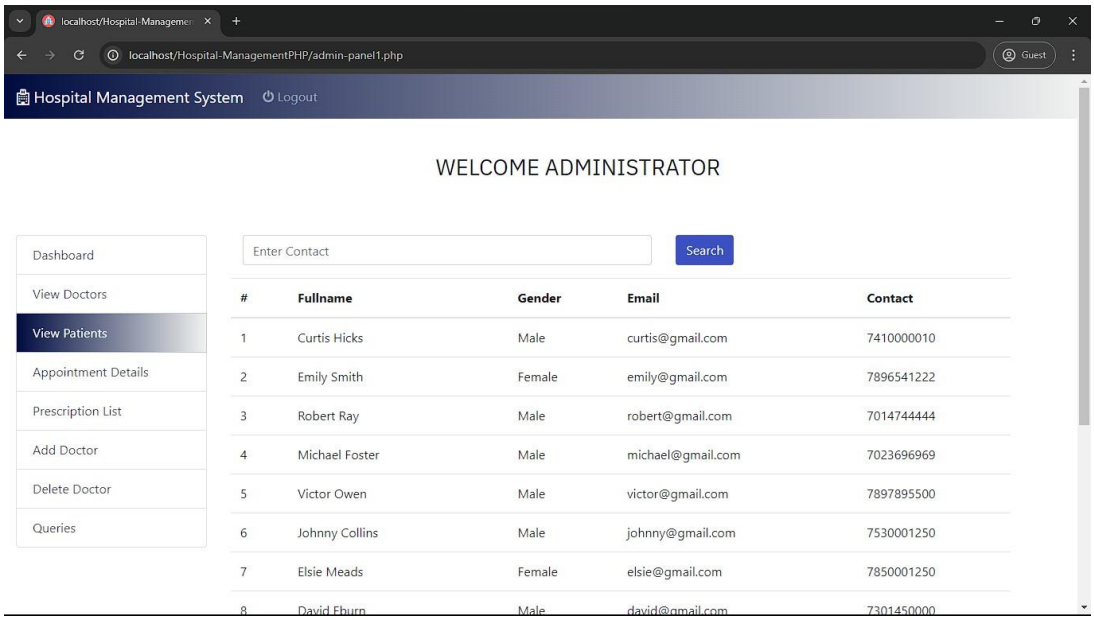

Fig 5.1.5 – Patient Management

- Patients can update personal information and view their medical history and prescriptions.
- Admins can access and manage patient records to maintain accurate and updated information.

4. **Doctor Management**:

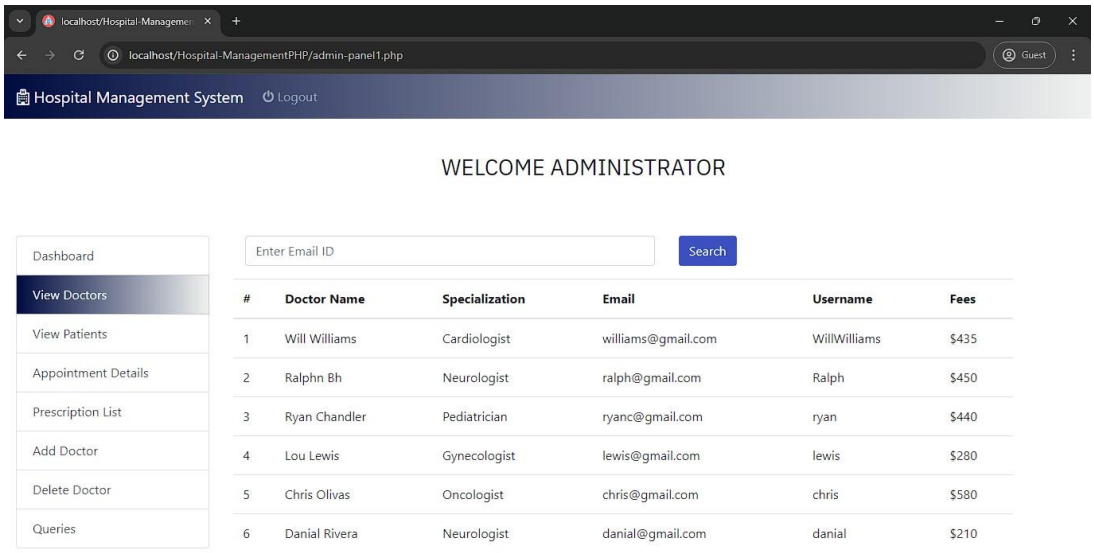

Fig 5.1.6 – Doctor management

- Doctors can update their profiles, view patient details, and record diagnoses and prescriptions.
- Admins can add, update, or remove doctor accounts, ensuring an up-to-date list of practitioners.

5. **Prescription Management**:

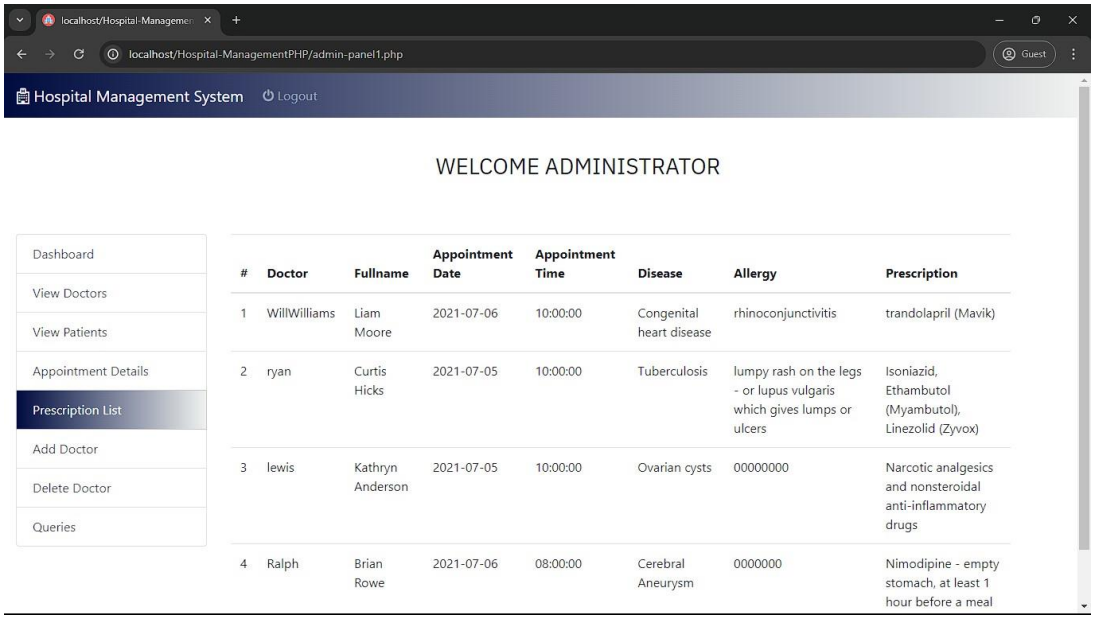

Fig 5.1.7 – Prescription Management

- Doctors can create and update prescriptions for patients based on consultations.
- Patients can view their prescriptions and follow prescribed treatments.

6. **Search and Filter**:
   - Admins can search and filter records for patients, doctors, and appointments to quickly find and manage data.

7. **Contact Management**:

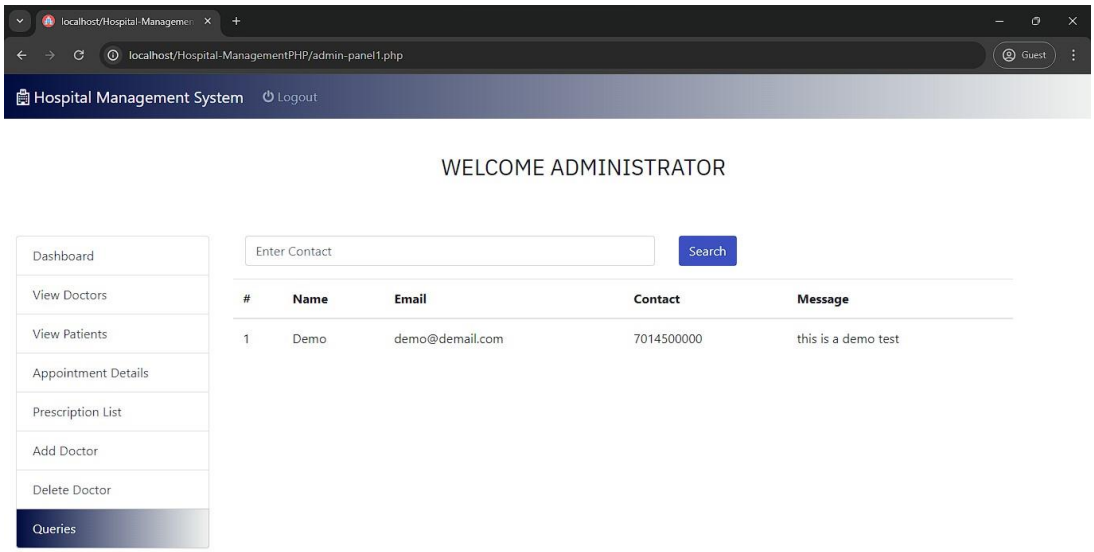

Fig 5.1.8 – Contact management

- Admins can view and manage contact queries from users, ensuring timely responses to patient and doctor inquiries.

5.2 User Feedback

The HMS has received positive feedback from users, particularly for its user-friendly interface and efficient management of hospital operations. Key points from user feedback include:

- **Patients** appreciated the ease of booking and managing appointments, as well as the ability to view their medical history and prescriptions online.
- **Doctors** found the system helpful for managing their schedules and patient records efficiently, allowing them to focus more on patient care.
- **Admins** highlighted the comprehensive control the system provides over hospital operations, enabling them to maintain accurate records and ensure smooth functionality.

Some suggestions for improvement included:

- Implementing a notification system to remind patients of upcoming appointments.
- Adding more detailed patient history and health records for better diagnosis.
- Enhancing the search functionality with more filters and sorting options.

5.3 Challenges Faced During Development

The development of the HMS project presented several challenges, which were addressed through collaborative efforts and strategic problem-solving:

1. **Data Security and Privacy**:
    - Ensuring the security and privacy of patient data was paramount. Implementing robust encryption techniques and secure authentication methods was challenging but necessary to protect sensitive information.
2. **Role-Based Access Control**:
    - Designing a system that accurately enforced role-based access control to restrict features based on user roles required careful planning and implementation to prevent unauthorized access.
3. **Scalability**:
    - Building a scalable system that could handle an increasing number of users and data without compromising performance involved optimizing database queries and ensuring efficient code practices.
4. **User Interface Design**:
    - Creating an intuitive and user-friendly interface for diverse user roles (patients, doctors, admins) was challenging. It required iterative design and feedback loops to ensure the UI met the needs of all users.
5. **Integration and Testing**:
    - Integrating various system components and ensuring they worked seamlessly together was complex. Rigorous testing was conducted to identify and fix bugs, ensuring a stable and reliable system.

By overcoming these challenges, the HMS project successfully delivered a robust and efficient system to manage hospital operations, improving the overall healthcare experience for patients, doctors, and administrators.

6. CONCLUSION

In conclusion, the Hospital Management System (HMS) project represents a comprehensive and effective solution for managing various aspects of hospital operations. Developed using PHP, MySQL, Bootstrap, HTML, CSS, and JavaScript, this system offers a user-friendly interface and a robust set of features tailored to the needs of patients, doctors, and administrators.

Through the patient panel, users can conveniently book appointments, access their appointment history, view prescriptions, and manage their personal information. The doctor panel streamlines the appointment approval process and enables doctors to prescribe medications seamlessly. Additionally, the admin panel provides administrators with full control over the system, allowing them to manage doctor accounts, view patient records, and handle contact queries efficiently.

With its intuitive design and comprehensive functionality, the HMS project serves as a valuable tool for enhancing the efficiency and effectiveness of hospital management. By automating key processes and centralizing patient data, this system facilitates smoother operations and improves the overall quality of healthcare services.

As healthcare continues to evolve, the Hospital Management System stands poised to adapt and innovate, supporting healthcare providers in delivering optimal care to patients. This project underscores the importance of technology in modern healthcare delivery and serves as a testament to the potential of software solutions to transform the healthcare industry for the better.

7. REFERENCES

a. Official PHP Documentation: https://www.php.net/docs.php
b. Official MySQL Documentation: https://dev.mysql.com/doc/
c. Official Bootstrap Documentation: https://getbootstrap.com/docs/5.0/getting-started/introduction/
d. MDN Web Docs for HTML: https://developer.mozilla.org/en-US/docs/Web/HTML
e. MDN Web Docs for CSS: https://developer.mozilla.org/en-US/docs/Web/CSS
f. MDN Web Docs for JavaScript: https://developer.mozilla.org/en-US/docs/Web/JavaScript
g. jQuery Documentation: https://api.jquery.com/
h. AJAX Introduction on W3Schools: https://www.w3schools.com/xml/ajax_intro.asp
i. CodexWorld Rating System: https://www.codexworld.com/star-rating-system-in-php-mysql-jquery/
j. PHP File Upload Documentation: https://www.php.net/manual/en/features.file-upload.php
k. OWASP Top Ten Security Risks: https://owasp.org/www-project-top-ten/
l. Prepared Statements in PHP: https://www.php.net/manual/en/mysqli.quickstart.prepared-statements.php
m. Nielsen Norman Group: https://www.nngroup.com/
n. Smashing Magazine: https://www.smashingmagazine.com/
o. Visual Studio Code: https://code.visualstudio.com/
p. XAMPP for PHP and MySQL: https://www.apachefriends.org/index.html
q. Stack Overflow: https://stackoverflow.com/
r. W3Schools: https://www.w3schools.com/
s. GitHub for version control: https://github.com/