A
MINI PROJECT REPORT
ON

"HOSPITAL MANAGEMENT SYSTEM"

Of

Database Management System Lab

SUBMITTED BY:

ARYAN SINGH (RA2111003011026)

UNDER THE GUIDANCE OF

Dr. Karthikeyan M

(Assistant Professor, Department of Computing Technologies)

Department Of Computing Technologies,
School of Computing
SRM Institute of Science and Technology.

Academic Year 2023- 24

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled "**HOSPITAL Management System**" is the bona fide work of **ARYAN SINGH [RA2111003011026]**, who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Karthikeyan M
Assistant Professor
Department of Computing Technologies

ABSTRACT

Hospital Management System (HMS) developed utilizing cutting-edge database management tools to optimize operational workflows and enhance patient care delivery. The HMS integrates an array of modules including patient registration, appointment scheduling, electronic medical records (EMR), billing and invoicing, pharmacy management, inventory control, and reporting. Powered by industry-standard database management systems such as MySQL, PostgreSQL, or MongoDB, the system ensures secure storage, efficient retrieval, and seamless manipulation of vast amounts of healthcare data.

Additionally, the HMS harnesses the capabilities of SQL (Structured Query Language) for complex data querying, indexing, and management, enabling healthcare professionals to extract valuable insights for informed decision-making.

Leveraging cloud-based platforms like AWS or Azure ensures scalability, flexibility, and accessibility of the system across multiple devices and locations. Through the integration of advanced database management tools, the HMS aims to enhance operational efficiency, reduce administrative burdens, and prioritize patient-centric care, thereby elevating the overall quality of healthcare services delivered by hospitals..

Table of Contents

INTRODUCTION

In today's rapidly evolving healthcare landscape, efficient management of hospital resources and seamless delivery of patient care are paramount. This presents a comprehensive Hospital Management System (HMS) designed to streamline operations, enhance communication, and improve patient outcomes.

The HMS encompasses various modules including patient registration, appointment scheduling, electronic medical records (EMR), billing and invoicing, pharmacy management, inventory control, and reporting. Leveraging modern technologies such as cloud computing, artificial intelligence, and data analytics, the HMS optimizes resource utilization, minimizes errors, and enhances decision-making processes.

Moreover, the system prioritizes patient-centric care by facilitating easy access to medical records, personalized treatment plans, and timely communication between healthcare providers and patients. By implementing the proposed HMS, hospitals can achieve operational efficiency, cost-effectiveness, and ultimately, elevate the quality of healthcare services provided to patients.

SCOPE

In today's rapidly evolving healthcare landscape, efficient management of hospital resources and seamless delivery of patient care are paramount. This presents a comprehensive Hospital Management System (HMS) designed to streamline operations, enhance communication, and improve patient outcomes.

- **Patient Management**: The system facilitates efficient management of patient records including registration, demographic details, medical history, diagnosis, treatment plans, and follow-up appointments.

- **Appointment Scheduling**: It provides tools for scheduling and managing patient appointments, optimizing the utilization of healthcare resources such as doctors' time and equipment.

- **Electronic Medical Records (EMR)**: The HMS enables the creation, storage, retrieval, and sharing of electronic medical records, ensuring accurate and up-to-date documentation of patient health information.

- **Billing and Invoicing**: It includes features for automated billing and invoicing, integrating with insurance systems, and generating accurate financial reports for billing and accounting purposes.

- **Pharmacy Management**: The system supports pharmacy operations including inventory management, prescription processing, dispensing medications, and tracking medication usage and interactions.

- **Inventory Control**: It helps in managing and tracking the inventory of medical supplies, equipment, and pharmaceuticals, ensuring availability and avoiding stockouts or wastage.

- **Laboratory Management**: The HMS may include modules for managing laboratory tests, results, and specimens, facilitating efficient communication between healthcare providers and laboratory staff.

- **Reporting and Analytics**: It provides tools for generating various reports such as patient demographics, financial summaries, utilization statistics, and quality metrics, enabling data-driven decision-making and performance monitoring.

REQUIREMENT

Functional Requirements:

1. **Patient Management**:
    - Allow registration of new patients with demographic details.
    - Enable tracking of patient visits, medical history, and allergies.
    - Provide functionality for managing patient appointments and scheduling.
2. **Electronic Medical Records (EMR)**:
    - Enable creation, storage, and retrieval of electronic medical records.
    - Facilitate integration with diagnostic tests, treatments, and medications.
    - Implement secure access control and audit trails for EMR data.
3. **Appointment Scheduling**:
    - Allow scheduling of appointments for patients with healthcare providers.
    - Provide real-time availability checking for doctors, rooms, and equipment.
    - Send automated appointment reminders to patients.
4. **Billing and Invoicing**:
    - Generate bills and invoices for patient services.
    - Integrate with insurance systems for claims processing.
    - Support different payment methods and billing cycles.
5. **Pharmacy Management**:
    - Manage inventory of medications and pharmaceuticals.
    - Process prescriptions and dispense medications.
    - Track medication usage and interactions.
6. **Inventory Control**:
    - Manage medical supplies, equipment, and consumables.
    - Implement automated reordering and stock level alerts.
    - Support barcode scanning and tracking of inventory movements.
7. **Laboratory Management**:
    - Enable ordering and tracking of laboratory tests and specimens.
    - Integrate with laboratory information systems (LIS).
    - Provide reporting of test results and interpretations.

Non-Functional Requirements:

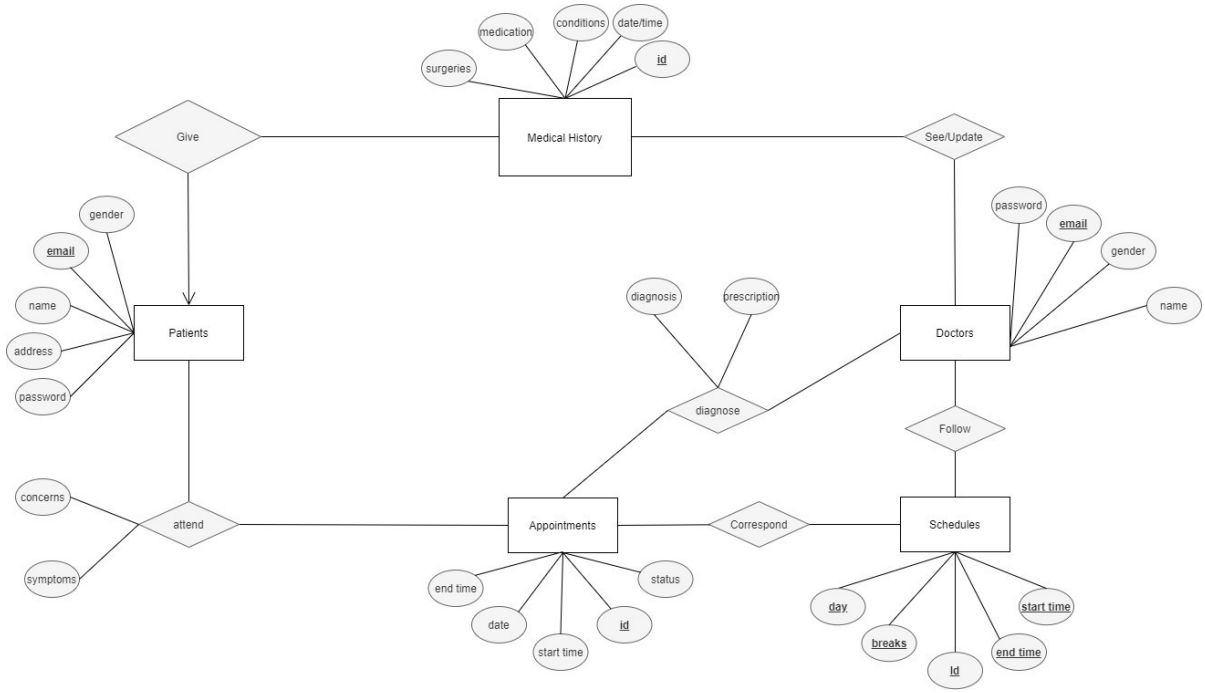
1. **Security**:
    - Ensure data security to protect patient information from unauthorized access or breaches.
    - Implement role-based access control to restrict access to sensitive information.
    - Encrypt data transmission and storage to maintain confidentiality.
2. **Performance**:
    - Ensure the system responds promptly to user requests and transactions.
    - Handle concurrent user sessions efficiently without performance degradation.
    - Optimize database queries and resource utilization for optimal performance.
3. **Scalability**:
    - Design the system to handle increasing volumes of data and users over time.
    - Implement scalable architecture to support future growth and expansion.
    - Provide mechanisms for horizontal and vertical scaling as needed.
4. **Reliability**:
    - Ensure the system operates reliably without frequent downtime or interruptions.
    - Implement fault tolerance mechanisms to handle system failures gracefully.

- Perform regular backups and data recovery procedures to prevent data loss.

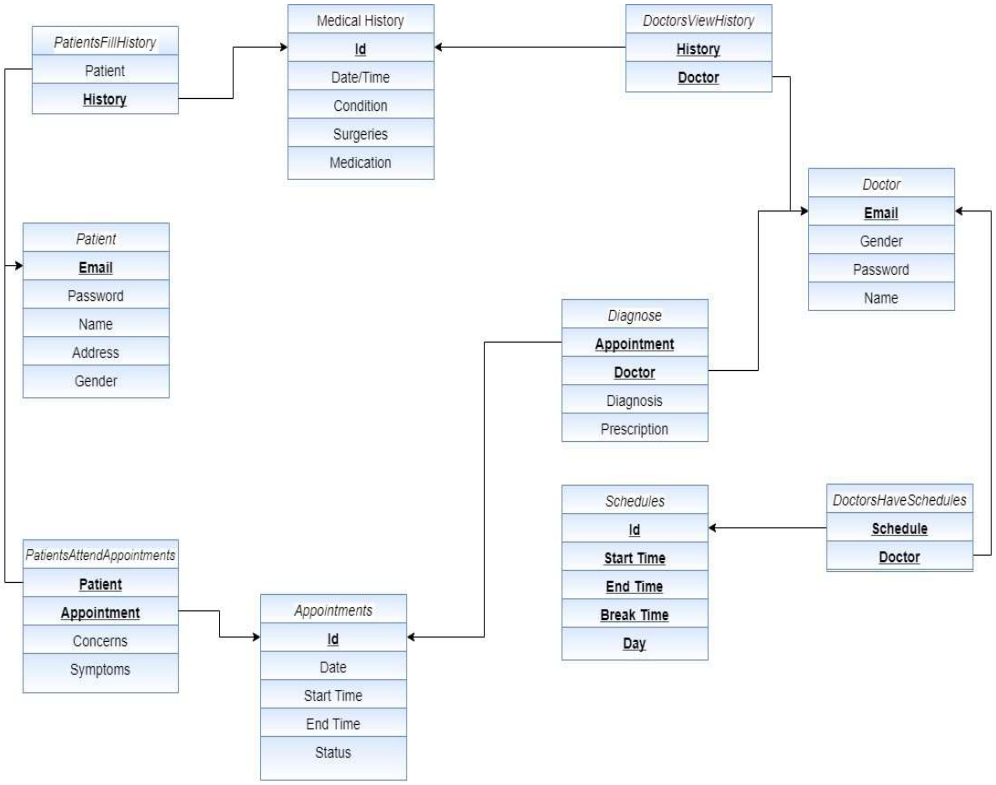
5. **Usability**:
   - Design a user-friendly interface that is intuitive and easy to navigate.
   - Provide clear documentation and training materials for users.
   - Support customization and personalization options to meet user preferences.
6. **Interoperability**:
   - Ensure compatibility and interoperability with external systems such as EHRs, billing systems, and medical devices.
   - Adhere to industry standards and protocols for data exchange and integration.

DATA MODELING

a. ER Diagram



b. Relational Model



c.

c. Normalization

To perform normalization on the provided schema diagram, we'll walk through the normal forms starting from the First Normal Form (1NF), through the Second Normal Form (2NF), and finally the Third Normal Form (3NF). I will assess the structure and relationships in the diagram to propose normalized versions of each entity and its attributes. Let's begin:

First Normal Form (1NF)

For a database schema to be in 1NF, it must meet the following criteria:
- Each table has a primary key: unique values that identify each row.
- All attributes (columns) in the table are atomic (no multi-valued or composite attributes).
- There are no repeating groups or arrays.

Tenant:
- Primary Key: tenant_id
- Attributes: password, dob, name, proof
- Appears to be in 1NF as there are no repeating groups, and all attributes are atomic.

Owner:
- Primary Key: owner_id
- Attributes: Room_no (FK), Name, age, password, proof, aggrement_status
- Appears to be in 1NF, but we should check if 'Room_no' is multivalued. If an owner can have multiple rooms, it should not be in this table.

Room:
- Primary Key: room_no
- Attributes: type, floor, parking_slot, reg_no, block_no (FK) - Appears to be in 1NF.

Rental:
- Composite Primary Key: tenant_id (FK), room_no (FK) - Attributes: date_of_joining, monthly_rent - Appears to be in 1NF.

Auth:
- Primary Key: id
- Attributes: password, user_id (U)
- The meaning of (U) is unclear. If it stands for "Unique", then it is in 1NF.

Block:
- Primary Key: block_no
- Attributes: block_name, complaints
- If complaints are stored as a list, it would violate 1NF. This needs to be atomic.

Block Admin:
- Primary Key: admin_id
- Attributes: admin_name, admin_pass, block_no (FK) - Appears to be in 1NF

Employees:
- Primary Key: emp_id
- Attributes: emp_name, salary, emp_type, age, block_no (FK) - Appears to be in 1NF.

Identity:
- Primary Key: tenant_id (FK), owner_id (FK)
- Attributes: Proof
- This table seems to act as a linking table but has a non-key attribute, which is unusual for a pure linking table. The design suggests that both a tenant and an owner can have an identity. This should be checked for correctness.

Second Normal Form (2NF)
For a database schema to be in 2NF, it must meet all the criteria of 1NF, and additionally: - All non-key attributes must be fully functionally dependent on the primary key.

Rental: (tenant_id, room_no) -> {date_of_joining, monthly_rent}
- No partial dependencies on a subset of the primary key, so it's in 2NF.

Identity:
- Not sure about the full functional dependency due to the composite key structure and its purpose.

For the other tables, as they have a single primary key and no composite keys, they are trivially in 2NF if they were in 1NF.

Third Normal Form (3NF)

For a database schema to be in 3NF, it must meet all the criteria of 2NF, and additionally:

- It should have no transitive dependencies (a non-key attribute depending on another non-key attribute).

We would need to examine the actual data and usage patterns to determine 3NF definitively. Still, from the diagram, we can infer the following:

Block Admin: (admin_id) -> admin_name, admin_pass, block_no

- We need to ensure that 'block_no' does not functionally determine 'admin_name' or 'admin_pass', which could be a transitive dependency.

Owner:

- If 'proof' is dependent on 'Name' or 'age' rather than 'owner_id', it would be a transitive dependency. Typically, this should not be the case, but the actual data should be checked.

Employees:

- If 'salary' depends on 'emp_type' rather than 'emp_id', there would be a transitive dependency.

SOFTWARE REQUIRMENTS

System Requirements:

1. **Operating System**:
   - Specify the supported operating systems for deploying the HMS (e.g., Windows Server, Linux distributions like Ubuntu or CentOS).
2. **Web Server**:
   - Choose a web server software to host the HMS application (e.g., Apache HTTP Server, Nginx).
3. **Application Server**:
   - Utilize an application server for deploying and managing the HMS application (e.g., Apache Tomcat, WildFly, Microsoft IIS).
4. **Database Server**:
   - Install and configure the chosen DBMS as the database server (e.g., MySQL Server, PostgreSQL Server, MongoDB).

FRONTEND REQUIREMENTS:

HTML5: Use HTML5 for structuring the frontend interface, providing semantic elements for content presentation.

Tailwind CSS: Implement Tailwind CSS for responsive and customizable styling, enabling rapid UI development with utility-first classes.

React JS: Utilize React JS for building interactive and dynamic user interfaces, leveraging component-based architecture for modularity and reusability.

State Management: Use Redux or React Context API for managing application state, ensuring efficient data flow and state synchronization across components.

Routing: Implement React Router for client-side routing, enabling navigation between different views and components within the application.

User Interface Design: Design intuitive and user-friendly interfaces adhering to design principles and accessibility guidelines.

Responsive Design: Ensure responsiveness across devices and screen sizes using responsive design techniques and media queries.

Testing Frameworks: Use testing frameworks like Jest and React Testing Library for unit and integration testing of frontend components.

BACKEND REQUIREMENTS:

Node.js: Utilize Node.js for backend server development, leveraging its event-driven, non-blocking I/O model for scalability and performance.

Express.js: Use Express.js as the web application framework for Node.js, facilitating the development of RESTful APIs and middleware for handling HTTP requests.

Database Management: Integrate with MySQL or another relational database management system (RDBMS) for storing and managing application data securely.

API Design: Design and implement RESTful APIs for frontend-backend communication, defining endpoints and data formats (e.g., JSON) for data exchange.

Authentication and Authorization: Implement secure authentication mechanisms (e.g., JWT) for user login and authorization, ensuring data privacy and access control.

Error Handling and Logging: Implement error handling and logging mechanisms to capture and handle exceptions, facilitating debugging and troubleshooting.

Middleware: Use middleware components for request processing, data validation, and security measures (e.g., CORS, CSRF protection).

Security: Apply security best practices such as input validation, parameterized queries, and encryption to protect against common security vulnerabilities.

Scalability and Performance: Optimize backend code and database queries for scalability and performance, considering factors like caching, indexing, and connection pooling.

Deployment: Deploy the backend application on a suitable hosting platform (e.g., AWS, Heroku) with appropriate configurations for scalability and reliability.

C. Database Connectivity:

Database connectivity is a critical aspect of software development, especially for systems like an Apartment Management System that require efficient and reliable data storage and retrieval. In this context, database connectivity refers to the ability of the application to interact with a database management system (DBMS) to perform CRUD (Create, Read, Update, Delete) operations on data. **MYSQL** and **MSQL Workbench** is used to database connectivity.

Key Elements of Database Connectivity:

Database Management System (DBMS): The choice of DBMS is crucial and often depends on factors like scalability, data structure, and transaction handling. For the Apartment Management System, MySQL is chosen, which is a popular relational database known for its robustness and ease of use.

Database Connection: The application establishes a connection to the MySQL database using a database driver specific to Node.js (e.g., mysql or mysql2). The connection parameters include the host, port, username, password, and database name.

Data Access Layer (DAL): The DAL is responsible for abstracting database interactions. It includes functions or methods to execute SQL queries, handle transactions, and map database records to application objects.

Connection Pooling: To optimize database performance, connection pooling is used to manage and reuse database connections efficiently. This helps reduce the overhead of establishing new connections for each database operation.

Error Handling: Proper error handling is implemented to manage database connection errors, SQL query errors, and transaction rollbacks in case of failures.

SAMPLE SOURCE CODE

```
<?php
include 'includes/connect.php';
    if($_SESSION['admin_sid']==session_id())
    {
        ?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1.0, user-scalable=no">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="msapplication-tap-highlight" content="no">
  <title>All orders</title>

  <!-- Favicons-->
  <link rel="icon" href="images/favicon/favicon-32x32.png" sizes="32x32">
  <!-- Favicons-->
  <link rel="apple-touch-icon-precomposed" href="images/favicon/apple-touch-icon-
152x152.png">
  <!-- For iPhone -->
  <meta name="msapplication-TileColor" content="#00bcd4">
  <meta name="msapplication-TileImage" content="images/favicon/mstile-
144x144.png">
  <!-- For Windows Phone -->


  <!-- CORE CSS-->
  <link href="css/materialize.min.css" type="text/css" rel="stylesheet"
media="screen,projection">
  <link href="css/style.min.css" type="text/css" rel="stylesheet"
media="screen,projection">
  <!-- Custome CSS-->
  <link href="css/custom/custom.min.css" type="text/css" rel="stylesheet"
media="screen,projection">

  <!-- INCLUDED PLUGIN CSS ON THIS PAGE -->
  <link href="js/plugins/perfect-scrollbar/perfect-scrollbar.css" type="text/css"
rel="stylesheet" media="screen,projection">
```

```
</head>

<body>
  <!-- Start Page Loading -->
  <div id="loader-wrapper">
      <div id="loader"></div>
      <div class="loader-section section-left"></div>
      <div class="loader-section section-right"></div>
  </div>
  <!-- End Page Loading -->

  <!--
////////////////////////////////////////////////////////////////////////////// -->

  <!-- START HEADER -->
  <header id="header" class="page-topbar">
        <!-- start header nav-->
        <div class="navbar-fixed">
            <nav class="navbar-color">
                <div class="nav-wrapper">
                    <ul class="left">
                      <li><h1 class="logo-wrapper"><a href="index.php"
class="brand-logo darken-1"><img src="images/materialize-logo.png" alt="logo"></a>
<span class="logo-text">Logo</span></h1></li>
                    </ul>
                </div>
            </nav>
        </div>
        <!-- end header nav-->
  </header>
  <!-- END HEADER -->

  <!--
////////////////////////////////////////////////////////////////////////////// -->

  <!-- START MAIN -->
  <div id="main">
    <!-- START WRAPPER -->
    <div class="wrapper">

      <!-- START LEFT SIDEBAR NAV-->
      <aside id="left-sidebar-nav">
        <ul id="slide-out" class="side-nav fixed leftside-navigation">
            <li class="user-details cyan darken-2">
            <div class="row">
                <div class="col col s4 m4 l4">
                    <img src="images/avatar.jpg" alt="" class="circle responsive-
img valign profile-image">
                </div>
                <div class="col col s8 m8 l8">
                    <ul id="profile-dropdown" class="dropdown-content">
```

```
                        <li><a href="routers/logout.php"><i class="mdi-hardware-
keyboard-tab"></i> Logout</a>
                        </li>
                    </ul>
                </div>
                <div class="col col s8 m8 l8">
                    <a class="btn-flat dropdown-button waves-effect waves-light
white-text profile-btn" href="#" data-activates="profile-dropdown"><?php echo
$name;?> <i class="mdi-navigation-arrow-drop-down right"></i></a>
                    <p class="user-roal"><?php echo $role;?></p>
                </div>
            </div>
            </li>
            <li class="bold"><a href="index.php" class="waves-effect waves-
cyan"><i class="mdi-editor-border-color"></i> Food Menu</a>
            </li>
                <li class="no-padding">
                    <ul class="collapsible collapsible-accordion">
                        <li class="bold"><a class="collapsible-header waves-effect
waves-cyan active"><i class="mdi-editor-insert-invitation"></i> Orders</a>
                            <div class="collapsible-body">
                                <ul>
                                <li class="<?php
                                if(!isset($_GET['status'])){
                                        echo 'active';
                                    }?>
                                    "><a href="all-orders.php">All Orders</a>
                                </li>
                                <?php
                                    $sql = mysqli_query($con, "SELECT DISTINCT
status FROM orders;");
                                    while($row = mysqli_fetch_array($sql)){
                                    if(isset($_GET['status'])){
                                        $status = $row['status'];
                                    }
                                    echo '<li
class='.(isset($_GET['status'])?($status == $_GET['status'] ? 'active' : ''):
'').'><a href="all-orders.php?status='.$row['status'].'">'.$row['status'].'</a>
                                    </li>';
                                    }
                                    ?>
                                </ul>
                            </div>
                        </li>
                    </ul>
                </li>
                 <li class="no-padding">
                    <ul class="collapsible collapsible-accordion">
                        <li class="bold"><a class="collapsible-header waves-effect
waves-cyan"><i class="mdi-action-question-answer"></i> Tickets</a>
                            <div class="collapsible-body">
```

```
                                <ul>
                                <li><a href="all-tickets.php">All Tickets</a>
                                </li>
                                <?php
                                    $sql = mysqli_query($con, "SELECT DISTINCT
status FROM tickets;");
                                    while($row = mysqli_fetch_array($sql)){
                                    echo '<li><a href="all-
tickets.php?status='.$row['status'].'">'.$row['status'].'</a>
                                    </li>';
                                    }
                                    ?>
                                </ul>
                            </div>
                        </li>
                    </ul>
                </li>
            <li class="bold"><a href="users.php" class="waves-effect waves-
cyan"><i class="mdi-social-person"></i> Users</a>
            </li>
            <?php
include 'includes/connect.php';
    if($_SESSION['admin_sid']==session_id())
    {
        ?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1.0, user-scalable=no">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="msapplication-tap-highlight" content="no">
  <title>All orders</title>

  <!-- Favicons-->
  <link rel="icon" href="images/favicon/favicon-32x32.png" sizes="32x32">
  <!-- Favicons-->
  <link rel="apple-touch-icon-precomposed" href="images/favicon/apple-touch-icon-
152x152.png">
  <!-- For iPhone -->
  <meta name="msapplication-TileColor" content="#00bcd4">
  <meta name="msapplication-TileImage" content="images/favicon/mstile-
144x144.png">
  <!-- For Windows Phone -->


  <!-- CORE CSS-->
  <link href="css/materialize.min.css" type="text/css" rel="stylesheet"
media="screen,projection">
```

```
  <link href="css/style.min.css" type="text/css" rel="stylesheet"
media="screen,projection">
  <!-- Custome CSS-->
  <link href="css/custom/custom.min.css" type="text/css" rel="stylesheet"
media="screen,projection">

  <!-- INCLUDED PLUGIN CSS ON THIS PAGE -->
  <link href="js/plugins/perfect-scrollbar/perfect-scrollbar.css" type="text/css"
rel="stylesheet" media="screen,projection">

</head>

<body>
  <!-- Start Page Loading -->
  <div id="loader-wrapper">
      <div id="loader"></div>
      <div class="loader-section section-left"></div>
      <div class="loader-section section-right"></div>
  </div>
  <!-- End Page Loading -->

  <!--
////////////////////////////////////////////////////////////////////////////// -->

  <!-- START HEADER -->
  <header id="header" class="page-topbar">
        <!-- start header nav-->
        <div class="navbar-fixed">
            <nav class="navbar-color">
                <div class="nav-wrapper">
                    <ul class="left">
                      <li><h1 class="logo-wrapper"><a href="index.php"
class="brand-logo darken-1"><img src="images/materialize-logo.png" alt="logo"></a>
<span class="logo-text">Logo</span></h1></li>
                    </ul>
                </div>
            </nav>
        </div>
        <!-- end header nav-->
  </header>
  <!-- END HEADER -->

  <!--
////////////////////////////////////////////////////////////////////////////// -->

  <!-- START MAIN -->
  <div id="main">
    <!-- START WRAPPER -->
    <div class="wrapper">

      <!-- START LEFT SIDEBAR NAV-->
```

```
      <aside id="left-sidebar-nav">
        <ul id="slide-out" class="side-nav fixed leftside-navigation">
            <li class="user-details cyan darken-2">
            <div class="row">
                <div class="col col s4 m4 l4">
                    <img src="images/avatar.jpg" alt="" class="circle responsive-
img valign profile-image">
                </div>
                <div class="col col s8 m8 l8">
                    <ul id="profile-dropdown" class="dropdown-content">
                        <li><a href="routers/logout.php"><i class="mdi-hardware-
keyboard-tab"></i> Logout</a>
                        </li>
                    </ul>
                </div>
                <div class="col col s8 m8 l8">
                    <a class="btn-flat dropdown-button waves-effect waves-light
white-text profile-btn" href="#" data-activates="profile-dropdown"><?php echo
$name;?> <i class="mdi-navigation-arrow-drop-down right"></i></a>
                    <p class="user-roal"><?php echo $role;?></p>
                </div>
            </div>
            </li>
            <li class="bold"><a href="index.php" class="waves-effect waves-
cyan"><i class="mdi-editor-border-color"></i> Food Menu</a>
            </li>
                <li class="no-padding">
                    <ul class="collapsible collapsible-accordion">
                        <li class="bold"><a class="collapsible-header waves-effect
waves-cyan active"><i class="mdi-editor-insert-invitation"></i> Orders</a>
                            <div class="collapsible-body">
                                <ul>
                                <li class="<?php
                                if(!isset($_GET['status'])){
                                        echo 'active';
                                    }?>
                                    "><a href="all-orders.php">All Orders</a>
                                </li>
                                <?php
                                    $sql = mysqli_query($con, "SELECT DISTINCT
status FROM orders;");

                                    while($row = mysqli_fetch_array($sql)){
                                    if(isset($_GET['status'])){
                                        $status = $row['status'];
                                    }
                                    echo '<li
class='.(isset($_GET['status'])?($status == $_GET['status'] ? 'active' : ''):
'').'><a href="all-orders.php?status='.$row['status'].'">'.$row['status'].'</a>
                                    </li>';
                                    }
                                    ?>
```

```
                        </ul>
                    </div>
                </li>
            </ul>
        </li>
```

Mysql Connectivity Code:-

```
{
  "name": "wecare-backend",
  "version": "0.0.0",
  "lockfileVersion": 1,
  "requires": true,
  "dependencies": {
    "accepts": {
      "version": "1.3.7",
      "resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.7.tgz",
      "integrity": "sha512-
Il80Qs2WjYlJIBNzNkK6KYqlVMTbZLXgHx2oT0pU/fjRHyEp+PEfEPY0R3WCwAGVOtauxh1hOxNgIf5bv7
dQpA==",
      "requires": {
        "mime-types": "~2.1.24",
        "negotiator": "0.6.2"
      }
    },
    "acorn": {
      "version": "2.7.0",
      "resolved": "https://registry.npmjs.org/acorn/-/acorn-2.7.0.tgz",
      "integrity": "sha1-q259nYhqrKiwhbwzEreaGYQz8Oc="
    },
    "acorn-globals": {
      "version": "1.0.9",
      "resolved": "https://registry.npmjs.org/acorn-globals/-/acorn-globals-
1.0.9.tgz",
      "integrity": "sha1-VbtemGkVB7dFedBRNBMhfDgMVM8=",
      "requires": {
        "acorn": "^2.1.0"
      }
    },
    "align-text": {
      "version": "0.1.4",
      "resolved": "https://registry.npmjs.org/align-text/-/align-text-0.1.4.tgz",
      "integrity": "sha1-DNkKVhCT810KmSVsIrcGlDP60Rc=",
      "requires": {
        "kind-of": "^3.0.2",
        "longest": "^1.0.1",
        "repeat-string": "^1.5.2"
      }
    },
```

```
    "amdefine": {
      "version": "1.0.1",
      "resolved": "https://registry.npmjs.org/amdefine/-/amdefine-1.0.1.tgz",
      "integrity": "sha1-SlKCrBZHKek2Gbz9OtFR+BfOkfU="
    },
    "array-flatten": {
      "version": "1.1.1",
      "resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-
1.1.1.tgz",
      "integrity": "sha1-ml9pkFGx5wczKPKgCJaLZOopVdI="
    },
    "asap": {
      "version": "1.0.0",
      "resolved": "https://registry.npmjs.org/asap/-/asap-1.0.0.tgz",
      "integrity": "sha1-sqRdpf36ILBJb8N2jMJ8EvqRan0="
    },
    "basic-auth": {
      "version": "2.0.1",
      "resolved": "https://registry.npmjs.org/basic-auth/-/basic-auth-2.0.1.tgz",
      "integrity": "sha512-
NF+epuEdnUYVlGuhaxbbq+dvJttwLnGY+YixlXlME5KpQ5W3CnXA5cVTneY3SPbPDRkcjMbifrwmFYcClg
OZeg==",
      "requires": {
        "safe-buffer": "5.1.2"
      }
    },
    "bignumber.js": {
      "version": "7.2.1",
      "resolved": "https://registry.npmjs.org/bignumber.js/-/bignumber.js-
7.2.1.tgz",
      "integrity": "sha512-
S4XzBk5sMB+Rcb/LNcpzXr57VRTxgAvaAEDAl1AwRx27j00hT84O6OkteE7u8UB3NuaaygCRrEpqox4uDO
rbdQ=="
    },
    "body-parser": {
      "version": "1.18.3",
      "resolved": "https://registry.npmjs.org/body-parser/-/body-parser-
1.18.3.tgz",
      "integrity": "sha1-WykhmP/dVTs6DyDe0FkrlWlVyLQ=",
      "requires": {
        "bytes": "3.0.0",
        "content-type": "~1.0.4",
        "debug": "2.6.9",
        "depd": "~1.1.2",
        "http-errors": "~1.6.3",
        "iconv-lite": "0.4.23",
        "on-finished": "~2.3.0",
        "qs": "6.5.2",
        "raw-body": "2.3.3",
        "type-is": "~1.6.16"
      }
```

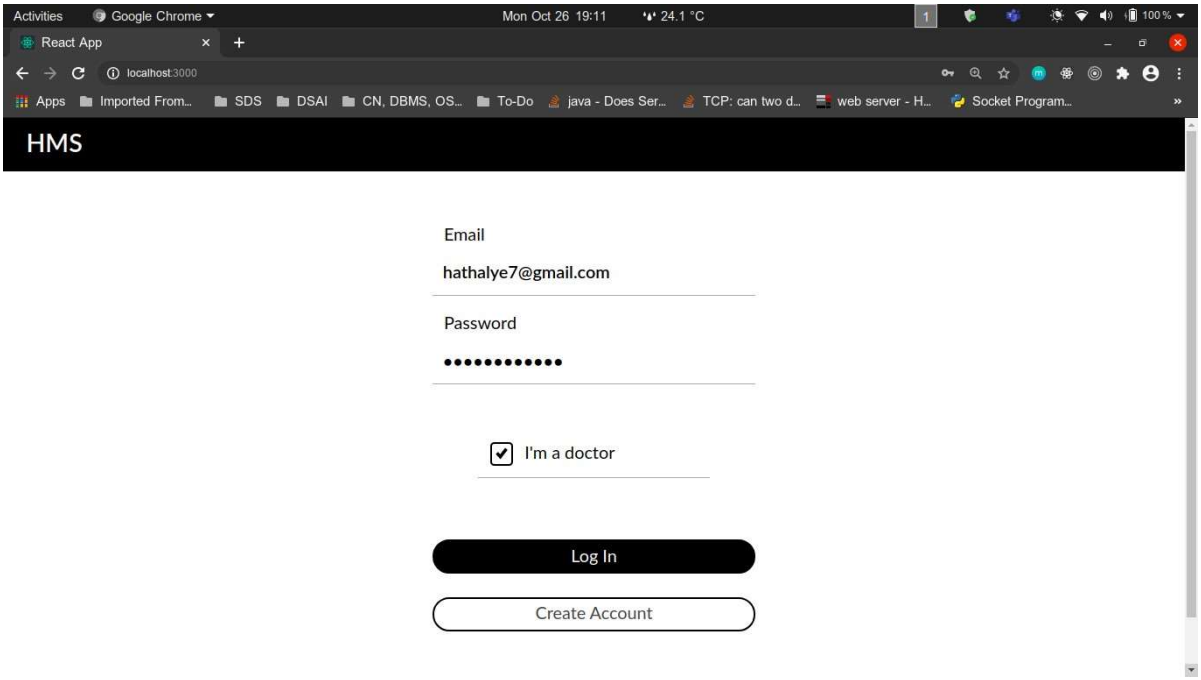
```
    },
    "bytes": {
      "version": "3.0.0",
      "resolved": "https://registry.npmjs.org/bytes/-/bytes-3.0.0.tgz",
      "integrity": "sha1-0ygVQE1olpn4Wk6k+odV3ROpYEg="
    },
    "camelcase": {
      "version": "1.2.1",
      "resolved": "https://registry.npmjs.org/camelcase/-/camelcase-1.2.1.tgz",
      "integrity": "sha1-m7UwTS4LVmmLLHWLCKPqqdqlijk="
    },
    "center-align": {
      "version": "0.1.3",
      "resolved": "https://registry.npmjs.org/center-align/-/center-align-
0.1.3.tgz",
      "integrity": "sha1-qg0yYptu6XIgBBHL1EYckHvCt60=",
      "requires": {
        "align-text": "^0.1.3",
        "lazy-cache": "^1.0.3"
      }
    },
    "character-parser": {
      "version": "1.2.1",
      "resolved": "https://registry.npmjs.org/character-parser/-/character-parser-
1.2.1.tgz",
      "integrity": "sha1-wN3kqxgnE7kZuXCVmhI+zBow/NY="
    },
    "clean-css": {
      "version": "3.4.28",
      "resolved": "https://registry.npmjs.org/clean-css/-/clean-css-3.4.28.tgz",
      "integrity": "sha1-vxlF6C/ICPVWlebd6uwBQA79A/8=",
      "requires": {
        "commander": "2.8.x",
        "source-map": "0.4.x"
      },
      "dependencies": {
        "commander": {
          "version": "2.8.1",
          "resolved": "https://registry.npmjs.org/commander/-/commander-
2.8.1.tgz",
          "integrity": "sha1-Br42f+v9oMMwqh4qBy09yXYkJdQ=",
          "requires": {
            "graceful-readlink": ">= 1.0.0"
          }
        }
      }
    },
    "cliui": {
      "version": "2.1.0",
      "resolved": "https://registry.npmjs.org/cliui/-/cliui-2.1.0.tgz",
      "integrity": "sha1-S0dXYP+AJkx2LDoXGQMukcf+oNE=",
```

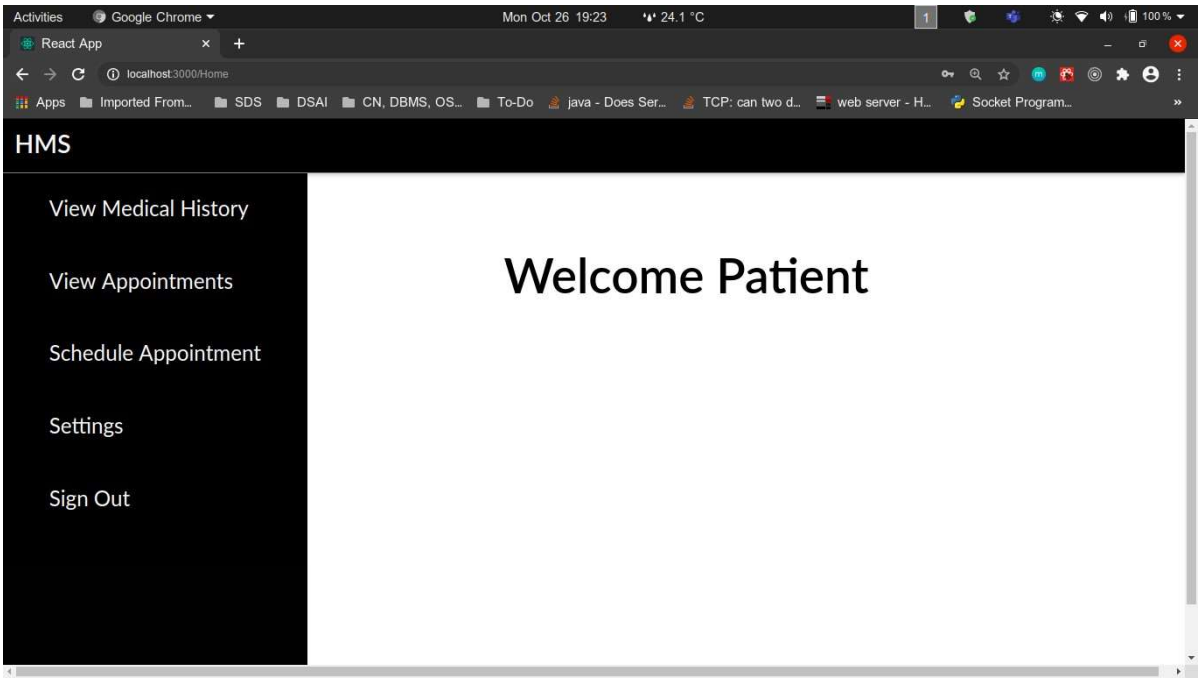
```
      "requires": {
        "center-align": "^0.1.1",
        "right-align": "^0.1.1",
        "wordwrap": "0.0.2"
      },
      "dependencies": {
        "wordwrap": {
          "version": "0.0.2",
          "resolved": "https://registry.npmjs.org/wordwrap/-/wordwrap-0.0.2.tgz",
          "integrity": "sha1-t5Zpu0LstAn4PVg8rVLKF+qhZD8="
        }
      }
    },
    "commander": {
      "version": "2.6.0",
      "resolved": "https://registry.npmjs.org/commander/-/commander-2.6.0.tgz",
      "integrity": "sha1-nfflL7Kgyw+4kFjugMMQQiXzfh0="
    },
    "constantinople": {
      "version": "3.0.2",
      "resolved": "https://registry.npmjs.org/constantinople/-/constantinople-
3.0.2.tgz",
      "integrity": "sha1-S5RdmTeQe82Y7ldRIsOBdRZUQUE=",
      "requires": {
        "acorn": "^2.1.0"
      }
    },
    "content-disposition": {
      "version": "0.5.2",
      "resolved": "https://registry.npmjs.org/content-disposition/-/content-
disposition-0.5.2.tgz",
      "integrity": "sha1-DPaLud318r55YcOoUXjLhdunjLQ="
    },
INSERT INTO Patient(email,password,name,address,gender)
VALUES
('ramesh@gmail.com','hrishikesh13','Ramesh','Tamil Nadu', 'male'),
('suresh@gmail.com','hrishikesh13','Suresh','Karnataka', 'male'),
('rakesh@gmail.com','hrishikesh13','Rakesh','Gujarat', 'male')
;

INSERT INTO MedicalHistory(id,date,conditions,surgeries,medication)
VALUES
(1,'19-01-14','Pain in abdomen','Heart Surgery','Crocin'),
(2,'19-01-14','Frequent Indigestion','none','none'),
(3,'19-01-14','Body Pain','none','Iodex')
;

INSERT INTO Doctor(email, gender, password, name)
VALUES
('hathalye7@gmail.com', 'male', 'hrishikesh13', 'Hrishikesh Athalye'),
('hathalye8@gmail.com', 'male', 'hrishikesh13', 'Hrishikesh Athalye')
```

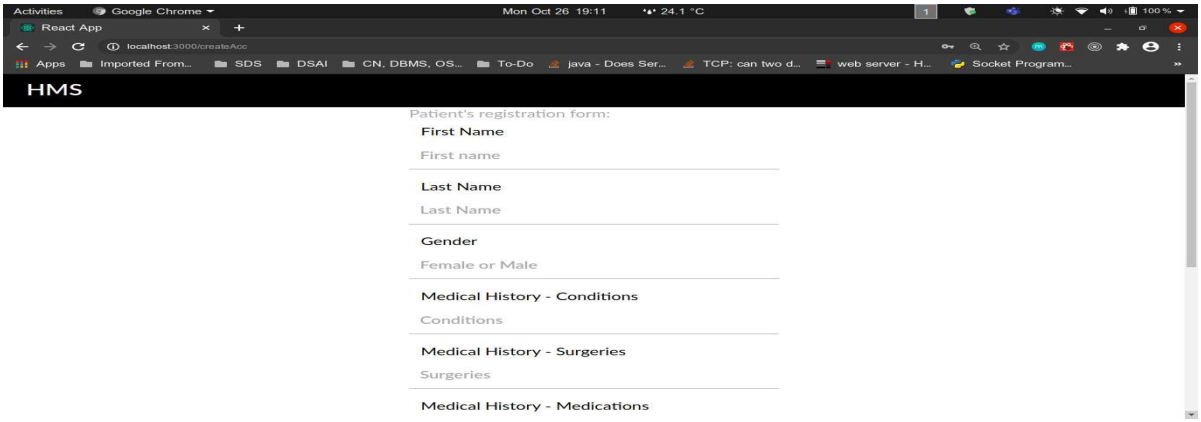
```
;

INSERT INTO Appointment(id,date,starttime,endtime,status)
VALUES
(1, '19-01-15', '09:00', '10:00', 'Done'),
(2, '19-01-16', '10:00', '11:00', 'Done'),
(3, '19-01-18', '14:00', '15:00', 'Done')
;

INSERT INTO PatientsAttendAppointments(patient,appt,concerns,symptoms)
VALUES
('ramesh@gmail.com',1, 'none', 'itchy throat'),
('suresh@gmail.com',2, 'infection', 'fever'),
('rakesh@gmail.com',3, 'nausea', 'fever')
;

INSERT INTO Schedule(id,starttime,endtime,breaktime,day)
VALUES
(001,'09:00','17:00','12:00','Tuesday'),
(001,'09:00','17:00','12:00','Friday'),
(001,'09:00','17:00','12:00','Saturday'),
(001,'09:00','17:00','12:00','Sunday'),
(002,'09:00','17:00','12:00','Wednesday'),
(002,'09:00','17:00','12:00','Friday')
;

INSERT INTO PatientsFillHistory(patient,history)
VALUES
('ramesh@gmail.com', 1),
('suresh@gmail.com', 2),
('rakesh@gmail.com', 3)
;

INSERT INTO Diagnose(appt,doctor,diagnosis,prescription)
VALUES
(1,'hathalye7@gmail.com', 'Bloating', 'Ibuprofen as needed'),
(2,'hathalye8@gmail.com', 'Muscle soreness', 'Stretch morning/night'),
(3,'hathalye8@gmail.com', 'Vitamin Deficiency', 'Good Diet')
;

INSERT INTO DocsHaveSchedules(sched,doctor)
VALUES
(001,'hathalye7@gmail.com'),
(002,'hathalye8@gmail.com')
;

INSERT INTO DoctorViewsHistory(history,doctor)
VALUES
(1,'hathalye7@gmail.com'),
(2,'hathalye8@gmail.com'),
(3,'hathalye8@gmail.com')
```

;

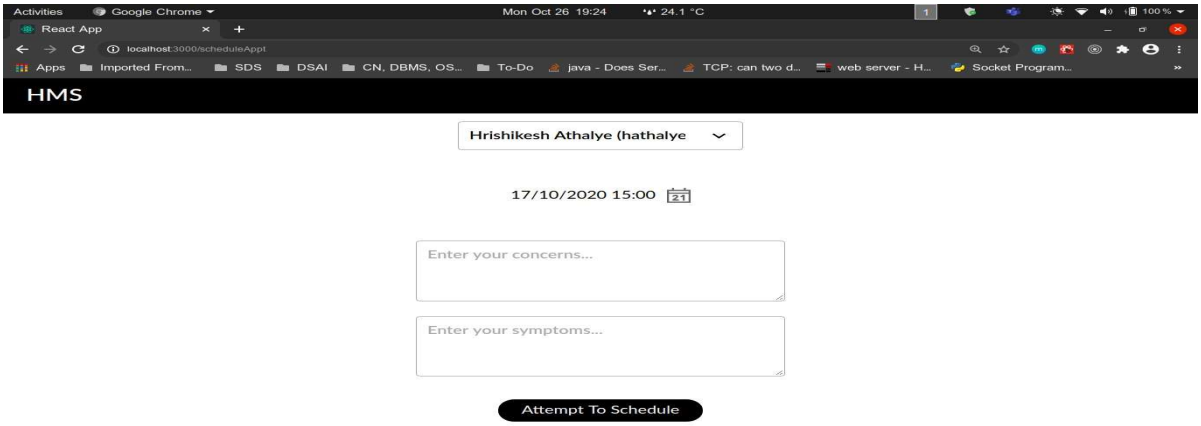
SAMPLE SCREEN SHOTS

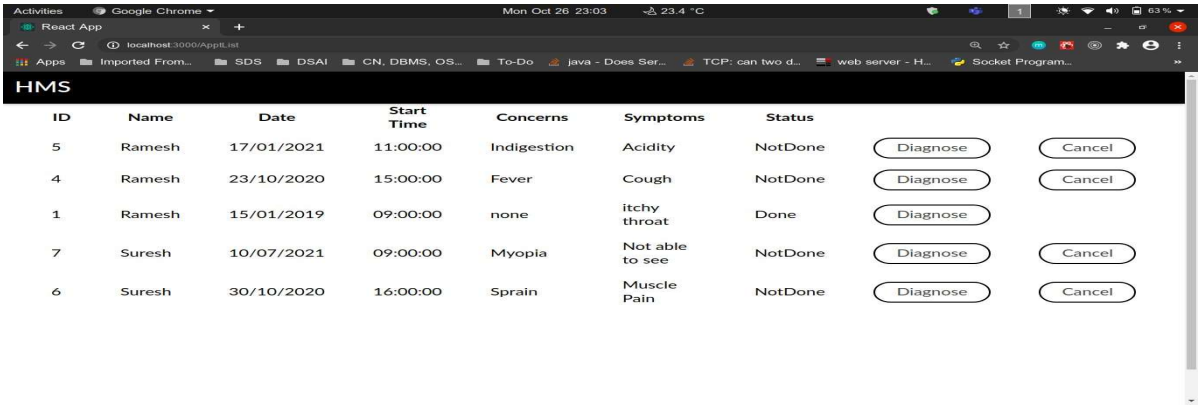






Date of Appointment	Start Time	End Time	Concerns	Symptoms	Status		
15/01/2019	09:00	10:00	none	itchy throat	Done	See Diagnosis	Delete
23/10/2020	15:00	16:00	Fever	Cough	NotDone	See Diagnosis	Cancel
17/01/2021	11:00	12:00	Indigestion	Acidity	NotDone	See Diagnosis	Cancel




Activities
Google Chrome
Mon Oct 26 23:03
23.4 °C
React App
localhost:3000/ApptList
Apps
Imported From...
SDS
DSAI
CN, DBMS, OS...
To-Do
java - Does Ser...
TCP: can two d...
web server - H...
Socket Program...
HMS
ID
Name
Date
Start Time
Concerns
Symptoms
Status
5 Ramesh 17/01/2021 11:00:00 Indigestion Acidity NotDone Diagnose Cancel
4 Ramesh 23/10/2020 15:00:00 Fever Cough NotDone Diagnose Cancel
1 Ramesh 15/01/2019 09:00:00 none itchy throat Done Diagnose
7 Suresh 10/07/2021 09:00:00 Myopia Not able to see NotDone Diagnose Cancel
6 Suresh 30/10/2020 16:00:00 Sprain Muscle Pain NotDone Diagnose Cancel

CONCLUSION

n conclusion, the development of a Hospital Management System (HMS) utilizing Database Management System (DBMS) technology offers a transformative solution to streamline hospital operations, enhance patient care delivery, and improve overall efficiency within healthcare facilities. By leveraging the capabilities of modern software tools and technologies, we have designed a comprehensive HMS that addresses the diverse needs of healthcare providers, administrators, and patients.

Through the implementation of a normalized database schema, we have organized healthcare data into tables and columns, reducing redundancy and dependency while ensuring data integrity and reliability. The functional requirements of the HMS encompass patient management, electronic medical records, appointment scheduling, billing and invoicing, pharmacy management, inventory control, laboratory management, reporting, and analytics. These functionalities empower healthcare professionals to efficiently manage patient information, appointments, medications, and treatments, thereby optimizing resource utilization and improving patient outcomes.

Furthermore, the HMS adheres to stringent non-functional requirements, including security, performance, scalability, reliability, usability, interoperability, compliance, and accessibility. By incorporating robust security measures, continuous monitoring, and adherence to healthcare regulations, the system safeguards patient data privacy and confidentiality. Scalable architecture, coupled with continuous integration and deployment pipelines, ensures the system can adapt to evolving healthcare needs and technological advancements.

REFERENCE

- React JS official documentation for React JS to understand its core concepts and features for building dynamic user interfaces.
  https://react.dev/blog/2023/03/16/introducing-react-dev

- Node.js documentation for guidance on building scalable and efficient server-side applications using JavaScript. https://devdocs.io/node/

- Express.js a minimalist web framework for Node.js, which is ideal for building robust APIs and web applications.
  https://devdocs.io/express/

  GITHUB: https://github.com/Aryan-Singh1026/DBMS-REPORT-PROJECT/blob/main/Hospital-Management-System-DBMS-master.zip

- Tailwind CSS to create modern and responsive UI designs using utility-first CSS framework.
  https://v2.tailwindcss.com/docs