GYM MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

SANTHOSH R

Register No:31022P08006

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

in

PG DEPARTMENT OF COMPUTER APPLICATIONS

GOVERNMENT THIRUMAGAL MILLS COLLEGE

GUDIYATTAM, VELLORE DIST - 632 602

MARCH/APRIL – 2024

GOVERNMENT THIRUMAGAL MILLS COLLEGE, GUDIYATTAM.
VELLORE DIST – 632 602

PG DEPARTMENT OF COMPUTER APPLICATIONS

PROJECT WORK

MARCH/APRIL - 2024

This is to certify that the project entitled

GYM MANAGEMENT SYSTEM

is the bonafide record of project work done by

SANTHOSH R

Register No:31022P08006

of MCA (PG Department of Computer Applications) during the year 2023 - 2024.

--------------------------	-----------------------------
Project Guide	Head of the Department

Submitted for the Project Viva-Voce examination held on ……………..

---------------------------	-----------------------------
1. External Examiner	2. External Examiner

DECLARATION

I affirm that the project work titled **GYM MANAGEMENT SYSTEM** being submitted in partial fulfillment for the award of **Master of Computer Applications** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

SANTHOSH R
31022P08006

I certify that the declaration made above by the candidate is true

Mrs. K. SATHIS KUMAR M.C.A., M.Phil

ACKNOWLEDGEMENT

I express my gratitude to our beloved Principal **Dr. G. KRISHNAN M.A., M.Phil., M.ED, Ph.D.,** Government Thirumagal Mills College Gudiyattam, for providing the resource facilities and encouraging gesture for completion of my project.

I wish to express my deep sense of gratitude and heartiest thanks to our professor **Dr. K. ARULANANDAM, MCA., M.Phil., Ph.D.,** Head of the Department of MCA, Government Thirumagal Mills College, Gudiyattam.

I am much privileged to have **Mrs. K. SATHIS KUMAR M.C.A., M.Phil** lecturer in MCA Department, as my guide for this project. I heartily thank her for her overall help and support at all stage.

I also take an opportunity to thank all the TEACHING and NON - TEACHING STAFF for their fullest and valuable assistance provided in this project work.

Last but not the least, I own my special thanks to my beloved all my friends and family member for their help in finishing this project successfully.

Place: Gudiyattam Submitted by

Date: **SANTHOSH R**

TABLE OF CONTENTS

ABSTRACT

This project aims to develop a comprehensive Gym Management System using HTML, PHP, and MySQL. The system is designed to streamline and automate various aspects of gym operations, including member management, trainer scheduling, equipment tracking, and financial transactions. The Gym Management System provides a user-friendly interface for both administrators and members. Administrators can manage member registrations, assign trainers, schedule classes, and monitor financial transactions. Members can easily sign up for memberships, view their schedules, make payments, and track their progress.

The system utilizes a MySQL database to store and manage data efficiently. PHP is used for server-side scripting to handle dynamic content generation and interaction with the database. HTML is employed for creating the user interface, ensuring compatibility across different web browsers.

Allows administrators to add, edit, and delete member profiles, including personal information, membership status, and attendance records. Enables administrators to assign trainers to specific classes or sessions based on availability and expertise. Provides a calendar-based interface for scheduling gym classes, including yoga, aerobics, and strength training. Allows administrators to manage gym equipment inventory, track usage, and schedule maintenance. Facilitates online payments for membership fees, class registrations, and personal training sessions. Generates invoices and receipts for transactions. Provides administrators with insights into gym performance, membership trends, class attendance, and financial summaries.

Overall, the Gym Management System aims to enhance the efficiency and effectiveness of gym operations while improving the experience for both administrators and members. By automating routine tasks and providing robust management tools, the system empowers gym.

CHAPTER – 1

PROJECT INFORMATION

1.1 INTRODUCTION:

Welcome to our Gym Management System! This system is designed to streamline the management of gym operations, making it easier for both administrators and members to handle various tasks efficiently. By integrating HTML for the front-end interface, PHP for server-side scripting, and MySQL for database management, we aim to provide a user-friendly and robust solution for gym management. Easily add, update, and delete member profiles. Keep track of member details such as name, contact information, membership status, and payment history. Manage gym classes efficiently by creating schedules, assigning instructors, and allowing members to sign up for classes online. Keep track of member attendance for gym sessions and classes to monitor member engagement and adherence to their fitness routines. Generate invoices, track payments, and manage membership subscriptions seamlessly. Automate billing reminders and notifications to ensure timely payments. Maintain an inventory of gym equipment, including details such as quantity, condition, and maintenance schedules. Receive alerts for equipment maintenance or replacement. Generate reports on various aspects of gym operations, including membership statistics, class attendance, revenue, and expenses. Analyze data to make informed decisions for business growth.

Our system utilizes a combination of HTML for creating the user interface, PHP for server-side scripting to handle dynamic content and interactions, and MySQL for database management to store and retrieve data efficiently. Here's a brief overview of the workflow: Members and administrators interact with the system through a user-friendly web interface created using HTML. The interface provides easy navigation and access to different features and functionalities. PHP scripts handle the processing of user requests and interactions. These scripts interact with the MySQL database to fetch or update information as required. MySQL is used to store all relevant data, including member profiles, class schedules, attendance records, billing information, and equipment inventory. The database ensures data integrity and allows for efficient retrieval and manipulation of information. The integration of HTML, PHP, and MySQL enables seamless communication between the front-end interface, server-side scripts, and database, ensuring smooth operation of the gym management system.

Our Gym Management System offers a comprehensive solution for efficiently managing various aspects of gym operations. Whether you're an administrator looking to streamline administrative tasks or a member seeking a convenient way to interact with the gym, our system is designed to meet your needs effectively. With its user-friendly interface and robust functionalities, we aim to enhance the overall experience of gym management for both administrators and members.

This introduction provides an overview of the system and its key features, highlighting how HTML, PHP, and MySQL are used to create a comprehensive gym management solution.

1.2 AIM OF THE SYSTEM:

A gym management system aims to streamline and enhance the operational aspects of running a fitness facility. It typically includes features such as member management, class scheduling, staff management, billing and invoicing, equipment tracking, and reporting. The primary goal is to improve efficiency, automate administrative tasks, enhance member experience, and optimize overall gym operations. By centralizing and automating these processes, gym owners and managers can focus more on delivering quality services, retaining members, and growing their business.

1.3 OBJECTIVES:

- The main objective of the project is to develop software that facilitates the data storage, data maintenance and its retrieval for the gym in an igneous way.
- To store the record of the customers, the staff that has the privileges to access, modify and delete any record and finally the service, gym provides to its customers.
- Also, only the staff has the privilege to access any database and make the required changes, if necessary.
- To develop easy-to-use software which handles the customer-staff relationship in an effective manner.
- To develop a user friendly system that requires minimal user training. Most of features and function are similar to those on any windows platform.

CHAPTER 2

REQUIREMENT AND ANALYSIS

2.1 WHAT IS REQUIREMENT ANALYSIS ?

Every application needs the software in which it has to be executed and a hardware the application is going to perform its function. Some application cannot run on every platforms and some applications needs some specific requirement in the software or in hardware to get operated. Lets take an example of the applications which cannot be run on every platforms like windows, android, Linux, etc. Applications made in visual basic is only supported for the windows, one cannot access this applications from the mobile phones, etc. So, here are some hardware and software specifications which are mandatory for the application to get operated.

Analysing Requirements:

This step helps to determine the quality of the requirements. It involves identifying whether the requirements are unclear, incomplete, ambiguous and contradictory.

Requirements Documentation:

This step involves documenting the requirements in various forms including summary list, natural language documents, visual documents, use-cases, or process specification. It includes different categories of requirement specification documents that are essential for designing this software.

2.2 PERSPECTIVE:

The Gym Management System is a web-based system. A design viewpoint in which the design target is a large organizational information system.

2.2.1 Interface:

There are two types of interface found in Gym Management System follows:

➔ **User Interface:**

- HTML has been used for developing the User Layout for the system.
- PHP Script has been used for creating all the validations and client side scripting functionality.
- CSS has been used for designing the web pages of the system.

➔ **Admin Interface:**

The Administrator is able to view the customer information, can complete the customer packages of the users, can update the services information, price, etc.

2.2.2 HARDWARE REQUIREMENTS:

The Gym management system shall provide minimum hardware requirements. The following hardware configurations are required for the PC for using the Gym management system.

□	Hard Disk	:	2GB or more of free space.
□	RAM	:	2 GB / 4 GB / 8 GB RAM.
□	Processor	:	Computer processor.

2.2.3 SOFTWARE REQUIREMENTS:

This section lists the requirements that are needed to run the system efficiently. The operating system needed for the system to run effectively, the interface to run application, the integrated development environment to develop the environment and the other tools used for editing purpose areas follow:

□	Operating System	:	Windows 10, MAC OS or Linux
□	Database	:	MySQL.
□	Platform Used	:	PHP
□	Editor Used	:	Notepad / Notepad ++.
□	Browser Supported	:	Google Chrome, Internet Explorer, Safari, Microsoft Edge.

2.2.4 COMMUNICATION PROTOCOL:

Following protocols are required to be permitted on the server side

- HTTP incoming request .
- HHTP response

2.3 SPECIFIC REQUIREMENTS:

This section contains detail about the system that are required for the designer to create a system to satisfy the user's requirements and for the testers to test the given requirements. This section contains the interface description for each GUI for the different system users. These sections also give description about all the system inputs, all the function performed by the system output (responses).

2.3.1 Functional Requirements:

The system runs of apache server so it is needed that server must have apache server version 2.0 available. We have used HTML for server side scripting so the current version of HTML must be available on the server:

- MySQL database has been used for storing the data of the website
- HTML has been used for creating the layout of the web application
- CSS has been used for creating the designing of the webpages
- PHP Script scripting language has been implemented on the system for performing all
- of the Client Side Server Validation.

2.3.2 MODULE DESCRIPTION:

User Module:

- Login Class: Used for performing all the operations of the login functionality.
- Page Class: Class for managing all the operations of the page.
- Traffic Class: Class for managing the traffic of the website.
- IP Class: It has been used for storing all the IPs which hits the website.
- Users Class: Class for managing all the user operations.
- Permission Class: This class has been used for managing all the permissions level opeations.

2.3.3 Non-Functional Requirements:

- Usability: These requirements focus on the fitness of the user interface and how people interact with it.
- Reliability: These requirements determine system availability that is able to workout 24*7 for 365 days if needed.
- Performance: The Gym management system provides the best performance as needed.
- Security: Security is required so that the customer information cannot be leaked and keep the data safe.
- Flexibility: This System is able to provide the flexibility to run in any browser and able to modify in every configuration.

2.4 PROCESS MODEL:

For this system, the Iterative model is selected.

What is an Iterative Model?

- In an iterative model, the iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to deploy.
- An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

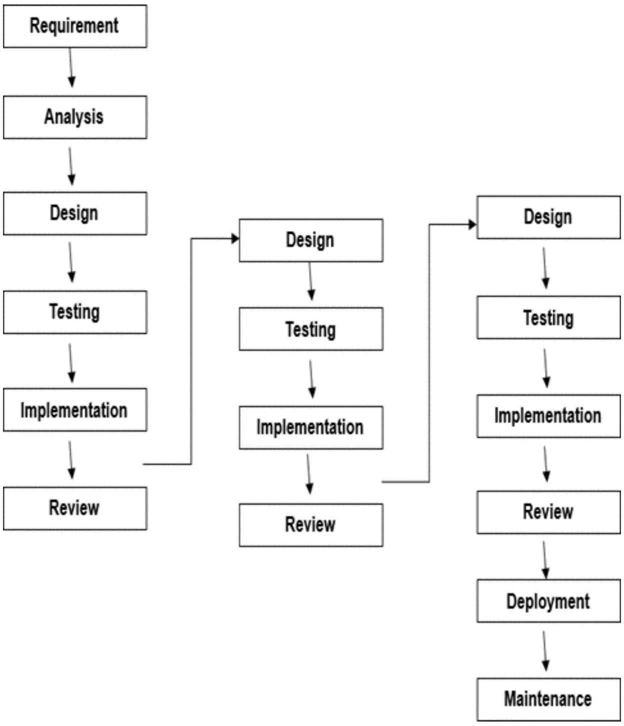
Iterative Model Design:

- Iterative process starts with a simple implementation of a subset of the requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.
- The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).
- Iterative and Incremental development is a combination of both iterative design or iterative method and incremental built model for development.

- “During software development, more than one iteration of the software development cycle may be in progress at the same time” and “This process may be described as an evolutionary acquisition or incremental built approach.
- In the incremental model the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases.
- The key to successful use of an iterative software development lifecycle is rigorous validation of requirements, verification and testing of each version of the software against those requirements within each cycle of the model.

Pictorial representation of Iterative Model:

Figure 1 : Iterative Model



Advantages:

- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change scope or requirements.
- Testing and debugging during smaller iteration are easy.

Disadvantages:

- More resources may be required.
- Although the cost of change is lesser, it is not very suitable for changing requirements.
- More management attention is required.
- Defining increments may require definition of complete system.

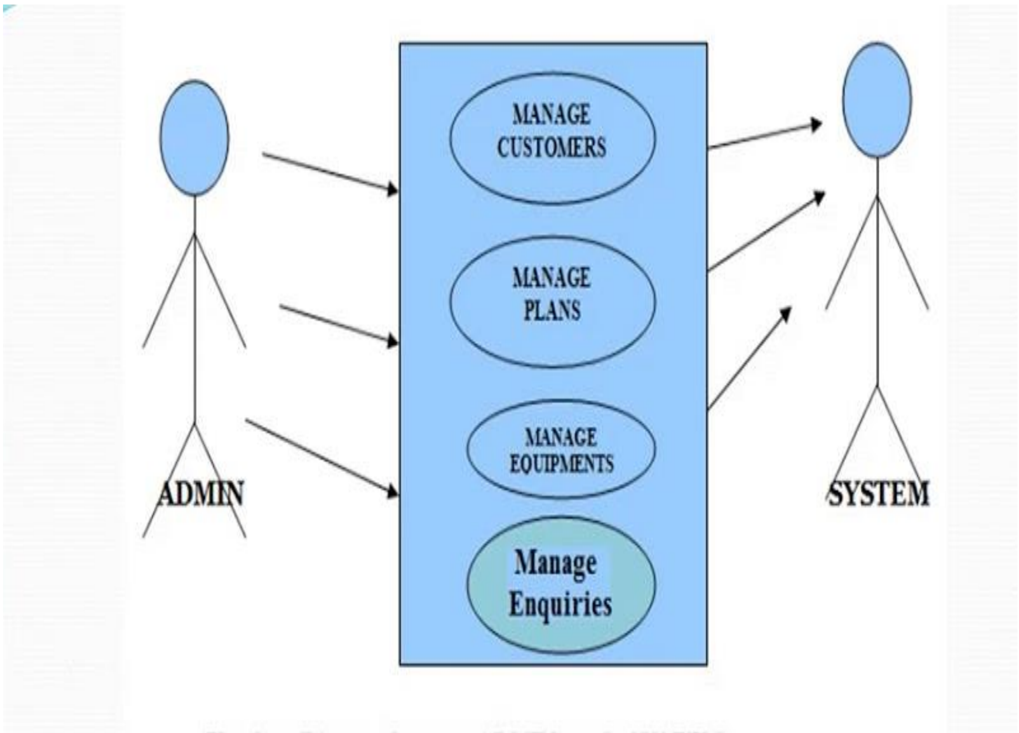
CHAPTER 3

SYSTEM DESIGN

Designing a gym management system involves several components such as user roles (Customer, Admin), use-case scenarios, activity diagrams, data dictionary, input/output designs, etc. Here's a brief overview of each component:

3.1 USE-CASE DIAGRAM:

A use-case diagram illustrates the interactions between different actors (such as Customers and Admins) and the system (Gym Management System). It helps in identifying the functionalities the system needs to provide.

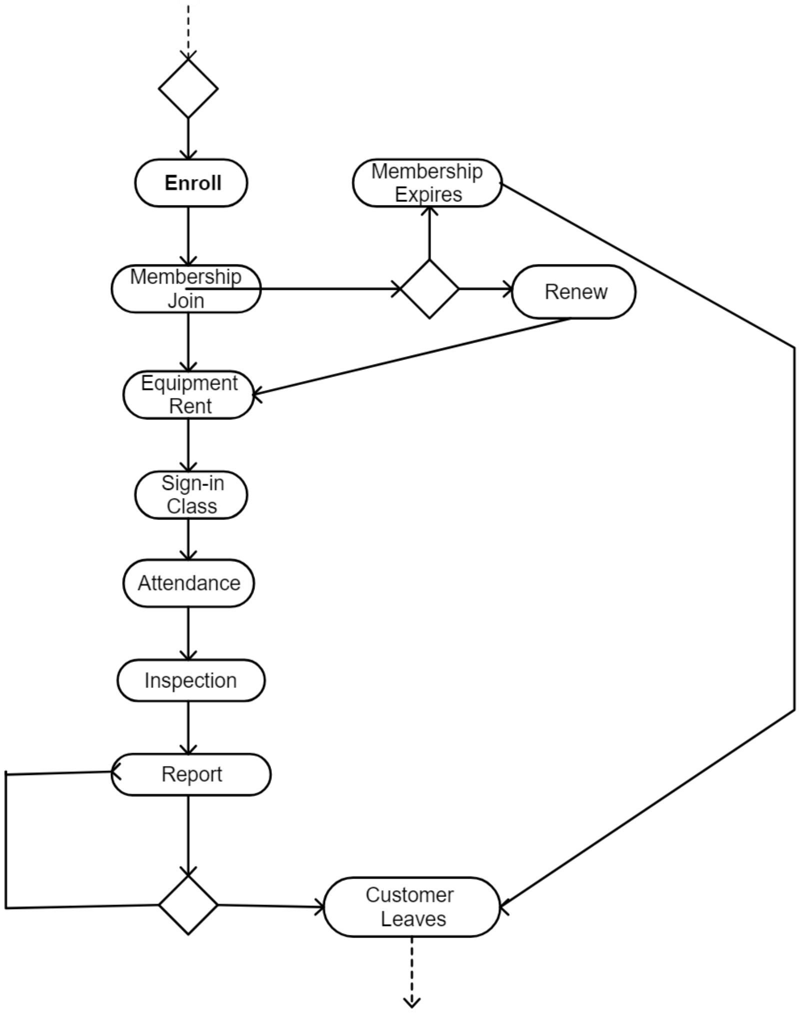


3.2 ACTIVITY DIAGRAM:

Activity diagrams represent the flow of activities within the system.

For example, it could outline the process of customer registration, booking

Sessions ,managing memberships, etc.



3.3 CUSTOMER AND ADMIN:

These are two distinct user roles in the system. Customers primarily interact with the system to book sessions, view schedules, manage memberships, etc., while Admins have access to functionalities such as managing user accounts, scheduling classes, viewing reports, etc.

3.4 DATA DICTIONARY:

This includes a detailed description of all the data entities used in the system. For example, it may include tables for customer information, session schedules, membership plans, payment details, etc.

3.5 INPUT/OUTPUT DESIGN:

This includes designing the forms and screens where users input data and receive outputs. For example, the input design might involve creating forms for customer registration, session booking, etc. The output design may include screens for displaying schedules, membership details, payment confirmation, etc.

3.5.1 INPUT DESIGN:

This involves designing forms and interfaces for users to input data into the system. For example, forms for customer registration, session booking, etc., need to be designed in a user-friendly manner.

3.5.2 OUTPUT DESIGN:

This involves designing screens and interfaces for the system to display information to users. For example, screens for displaying session schedules, membership details, payment confirmation, etc., need to be designed in a clear and organized manner.

When implementing this system using PHP, you would create the necessary scripts and database interactions to handle user input, process data, and generate appropriate outputs based on the user's actions. Additionally, you would need to ensure security measures are in place to protect user data and system integrity.

CHAPTER 4

TESTING IMPLEMENTATION

Testing Implementation for a gym management system involves verifying that the system functions correctly and meets its requirements. Here's an overview of the testing approach, including different types of testing and test cases:

4.1 TESTING APPROACH:

4.1.1 Black Box Testing:

This involves testing the functionality of the system without knowledge of its internal structure. Testers interact with the system through its user interfaces and test its inputs and outputs.

4.1.2 White Box Testing:

This involves testing the internal workings of the system, including code-level testing. Testers have knowledge of the system's internal structure and use this knowledge to design test cases that ensure all code paths are tested.

4.1.3 Gray Box Testing:

This is a combination of black box and white box testing. Testers have partial knowledge of the system's internal structure and use it to design comprehensive test cases.

4.2 TEST CASES:

4.2.1 Customer Registration:

- Black Box: Verify that customers can successfully register with valid information.
- White Box: Test different scenarios such as valid/invalid input fields, database connectivity, error handling, etc.
- Session Booking:
- Black Box: Test if customers can book sessions based on availability and membership status.
- White Box: Test session booking logic, database updates, and error handling.
- Admin Functionality:
- Black Box: Verify that admins can add/edit/delete sessions, manage memberships, view reports, etc.
- White Box: Test admin interfaces, database operations, and access control.

4.2.2 Membership Management:

- Black Box: Test membership creation, renewal, and cancellation processes.
- White Box: Verify database updates, calculation of membership durations, and notifications.
- Payment Processing:
- Black Box: Test payment processing for membership fees, session bookings, etc.
- White Box: Verify payment gateway integration, transaction handling, and error recovery.
- Reporting:
- Black Box: Test generation and viewing of reports such as attendance, revenue, etc.
- White Box: Verify data retrieval, report formatting, and access control.

4.3 IMPLEMENTATION APPROACH:

- Modular Development: Divide the system into modules such as user management, session scheduling, membership management, etc., and develop each module separately.
- Database Design: Design a robust database schema to store customer information, session details, membership data, etc.
- Security Measures: Implement security measures such as input validation, encryption of sensitive data, role-based access control, etc.
- User-Friendly Interface: Design intuitive user interfaces for both customers and admins to facilitate easy interaction with the system.
- Continuous Testing: Perform testing at each stage of development to catch bugs early and ensure the system meets its requirements.

By following these approaches, you can ensure that the gym management system is implemented effectively and thoroughly tested to meet user needs and expectations

CHAPTER 5

CONCLUSION

5.1 CONCLUSION:

In conclusion, the gym management system offers an efficient solution for managing various aspects of a fitness facility, including customer registration, session booking, membership management, and administrative tasks. Through the use of PHP and appropriate design principles, the system provides a user-friendly interface for both customers and administrators. The implementation of modular development, database design, security measures, and continuous testing ensures the system's reliability and functionality.

5.2 LIMITATIONS:

- Despite its advantages, the gym management system may have some limitations, including:
- Scalability: The system may face challenges in handling a large volume of users or data as the gym grows.
- Integration: Integration with external systems or services, such as payment gateways or fitness equipment, may pose challenges.
- Customization: Limited customization options for specific gym requirements or workflows.
- Accessibility: Accessibility features for users with disabilities may not be fully implemented.
- Performance: Performance issues may arise under heavy usage or concurrent access.

5.3 FUTURE SCOPE:

To address the limitations and enhance the system's capabilities, several future enhancements and scope expansions can be considered, such as:

- Scalability Enhancements: Implementing scalable architecture and optimizing database performance to handle increased user loads.
- Integration with IoT Devices: Integrating with IoT devices like fitness trackers or smart equipment to provide personalized workout experiences and track user progress.
- Mobile Application Development: Developing a mobile app for convenient access to the system's functionalities on smartphones or tablets.

- Advanced Analytics: Incorporating advanced analytics features to analyze user data, predict trends, and optimize business operations.Customization Options: Offering customizable features to tailor the system according to individual gym requirements and workflows.
- Enhanced Security Measures: Strengthening security measures to protect user data and prevent unauthorized access or data breaches.
- Accessibility Improvements: Ensuring compliance with accessibility standards and enhancing features for users with disabilities.
- Social Integration: Integrating social media features to foster community engagement and encourage user interaction.

By continuously evolving and adapting to meet the evolving needs of gyms and fitness centers, the gym management system can remain a valuable tool for streamlining operations, enhancing user experiences, and driving business growth.

CHAPTER 6

APPENDIX

6.1 SOURCE CODE:

index.php

```
<?php
session_start();
error_reporting(0);
include 'include/config.php';
$uid=$_SESSION['uid'];

if(isset($_POST['submit']))
{
$pid=$_POST['pid'];
$sql="INSERT INTO tblbooking (package_id,userid) Values(:pid,:uid)";
$query = $dbh -> prepare($sql);
$query->bindParam(':pid',$pid,PDO::PARAM_STR);
$query->bindParam(':uid',$uid,PDO::PARAM_STR);
$query -> execute();
echo "<script>alert('Package has been booked.');</script>";
echo "<script>window.location.href='booking-history.php'</script>";
}
?>
<!DOCTYPE html>
<html lang="zxx">
<head>
	<title>Gym Management System</title>
	<meta charset="UTF-8">
	<meta name="description" content="Ahana Yoga HTML Template">
	<meta name="keywords" content="yoga, html">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<!-- Stylesheets -->
	<link rel="stylesheet" href="css/bootstrap.min.css"/>
	<link rel="stylesheet" href="css/font-awesome.min.css"/>
	<link rel="stylesheet" href="css/owl.carousel.min.css"/>
	<link rel="stylesheet" href="css/nice-select.css"/>
	<link rel="stylesheet" href="css/magnific-popup.css"/>
	<link rel="stylesheet" href="css/slicknav.min.css"/>
	<link rel="stylesheet" href="css/animate.css"/>
	<!-- Main Stylesheets -->
	<link rel="stylesheet" href="css/style.css"/>
</head>
<body>
	<!-- Page Preloder -->
	<!-- Header Section -->
	<?php include 'include/header.php';?>
	<!-- Header Section end -->
	<!-- Page top Section -->
	<section class="page-top-section set-bg" data-setbg="img/page-top-bg.jpg">
		<div class="container">
			<div class="row">
```

```
                    <div class="col-lg-7 m-auto text-white">
                        <h2>Home</h2>
                        <p>Physical Activity Or Can Improve Your Health</p>
                    </div>
                </div>
            </div>
        </section>
        <!-- Pricing Section -->
        <section class="pricing-section spad">
            <div class="container">
                <div class="section-title text-center">
                    <img src="img/icons/logo-icon.png" alt="">
                    <h2>Pricing plans</h2>
                    <p>Practice Yoga to perfect physical beauty, take care of your
soul and enjoy life more fully!</p>
                </div>
                <div class="row">
 <?php
$sql ="SELECT id, category, titlename, PackageType, PackageDuratiobn, Price, uploadphoto,
Description, create_date from tbladdpackage";
$query= $dbh -> prepare($sql);
$query-> execute();
$results = $query -> fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query -> rowCount() > 0)
```

```
{
foreach($results as $result)
{
?>
<div class="col-lg-3 col-sm-6">
<div class="pricing-item begginer">
<div class="pi-top">
<h4><?php echo $result->titlename;?></h4>
</div>
<div class="pi-price">
<h3><?php echo htmlentities($result->Price);?></h3>
<p><?php echo $result->PackageDuratiobn;?></p>
</div>
<ul>
<?php echo $result->Description;?>
</ul>
<?php if(strlen($_SESSION['uid'])==0): ?>
<a href="login.php" class="site-btn sb-line-gradient">Booking Now</a>
<?php else :?>
<!-- <a href="#" class="site-btn sb-line-gradient">Booking Now</a> -->
 <form method='post'>
             <input type='hidden' name='pid' value='<?php echo htmlentities($result->id);?>'>
              <input class='site-btn sb-line-gradient' type='submit' name='submit' value='Booking Now' onclick="return confirm('Do you really want to book this package.');">
```

```
                </form>
                                                    <?php endif;?>
                                        </div>
                                    </div>
                                    <?php  $cnt=$cnt+1; } } ?>
                            </div>
                </div>
        </section>
        <!-- Footer Section -->
        <?php include 'include/footer.php'; ?>
        <!-- Footer Section end -->
        <div class="back-to-top"><img src="img/icons/up-arrow.png" alt=""></div>
        <!-- Search model end -->
        <!--====== Javascripts & Jquery ======-->
        <script src="js/vendor/jquery-3.2.1.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <script src="js/jquery.slicknav.min.js"></script>
        <script src="js/owl.carousel.min.js"></script>
        <script src="js/jquery.nice-select.min.js"></script><script src="js/jquery-ui.min.js"></script>
        <script src="js/jquery.magnific-popup.min.js"></script>
        <script src="js/main.js"></script>
        </body>
</html>
```

admin.php

```
<?php  session_start();
error_reporting(0);
include  'include/config.php';
if (strlen($_SESSION['adminid']==0)) {
  header('location:logout.php');
  } else{
?>
<!DOCTYPE html>
<html lang="en">
  <head>
   <title>Admin | Dashboard</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Main CSS-->
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <!-- Font-icon css-->
    <link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
  </head>
  <body class="app sidebar-mini rtl">
    <!-- Navbar-->
    <?php include 'include/header.php'; ?>
```

```
<!-- Sidebar menu-->
<div class="app-sidebar__overlay" data-toggle="sidebar"></div>
<?php include 'include/sidebar.php'; ?>
<main class="app-content">
  <div class="app-title">
    <div>
      <h1><i class="fa fa-dashboard"></i> Dashboard</h1>
    </div>
    <ul class="app-breadcrumb breadcrumb">
      <li class="breadcrumb-item"><i class="fa fa-home fa-lg"></i></li>
      <li class="breadcrumb-item"><a href="#">Dashboard</a></li>
    </ul>
  </div>
  <div class="row">
    <div class="col-md-6 col-lg-6">
      <?php
              $sql="SELECT count(id) as totalcat FROM tblcategory;";
              $query= $dbh->prepare($sql);
              $query-> execute();
              $results = $query -> fetchAll(PDO::FETCH_OBJ);
              foreach($results as $result)
              {
              ?>
                   <a href="add-category.php">
```

```
        <div class="widget-small info coloured-icon"><i class="icon fa fa-files-o fa-3x"></i>
          <div class="info">
            <h4>Listed Categories</h4>
            <p><b><?php echo $result->totalcat;?></b></p>
          </div>
        </div></a>
          <?php  } ?>
      </div>
<div class="col-md-6 col-lg-6">
        <?php
                $sql="SELECT count(id) as totalpackagetype FROM tblcategory;";
                $query= $dbh->prepare($sql);
                $query-> execute();
                $results = $query -> fetchAll(PDO::FETCH_OBJ);
                foreach($results as $result)
                {
                ?>
                     <a href="add-package.php">
   <div class="widget-small primary coloured-icon"><i class="icon fa fa-files-o fa-3x"></i>
          <div class="info">
            <h4>Listed Package Type</h4>
            <p><b><?php echo $result->totalpackagetype;?></b></p>
          </div>
        </div></a>
```

```
    <?php  } ?>
</div>
<div class="col-md-6 col-lg-6">
  <?php
          $sql="SELECT count(id) as totalpost FROM tbladdpackage;";
          $query= $dbh->prepare($sql);
          $query-> execute();
          $results = $query -> fetchAll(PDO::FETCH_OBJ);
          $cnt=1;
          if($query -> rowCount() > 0)
          {
          foreach($results as $result)
          {
          ?>
           <a href="manage-post.php">
  <div class="widget-small primary coloured-icon"><i class="icon fa fa-file fa-3x"></i>
    <div class="info">
      <h4>Listed Packages</h4>
      <p><b><?php echo $result->totalpost;?></b></p>
    </div>
  </div>
</a>
    <?php  $cnt=$cnt+1; } } ?>
</div>
```

```
        <div class="col-md-6 col-lg-6">
          <?php
                $sql="SELECT count(id) as totalbookings FROM tblbooking;";
                $query= $dbh->prepare($sql);
                $query-> execute();
                $results = $query -> fetchAll(PDO::FETCH_OBJ);
                foreach($results as $result)
                {
                ?>
                <a href="booking-history.php">
          <div class="widget-small info coloured-icon"><i class="icon fa fa-users fa-3x"></i>
            <div class="info">
              <h4>Total Bookings</h4>
              <p><b><?php echo $result->totalbookings;?></b></p>
            </div>
          </div>
        </a>
            <?php  } ?>
        </div>
      <div class="col-md-6 col-lg-6">
          <?php $sql="SELECT count(id) as totalbookings FROM tblbooking where paymentType is null or paymentType="";
                $query= $dbh->prepare($sql);
                $query-> execute();
```

```
         $results = $query -> fetchAll(PDO::FETCH_OBJ);
         foreach($results as $result)
         {
         ?>
         <a href="new-bookings.php">
     <div class="widget-small danger coloured-icon"><i class="icon fa fa-user fa-3x"></i>
      <div class="info">
       <h4>New Bookings</h4>
       <p><b><?php echo $result->totalbookings;?></b></p>
      </div>
     </div>
    </a>
      <?php  } ?>
    </div>
  <div class="col-md-6 col-lg-6">
     <?php
         $sql="SELECT count(id) as totalbookings FROM tblbooking where paymentType='Partial Payment'";
         $query= $dbh->prepare($sql);
         $query-> execute();
         $results = $query -> fetchAll(PDO::FETCH_OBJ);
         foreach($results as $result)
         ?>
         <a href="partial-payment-bookings.php">
    <div class="widget-small warning coloured-icon"><i class="icon fa fa-user fa-3x"></i>
      <div class="info">
        <h4>Partial Payment Bookings</h4>
        <p><b><?php echo $result->totalbookings;?></b></p>
      </div>
    </div>
  </a>
      <?php  } ?>
  </div>
   <div class="col-md-6 col-lg-6">
    <?php
            $sql="SELECT count(id) as totalbookings FROM tblbooking where paymentType='Full Payment'";
            $query= $dbh->prepare($sql);
            $query-> execute();
            $results = $query -> fetchAll(PDO::FETCH_OBJ);
            foreach($results as $result)
            {
            ?>
            <a href="full-payment-bookings.php">
    <div class="widget-small primary coloured-icon"><i class="icon fa fa-user fa-3x"></i>
      <div class="info">
        <h4>Full Payment Bookings</h4>
        <p><b><?php echo $result->totalbookings;?></b></p>
```

```
              </div>
            </div>
          </a>
              <?php  } ?>
          </div>
        </div>
      </main>
      <!-- Essential javascripts for application to work-->
      <script src="js/jquery-3.2.1.min.js"></script>
      <script src="js/popper.min.js"></script>
      <script src="js/bootstrap.min.js"></script>
      <script src="js/main.js"></script>
      <!-- The javascript plugin to display page loading on top-->
      <script src="js/plugins/pace.min.js"></script>
      <!-- Page specific javascripts-->
      <!-- Data table plugin-->
      <script type="text/javascript" src="js/plugins/jquery.dataTables.min.js"></script>
      <script type="text/javascript" src="js/plugins/dataTables.bootstrap.min.js"></script>
      <script type="text/javascript">$('#sampleTable').DataTable();</script>
    </body>
  </html>
  <?php } ?>
```

booking.php:

```
<?php session_start();
error_reporting(0);
require_once('include/config.php');
if(strlen( $_SESSION["uid"])==0)
    {
header('location:login.php');
}
else{
$uid=$_SESSION['uid'];
?>
<!DOCTYPE html>
<html lang="zxx">
<head>
        <title>User | Booking History</title>
        <meta charset="UTF-8">
        <meta name="description" content="Ahana Yoga HTML Template">
        <meta name="keywords" content="yoga, html">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <!-- Stylesheets -->
        <link rel="stylesheet" href="css/bootstrap.min.css"/>
        <link rel="stylesheet" href="css/font-awesome.min.css"/>
        <link rel="stylesheet" href="css/owl.carousel.min.css"/>
        <link rel="stylesheet" href="css/nice-select.css"/>
```

```
		<link rel="stylesheet" href="css/slicknav.min.css"/>

		<!-- Main Stylesheets -->

		<link rel="stylesheet" href="css/style.css"/>

</head>

<body>

		<!-- Page Preloder -->

		<!-- Header Section -->

		<?php include 'include/header.php';?>

		<!-- Header Section end -->

		<!-- Page top Section -->

		<section class="page-top-section set-bg" data-setbg="img/page-top-bg.jpg">

				<div class="container">

						<div class="row">

								<div class="col-lg-7 m-auto text-white">

										<h2>Booking History</h2>


								</div>

						</div>

				</div>

		</section>

		<!-- Page top Section end -->

		<!-- Contact Section -->

		<section class="contact-page-section spad overflow-hidden">

				<div class="container">
```

```
                        <div class="row">


                                <div class="col-lg-12">

                                            <table class="table table-hover table-bordered">

                <thead>

                 <?php $bookindid=$_GET['bookingid'];

                $sql="SELECT t1.id as bookingid,t3.fname as Name, t3.email as
email,t1.booking_date as bookingdate,t2.titlename as title,t2.PackageDuratiobn as
PackageDuratiobn,

t2.Price as Price,t2.Description as Description,t4.category_name as
category_name,t5.PackageName as PackageName,payment,paymentType FROM tblbooking
as t1

 join tbladdpackage as t2

on t1.package_id =t2.id

join tbluser as t3

on t1.userid=t3.id

join tblcategory as t4

on t2.category=t4.id

join tblpackage as t5

on t2.PackageType=t5.id

 where t1.id=:bookindid";

                $query= $dbh->prepare($sql);

                $query->bindParam(':bookindid',$bookindid, PDO::PARAM_STR);

                $query-> execute();

                $results = $query -> fetchAll(PDO::FETCH_OBJ);
```

```
$cnt=1;
if($query -> rowCount() > 0)
{
foreach($results as $result)
{
?>
<tr>
 <th>Booking Date</th>
 <td><?php echo $result->bookingdate; ?></td>
  <th>Name</th>
  <td><?php echo $result->Name; ?></td>
</tr>
<tr>
 <th>Email</th>
 <td><?php echo $result->email; ?></td>
  <th>Category</th>
  <td><?php echo $result->category_name; ?></td>
</tr>
<tr>
 <th>Package Name:</th>
 <td><?php echo $result->PackageName; ?></td>
  <th>Title</th>
  <td><?php echo $result->title; ?></td>
</tr>
```

```
                <tr>
                 <th>Package Duratiobn</th>
                 <td><?php echo $result->PackageDuratiobn; ?></td>
                  <th>Price</th>
                  <td><?php echo $result->Price; ?></td>
                  <?php $pricess=$result->Price; ?>
                </tr>
                <tr>
                 <th>Description</th>
                 <td colspan="3"><?php echo $result->Description; ?></td>

                </tr>

                <tr>
                 <th>PaymentType</th>
                 <td colspan="3"><?php  $ptype=$result->paymentType;
if($ptype==''):
        echo "Payment not made yet";
else:
        echo $ptype;
endif;
               ?></td>

                </tr>
```

```
            <?php  $cnt=$cnt+1; } } ?>
          </thead>
        </table>

        <?php   $sql="SELECT * from tblpayment
 where bookingID=:bookindid";
            $query= $dbh->prepare($sql);
            $query->bindParam(':bookindid',$bookindid, PDO::PARAM_STR);
            $query-> execute();
            $results = $query -> fetchAll(PDO::FETCH_OBJ);
            $cnt=1;
            if($query -> rowCount() > 0)
            { ?>
                <table class="table table-hover table-bordered">
                 <tr>
                   <th        colspan="3"        style="text-align:center;font-size:20px;">Payment
History</th>
                 </tr>
                 <tr>
                   <th>Payment Type</th>
                   <th>Amount Paid</th>
                   <th>Payment Date</th>
                 </tr>
 <?php foreach($results as $result)
```

```
                 { ?>
<tr>
  <td><?php echo $result->paymentType; ?></td>
  <td><?php echo $tpayment=$result->payment; ?></td>
  <td><?php echo $result->payment_date; ?></td>
</tr>
<?php
$gpayment+=$tpayment;
}  ?>
<tr>
  <th>Total</th>
  <th><?php echo $gpayment;?></th>
</tr>
                       </table>
                     <?php } ?>
                                              </div>


                                  </div>
                        </div>
            </section>
            <!-- Trainers Section end -->
            <!-- Footer Section -->
<?php include 'include/footer.php'; ?>
            <!-- Footer Section end -->
```

```
        <div class="back-to-top"><img src="img/icons/up-arrow.png" alt=""></div>
        <!--====== Javascripts & Jquery ======-->
        <script src="js/vendor/jquery-3.2.1.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <script src="js/jquery.slicknav.min.js"></script>
        <script src="js/owl.carousel.min.js"></script>
        <script src="js/jquery.nice-select.min.js"></script>
        <script src="js/jquery-ui.min.js"></script>
        <script src="js/jquery.magnific-popup.min.js"></script>
        <script src="js/main.js"></script>
        </body>
</html>
 <style>
.errorWrap {
    padding: 10px;
    margin: 0 0 20px 0;
    background: #dd3d36;
    color:#fff;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
.succWrap{
    padding: 10px;
margin: 0 0 20px 0;
    background: #5cb85c;
    color:#fff;
    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
}
        </style>
        <?php } ?>
```

registration.php

```
<?php
error_reporting(0);
require_once('include/config.php');
if(isset($_POST['submit']))
{
$fname=$_POST['fname'];
$lname=$_POST['lname'];
$mobile=$_POST['mobile'];
$email=$_POST['email'];
$state=$_POST['state'];
$city=$_POST['city'];
$Password=$_POST['password'];
$pass=md5($Password);
$RepeatPassword = $_POST['RepeatPassword'];
}



// Email id Already Exit
$usermatch=$dbh->prepare("SELECT mobile,email FROM tbluser WHERE (email=:usreml
|| mobile=:mblenmbr)");
$usermatch->execute(array(':usreml'=>$email,':mblenmbr'=>$mobile));
while($row=$usermatch->fetch(PDO::FETCH_ASSOC))
{
$usrdbeml= $row['email'];
$usrdbmble=$row['mobile'];
}
if(empty($fname))
{
  $nameerror="Please Enter First Name";
}

 else if(empty($mobile))
 {
 $mobileerror="Please Enter Mobile No";
 }
 else if(empty($email))
 {
 $emailerror="Please Enter Email";
 }
```

```
else if($email==$usrdbeml || $mobile==$usrdbmble)
 {
  $error="Email Id or Mobile Number Already Exists!";
 }
  else if($Password=="" || $RepeatPassword=="")
 {


   $error="Password And Confirm Password Not Empty!";


 }
 else if($_POST['password'] != $_POST['RepeatPassword'])
 {


   $error="Password And Confirm Password Not Matched";
 }
else{
$sql="INSERT INTO tbluser (fname,lname,email,mobile,state,city,password) Values(:fname,:lname,:email,:mobile,:state,:city,:Password)";


$query = $dbh -> prepare($sql);
$query->bindParam(':fname',$fname,PDO::PARAM_STR);
$query->bindParam(':lname',$lname,PDO::PARAM_STR);
$query->bindParam(':email',$email,PDO::PARAM_STR);
$query->bindParam(':mobile',$mobile,PDO::PARAM_STR);
$query->bindParam(':state',$state,PDO::PARAM_STR);
$query->bindParam(':city',$city,PDO::PARAM_STR);
$query->bindParam(':Password',$pass,PDO::PARAM_STR);

$query -> execute();
$lastInsertId = $dbh->lastInsertId();
if($lastInsertId>0)
{
echo "<script>alert('Registration successfull. Please login');</script>";
echo "<script> window.location.href='login.php';</script>";
}
else
{
$error ="Registration Not successfully";
 }
}
 }

 ?>
<!DOCTYPE html>
<html lang="zxx">
<head>
        <title>Gym Management System</title>
        <meta charset="UTF-8">
```

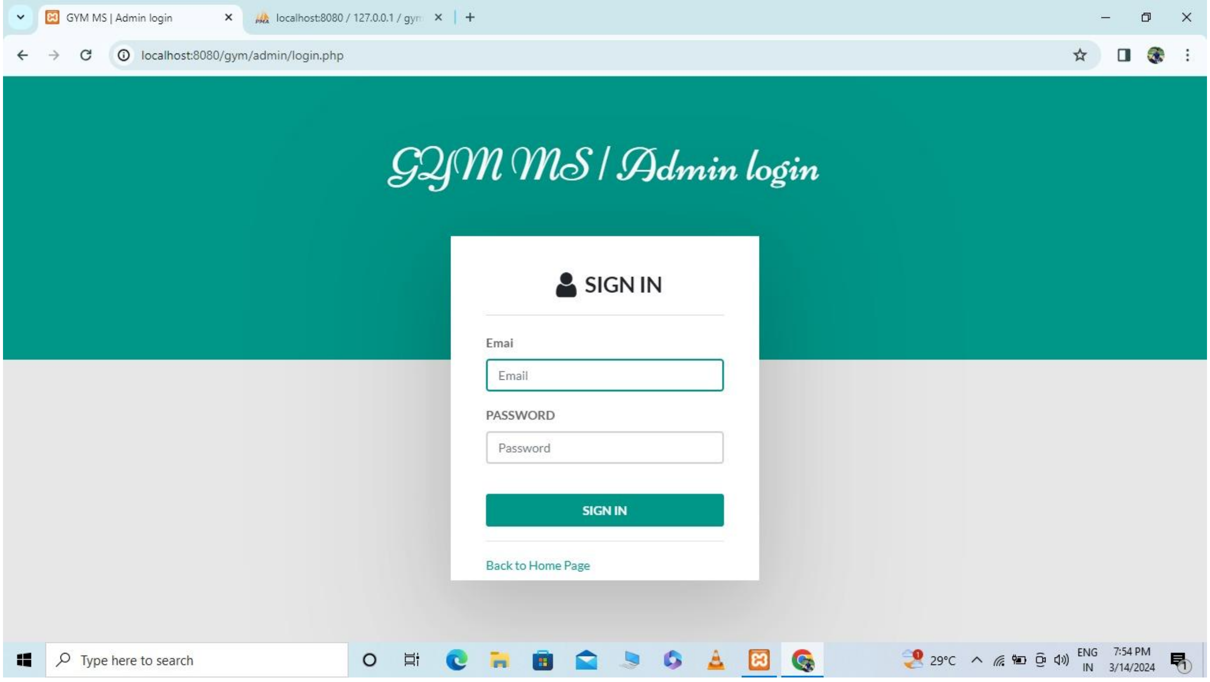
```
    <!-- Stylesheets -->
    <link rel="stylesheet" href="css/bootstrap.min.css"/>
    <link rel="stylesheet" href="css/font-awesome.min.css"/>
    <link rel="stylesheet" href="css/owl.carousel.min.css"/>
    <link rel="stylesheet" href="css/nice-select.css"/>
    <link rel="stylesheet" href="css/slicknav.min.css"/>

    <!-- Main Stylesheets -->
    <link rel="stylesheet" href="css/style.css"/>

</head>
<body>
    <!-- Page Preloder -->


    <!-- Header Section -->
    <?php include 'include/header.php';?>
    <!-- Header Section end -->

    <!-- Page top Section -->
    <section class="page-top-section set-bg" data-setbg="img/page-top-bg.jpg">
        <div class="container">
            <div class="row">
                <div class="col-lg-7 m-auto text-white">
```

```
                    <h2>Registration</h2>
                </div>
            </div>
        </div>
    </section>
    <!-- Page top Section end -->

    <!-- Contact Section -->
    <section class="contact-page-section spad overflow-hidden">
        <div class="container">

            <div class="row">
                <div class="col-lg-2">
                </div>
                <div class="col-lg-8">
                    <?php if($error){?><div class="errorWrap"><strong>ERROR</strong>:<?php echo htmlentities($error); ?> </div><?php }
    else if($succmsg){?><div class="succWrap"><strong>SUCCESS</strong>:<?php echo htmlentities($succmsg); ?> </div><?php }?><br><br>
                    <form class="singup-form contact-form" method="post">
                        <div class="row">
                            <div class="col-md-6">
                                <input type="text" name="fname" id="fname" placeholder="First Name" autocomplete="off" value="<?php echo $fname;?>" required>
                            </div>
                            <div class="col-md-6">
                                <input type="text" name="lname" id="lname" placeholder="Last Name" autocomplete="off" value="<?php echo $lname;?>" required>
                            </div>
                            <div class="col-md-6">
                                <input type="text" name="email" id="email" placeholder="Your Email" autocomplete="off" value="<?php echo $email;?>" required>
                            </div>
                            <div class="col-md-6">
                                <input type="text" name="mobile" id="mobile" maxlength="10" placeholder="Mobile Number" autocomplete="off" value="<?php echo $mobile;?>" required>
                            </div>
                            <div class="col-md-6">
                                <input type="text" name="state" id="state" placeholder="Your State" autocomplete="off" value="<?php echo $state;?>" required>
                            </div>
                            <div class="col-md-6">
                                <input type="text" name="city" id="city" placeholder="Your City" autocomplete="off" value="<?php echo $city;?>" required>
```

```
                            </div>

                            <div class="col-md-6">

                                <input type="password"
name="password" id="password" placeholder="Password" autocomplete="off">

                            </div>

                            <div class="col-md-6">

                                <input type="password"
name="RepeatPassword" id="RepeatPassword" placeholder="Confirm Password"
autocomplete="off" required>

                            </div>

                            <div class="col-md-4">

                        <input type="submit" id="submit"
name="submit" value="Register Now" class="site-btn sb-gradient">


                            </div>

                        </div>

                    </form>

                </div>

                <div class="col-lg-2">

                </div>

            </div>

        </div>

    </section>

    <!-- Trainers Section end -->
```

```
	<!-- Footer Section -->
<?php include 'include/footer.php'; ?>
	<!-- Footer Section end -->

	<div class="back-to-top"><img src="img/icons/up-arrow.png" alt=""></div>

	<!--====== Javascripts & Jquery ======-->
	<script src="js/vendor/jquery-3.2.1.min.js"></script>
	<script src="js/bootstrap.min.js"></script>
	<script src="js/jquery.slicknav.min.js"></script>
	<script src="js/owl.carousel.min.js"></script>
	<script src="js/jquery.nice-select.min.js"></script>
	<script src="js/jquery-ui.min.js"></script>
	<script src="js/jquery.magnific-popup.min.js"></script>
	<script src="js/main.js"></script>

	</body>
</html>
 <style>
.errorWrap {
    padding: 10px;
    margin: 0 0 20px 0;
    background: #dd3d36;

    color:#fff;

    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

}

.succWrap{

    padding: 10px;

    margin: 0 0 20px 0;

    background: #5cb85c;

    color:#fff;

    -webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

    box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);

}

</style>
```

6.2 SYSTEM SNAPCHATS:


GYM MS | Admin login
localhost:8080 / 127.0.0.1 / gym
localhost:8080/gym/admin/login.php
Type here to search
29°C
ENG IN
7:54 PM 3/14/2024

GYM MS | Admin login

SIGN IN

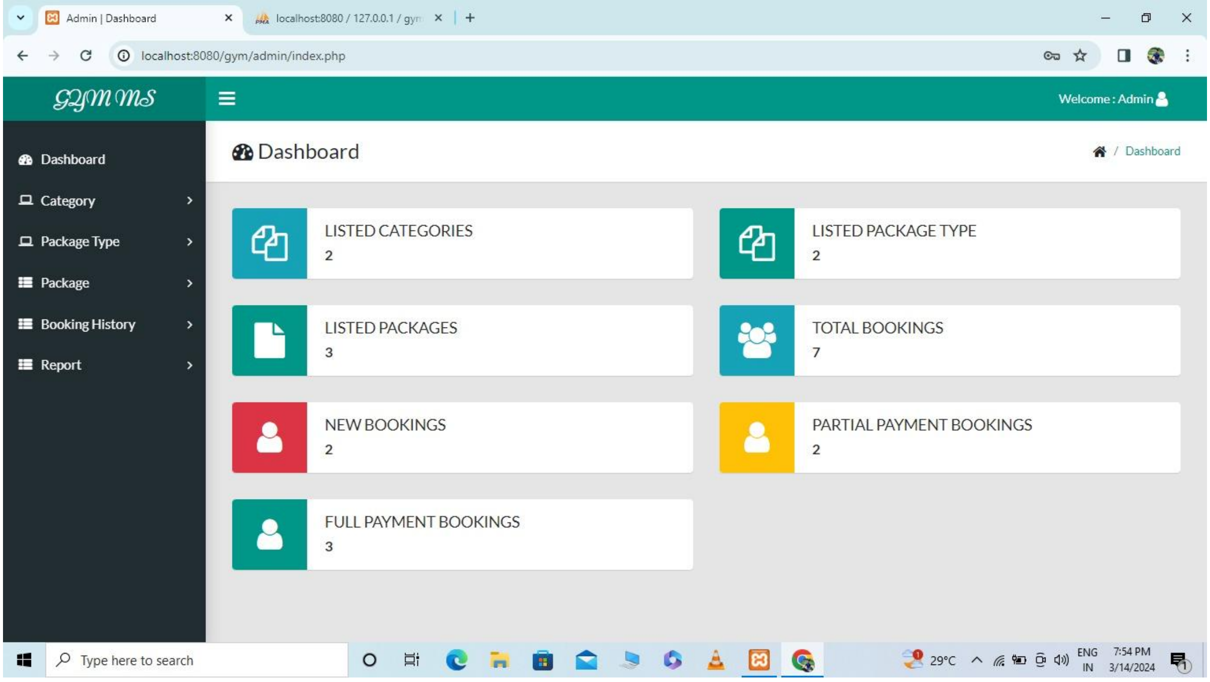
Emai

Email

PASSWORD

Password

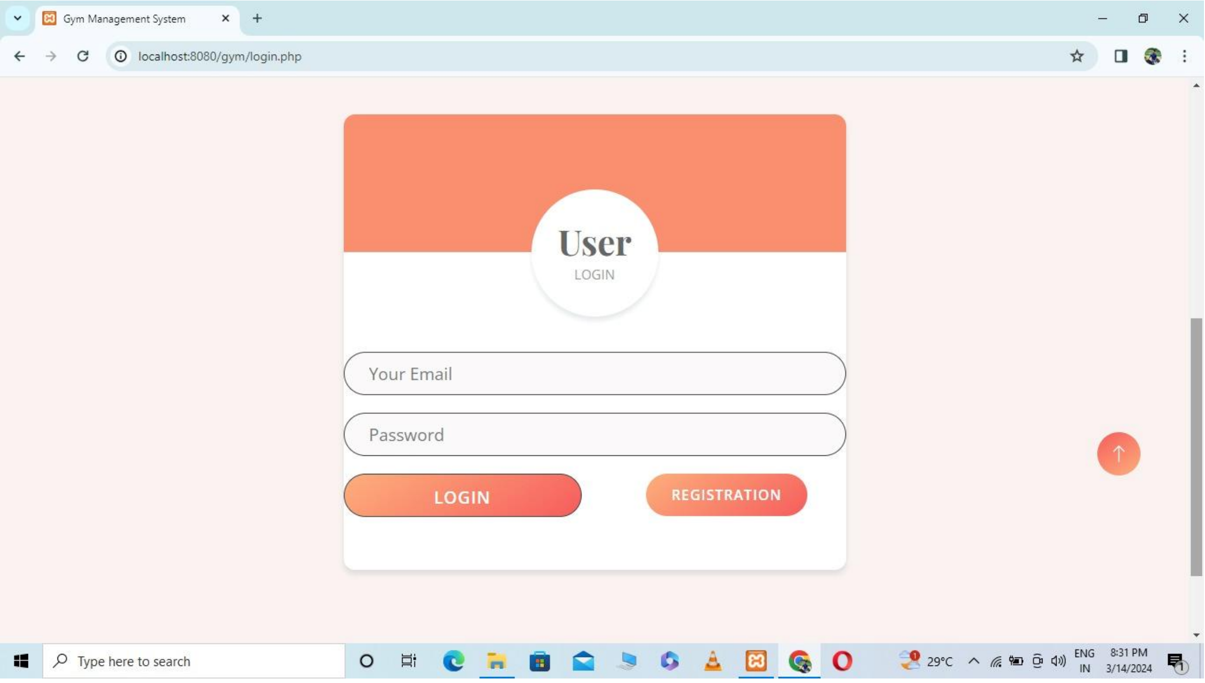
ADMIN LOGIN



Dashboard

LISTED CATEGORIES	2
LISTED PACKAGE TYPE	2
LISTED PACKAGES	3
TOTAL BOOKINGS	7
NEW BOOKINGS	2
PARTIAL PAYMENT BOOKINGS	2
FULL PAYMENT BOOKINGS	3

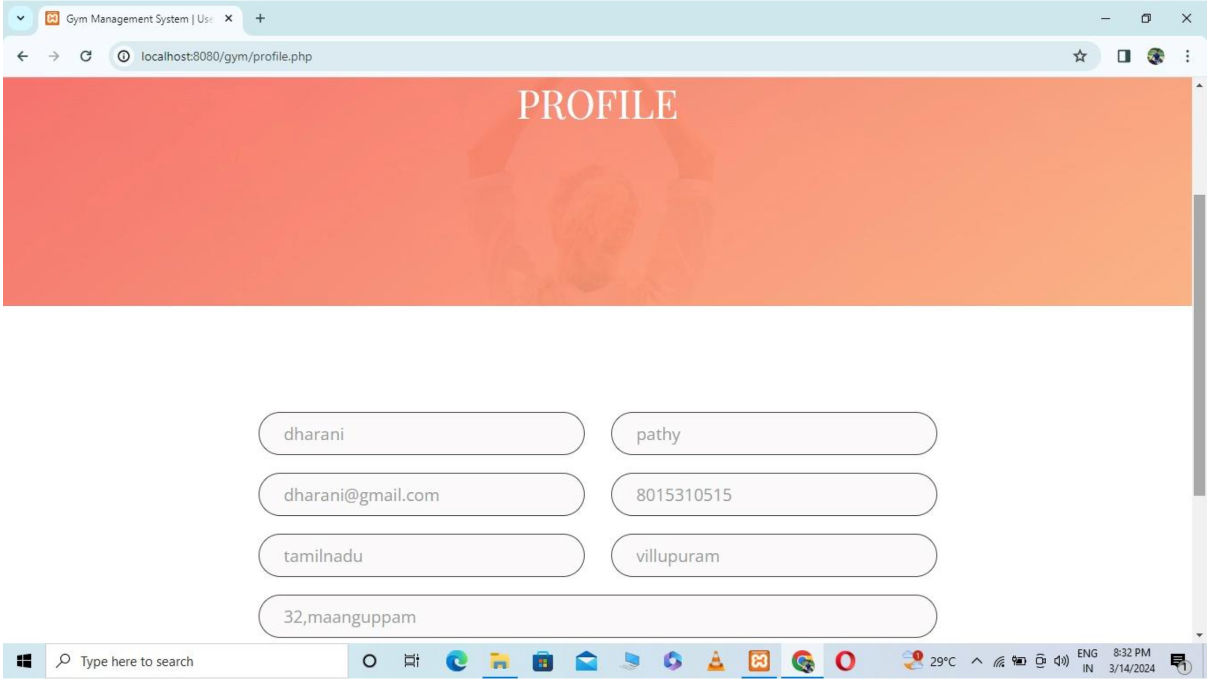
ADMIN DASHBOARD



Your Email

Password

USER LOGIN



dharani

pathy

dharani@gmail.com

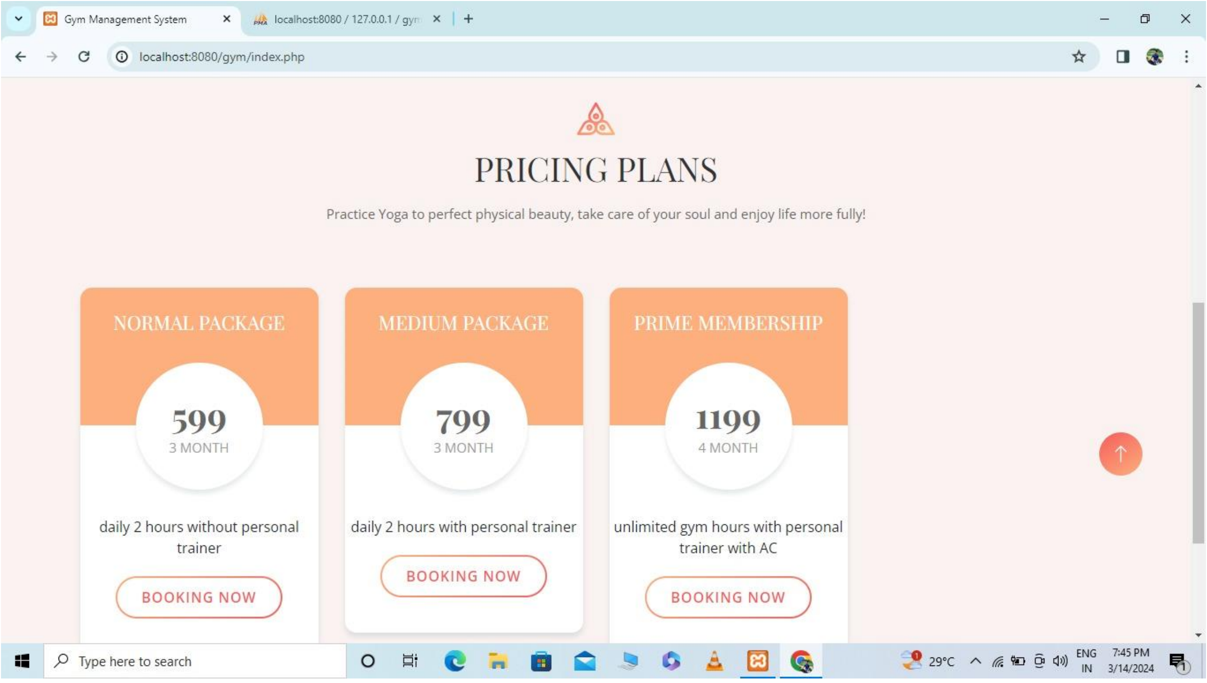
8015310515

tamilnadu

villupuram

32,maanguppam

USER PROFILE



PRICING PLANS

Practice Yoga to perfect physical beauty, take care of your soul and enjoy life more fully!

NORMAL PACKAGE

599
3 MONTH

daily 2 hours without personal trainer

BOOKING NOW

MEDIUM PACKAGE

799
3 MONTH

daily 2 hours with personal trainer

BOOKING NOW

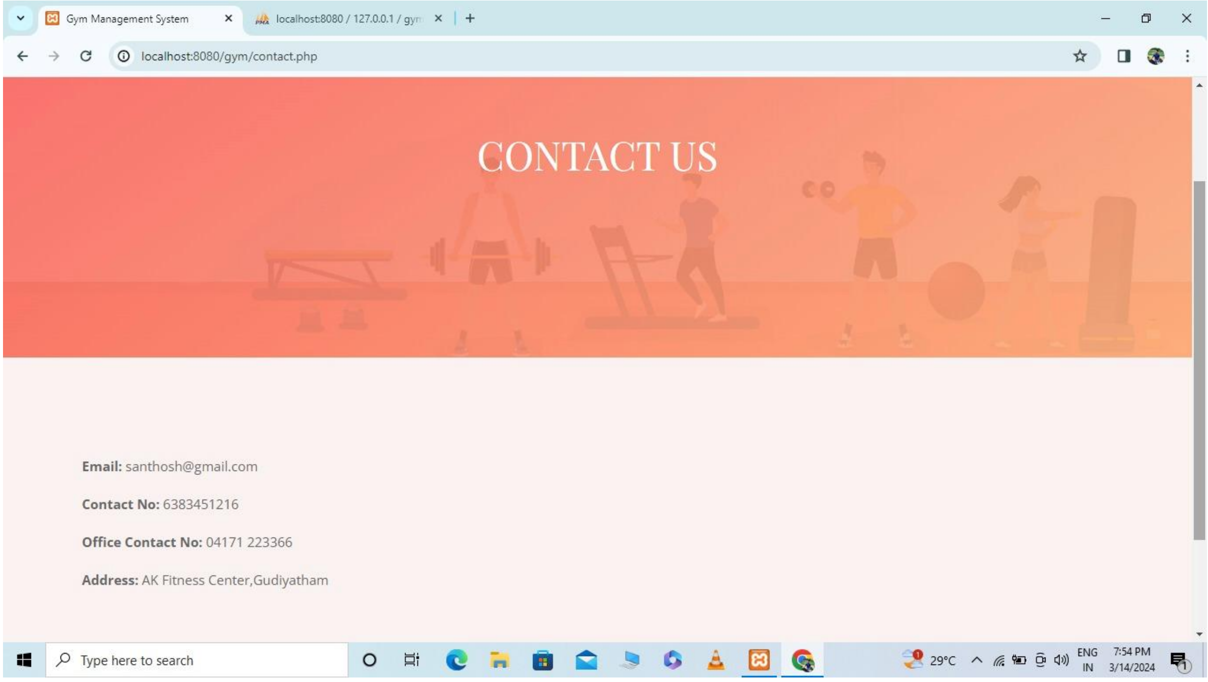
PRIME MEMBERSHIP

1199
4 MONTH

unlimited gym hours with personal trainer with AC

BOOKING NOW

PRICING PLANS



CONTACT US

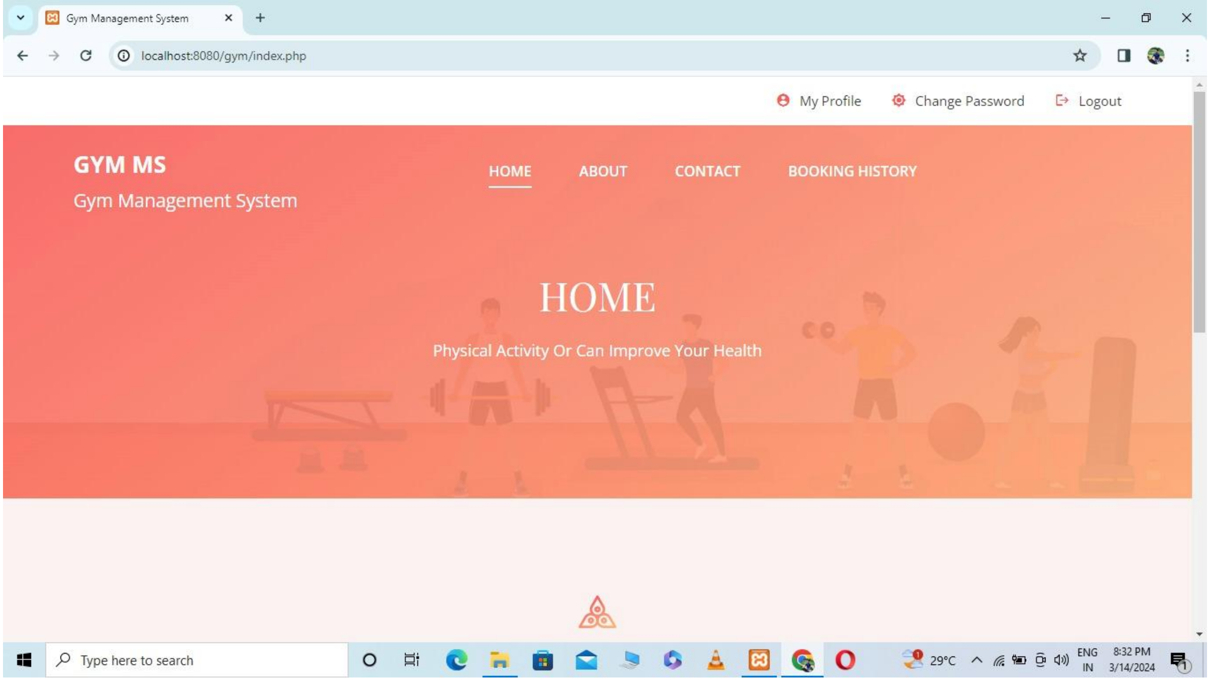
Email: santhosh@gmail.com

Contact No: 6383451216

Office Contact No: 04171 223366

Address: AK Fitness Center,Gudiyatham

CONTACT US



Physical Activity Or Can Improve Your Health

HOME PAGE

CHAPTER 7

BIBLIOGRAPHY

BIBLIOGRAPHY:

Books:

"PHP and MySQL Web Development" by Luke Welling and Laura Thomson

"Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5" by Robin Nixon

"Database Systems: Design, Implementation, and Management" by Carlos Coronel, Steven Morris, and Peter Rob

Online Tutorials:

W3Schools (https://www.w3schools.com/): Provides comprehensive tutorials on HTML, CSS, PHP, SQL, and other web development technologies.

Codecademy (https://www.codecademy.com/): Offers interactive courses on PHP, HTML, SQL, and other programming languages.

PHP.net (https://www.php.net/): Official PHP documentation with guides, tutorials, and references.