NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte (DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

Department of Computer Science & Engineering

Report on Mini Project

Gym Management System

Course Code: CS2102-1
Course Name: Database Management

System Semester: IV SEM Section: A

Submitted To:

Ms. Jayapadmini Kanchan
Assistant Professor Gd-II
Department of Computer Science
and Engineering

Submitted By:

Adib Haris Neelambra (NNM22CS009)
Mohammed Saad Shariff (NNM22CS101)

Date of submission:
18/04/2024

Signature of Course Instructor

s

ABSTRACT

The Gym Management System Project is a comprehensive application developed using PHP language with MySQL database integration. This project efficiently manages various aspects of a gym, offering functionalities such as member registration, attendance tracking, and workout scheduling, and billing management. Users can register online, view available classes, and schedule sessions based on their preferences. The system tracks member attendance, enabling administrators to monitor participation levels and optimize resources accordingly. Billing management features facilitate seamless payment processing and membership renewals. Additionally, the system generates reports on attendance, revenue, and member demographics, aiding in strategic decision-making and business analysis. With a user-friendly interface and robust functionality, this project enhances the efficiency and organization of gym operations while providing a seamless experience for both administrators and members. Its integration of PHP and MySQL ensures reliability, scalability, and security, making it a valuable tool for gym owners and managers seeking to streamline their operations and enhance member satisfaction

ACKNOWLEDGMENT

I extend my heartfelt appreciation to each individual who contributed to the successful culmination of our **Database Management System project**, specifically focused on the **Gym Management System**. The collaborative effort and unwavering dedication of every participant were instrumental in bringing this project to fruition.

I would like to extend a special note of gratitude to our esteemed principal, **Dr. Niranjan N. Chiplunkar**, for graciously granting us the opportunity to undertake this project within the premises of our college and for furnishing us with all the necessary resources and facilities essential for its execution.

Furthermore, I express profound gratitude **to Dr. Jyothi Shetty,** the Head of the Computer Science and Engineering Department at **NMAM Institute of Technology, Nitte. Dr. Shetty's** guidance and support were invaluable throughout the duration of our project, and her leadership provided us with a strong foundation upon which to build our efforts.

A debt of gratitude is also owed to **Ms. Jayapadmini Kanchan**, Assistant Professor GD II in the Department of Computer Science and Engineering. Her mentorship, encouragement, and constructive feedback were indispensable in navigating the complexities of our project, inspiring us to strive for excellence at every juncture.

Lastly, I extend my thanks to all individuals who supported us in various capacities, whether directly or indirectly, throughout the course of this project. It is through the collective support and encouragement of our peers, mentors, well-wishers that we were able to achieve such success.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

The Gym Management System Project in PHP streamlines gym operations by efficiently managing customer records and service activities. It features online registration with admin approval, ensuring secure membership access. Customers can explore service packages, manage their activities, and receive gym announcements. The system includes intuitive admin, customer, and staff panels for smooth operation management. Customers can track their progress and view personalized reports, while admins have visibility into all activities for proactive support. With its centralized platform and user-friendly interface, this project enhances communication, engagement, and overall gym administration.

1.1 Purpose

The Gym Management System serves as a comprehensive tool to streamline the operations of fitness facilities, aiming to optimize efficiency and enhance customer satisfaction. It automates administrative tasks like registration, scheduling, and payment processing, saving time and resources. With detailed customer records, including membership status and activity history, the system offers personalized services and support. It provides a centralized platform for customers to explore and select service packages tailored to their needs, improving their fitness journey. Financial tracking features ensure the financial health of the gym, facilitating informed decision-making. Through announcements, notifications, and personalized reports, the system fosters communication and engagement between management, staff, and customers.

1.2 Scope

The scope of the Gym Management System encompasses the automation and optimization of various administrative tasks within fitness facilities. It includes features for customer registration, membership management, service package selection, scheduling, payment processing, and financial tracking. Additionally, the system facilitates communication and engagement through announcements, notifications, and personalized reports. Security measures ensure data protection and membership control. The system is designed to cater to the needs of gym owners, staff, and customers by providing a user-friendly interface and centralized platform for efficient management and enhanced fitness experiences.

1.3 Overview

The Gym Management System is a comprehensive solution designed to streamline the operations of fitness facilities. It automates administrative tasks such as customer registration, membership management, scheduling, and payment processing. With a user-friendly interface, it offers customers the ability to explore and select service packages tailored to their needs. The system enhances communication and engagement through announcements, notifications, and personalized reports, fostering a positive customer experience. Security measures ensure data protection and membership control. Overall, the Gym Management System provides gym owners, staff, and customers with a centralized platform for efficient management, financial tracking, and enhanced fitness experiences

Chapter 2

Requirements Specification

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Database: MySQL 5.5
- Markup Language: HTML5
- Scripting Language: PHP 7.0.1
- IDE: Visual Studio Code
- Server: Apache
- Browser: Google Chrome, Microsoft Edge , Firefox

Chapter 3

System Design

3.1 ER Diagram

For the project we are taking 6 strong entities Admin, Reminder, Attendance, Rates, Members and Equipment.

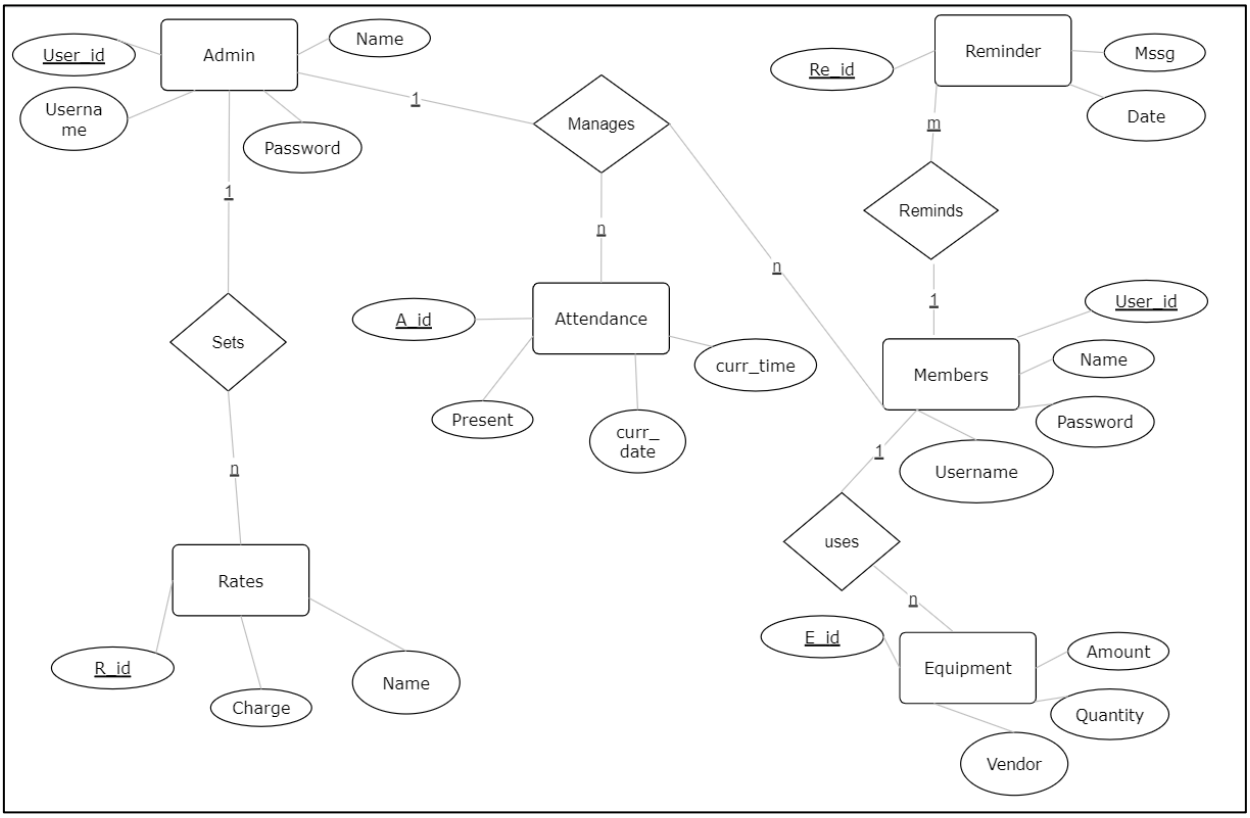


Fig 3.1: ER Diagram

3.2 Mapping From ER diagram to Schema

To convert ER to schema we follow 7 steps which are as follows:

1. **Mapping of Regular Entities**: This step involves mapping all the regular strong entities types to tabular format by identifying their primary keys.
2. **Mapping of 1:1 Relation:** In this step foreign keys are assigned using foreign key approach. The primary key of the participating relations are added as primary key to second entity types by looking at the participating constraints.
3. **Mapping of 1:N Relation:** Foreign key approach is used to add one sided primary key to the n sided entity at foreign key.
4. **Mapping of M:N Relation:** Here we use the cross-reference approach where the relationship is converted to a new relation within attributes on primary keys of both participating relations.
5. **Mapping of Weak Entity:** When mapping weak entity types along with other attributes the partial key and primary key of parent entity together will form their primary key of the new relation.
6. **Mapping of N-ary Relation:** For mapping N array relationship we create a new relation with a relationship name in its attribute and primary keys of all participating entity types.
7. **Mapping of Multivalued Relation:** For multivalued attributes a separate relation has to be created along with primary key of parent relation.

To get schema for database we will follow these steps:

1. **Mapping of Regular Entities**: Initially, we will identify all the strong entities (the entities which have primary key in them). In our database these are the entities with the attributes

   Admin (User_id, name , username , password)

   Reminder (Re_id, Message ,Date)

   Attendance (A_id , Present , curr_date ,curr_time)

   Members (User_id , Name , Password , Username)

Rates (R_id , Charge , Name)

Equipment (E_id, Amount, Quantity, Vendor

2. **Mapping of 1:1 Relation**: None of the entities are participating in the 1:1 relation type. In it each record in 1 table corresponds uniquely to a record in another table.
3. **Mapping of 1:N Relation**: In our database all the entities are participating in 1:n. In a one-to-many relationship, the "n" side entity includes a foreign key referencing the primary key of the "one" side entity.

   The entities and attributes which are in 1:n are :

   Reminder (Re_id, Message ,Date)

   Attendance (A_id , Present , curr_date ,curr_time)
   Members (User_id , Name , Password , Username)
   Rates (R_id , Charge , Name)
   Equipment (E_id, Amount, Quantity, Vendor)

4. **Mapping of M:N Relation**: None of the entities are participating in m:n relation. In a many-to-many relationship, a separate associative entity is created to link the participating entities.
5. **Mapping of Weak Entities**: We will identify all the weak entities (the entities which don't have primary key in them). In our database these are no entities with such attributes.
6. **Mapping of N-ary Relation:** None of the entities are participating in this relation. In it the relation is linked to and linked from same entity.
7. **Mapping of Multivalued relation**: A multivalued attribute allows an entity to have multiple values for a single attribute. This is typically represented as a separate table with a foreign key referencing the primary key of the original entity.

3.3 Assumptions

1. Assumption of Database Usage: The assumption that the MySQL database is used for storing data related to the project.

2. Assumption of Table Structure: The assumption that the database consists of several tables representing different entities such as admin, members, attendance, equipment, rates, and reminder.

3. Assumption of Primary Keys: The assumption that each table has a primary key column to uniquely identify each record within that table.

4. Assumption of Indexes: The assumption that indexes are created on certain columns (e.g., email, id) to improve query performance.

5. Assumption of Auto-increment: The assumption that certain primary key columns are set to auto-increment to automatically generate unique values for new records.

3.4 Schema Diagram

A Schema is a pictorial representation of the relationship between the tables in the database that is created. The term "schema" refers to the representation of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language .The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

Chapter 4

Implementation

4.1 Pseudocodes used

Pseudocode to connect sql and php

In order to store or access the data inside a MySQL database, we first need to connect to the MySQL database server. In PHP we can do this using the mysqli connect() function. All communication between PHP and the MySQL database server takes place through this connection. The hostname parameter in the above syntax specify the host name, whereas the username and password parameters specifies the credentials to access MySQL server, and the database parameter, if provided will specify the default MySQL database to be used when performing queries. The default username for MySQL database server is root and there is no password and hostname is localhost.

C: > xampp > htdocs > Gym-System > dbcon.php

```php
<?php
$con = mysqli_connect("localhost","root","","gymnsb");

// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
?><!-- Visit codeastro.com for more projects -->
```

Fig 4.1 Pseudocode to connect sql

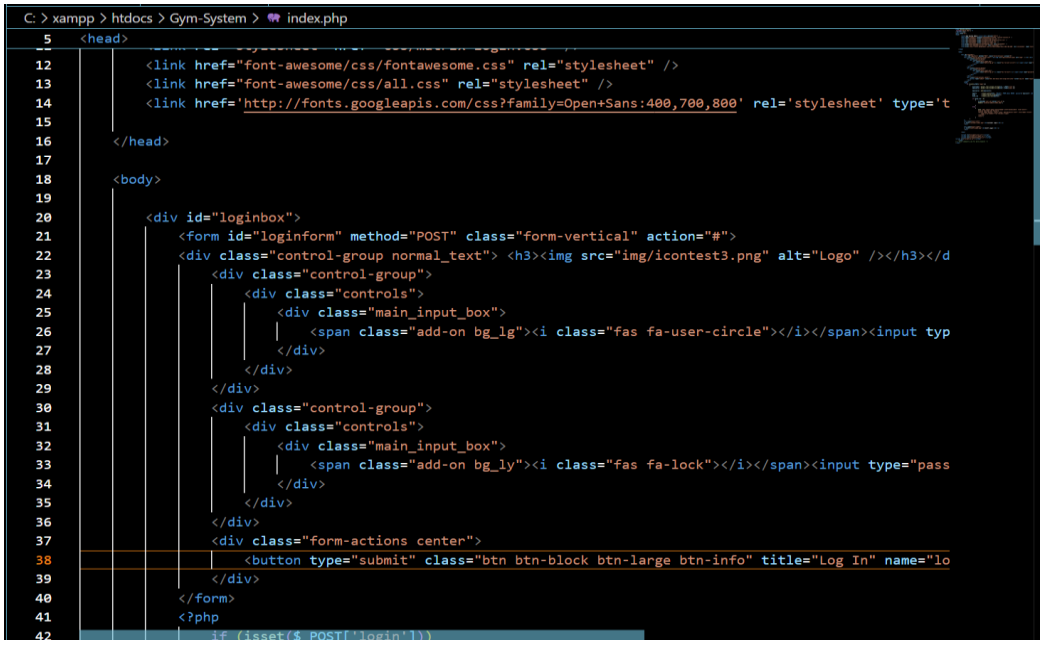
Pseudocode for INSERT

Insert statement is a DML (Data modification language) statement which is used to insert data in the MySQL table. PHP$ POST is a PHP super global variable which is used to collect the form data from the user.

The Sign In, Sign Up Pages

```
<head>
    <body>
        <div id="loginbox">
            <div class="control-group normal_text"> <h3><img src="img/icontest3.png" alt="Logo" /></h3></d
                </div>
            </form>
            <?php
                if (isset($_POST['login']))
                {
                    $username = mysqli_real_escape_string($con, $_POST['user']);
                    $password = mysqli_real_escape_string($con, $_POST['pass']);

                    $password = md5($password);

                    $query      = mysqli_query($con, "SELECT * FROM admin WHERE  password='$password'
                    $row        = mysqli_fetch_array($query);
                    $num_row    = mysqli_num_rows($query);

                    if ($num_row > 0)
                        {
                            $_SESSION['user_id']=$row['user_id'];
                            header('location:admin/index.php');

                        }
                    else
                        {
                            echo "<div class='alert alert-danger alert-dismissible' role='alert'>
                            Invalid Username and Password
                            <button type='button' class='close' data-dismiss='alert' aria-label='Close
                                <span aria-hidden='true'>&times;</span>
                            </button>
```

Fig 4.2: Sign-up code



```
C: > xampp > htdocs > Gym-System > index.php
<head>
    <link href="font-awesome/css/fontawesome.css" rel="stylesheet" />
    <link href="font-awesome/css/all.css" rel="stylesheet" />
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,700,800' rel='stylesheet' type='t

</head>

<body>

    <div id="loginbox">
        <form id="loginform" method="POST" class="form-vertical" action="#">
        <div class="control-group normal_text"> <h3><img src="img/icontest3.png" alt="Logo" /></h3></d
            <div class="control-group">
                <div class="controls">
                    <div class="main_input_box">
                        <span class="add-on bg_lg"><i class="fas fa-user-circle"></i></span><input typ
                    </div>
                </div>
            </div>
            <div class="control-group">
                <div class="controls">
                    <div class="main_input_box">
                        <span class="add-on bg_ly"><i class="fas fa-lock"></i></span><input type="pass
                    </div>
                </div>
            </div>
            <div class="form-actions center">
                <button type="submit" class="btn btn-block btn-large btn-info" title="Log In" name="lo
            </div>
        </form>
        <?php
```

Fig 4.3 Sign-in code

Pseudocode for UPDATE

The UPDATE statement is used to modify the existing records in a table. The WHERE clause specifies which record(s) that should be updated.

```
<?php
include 'dbcon.php';
$id=$_GET['id'];
$qry= "select * from members where user_id='$id'";
$result=mysqli_query($conn,$qry);
while($row=mysqli_fetch_array($result)){
?>

<div id="content">
  <div id="content-header">
    <div id="breadcrumb"> <a href="index.php" title="Go to Home" class="tip-bottom"><i class="fas fa-home"
    <h1 class="text-center">Update Customer's Progress <i class="fas fa-signal"></i></h1>
  </div>


  <div class="container-fluid" style="margin-top:-38px;">
    <div class="row-fluid">
      <div class="span12">
        <div class="widget-box">
          <div class="widget-title"> <span class="icon"> <i class="fas fa-signal"></i> </span>
            <h5>Progress </h5>
          </div>
          <div class="widget-content">
            <div class="row-fluid">

          <div class="span2"></div>

                <div class="span8">
                  <table class="table table-bordered table-invoice">

                    <tbody>
          <form action="userprogress-req.php" method="POST">
```

Fig 4.4 To Update values in table

4.2 Tables used

The 5 tables used are as following:

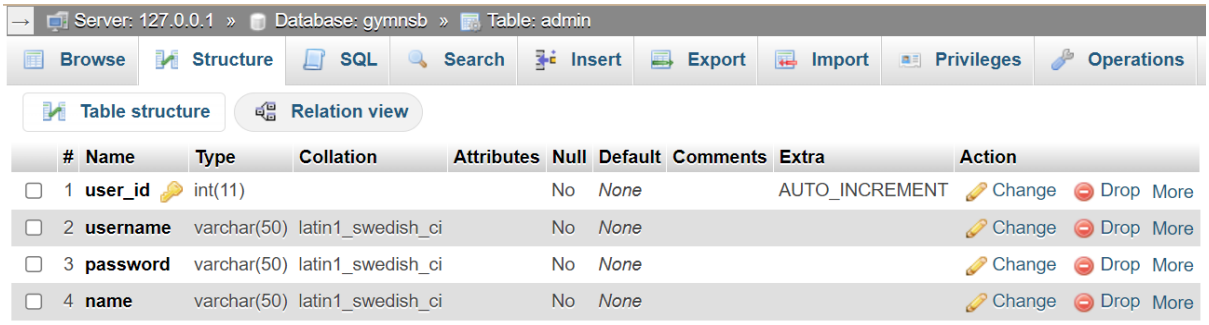
Admin (User_id, name, username , password)

Reminder (Re_id, Message, Date)

Attendance (A_id , Present , curr_date ,curr_time)

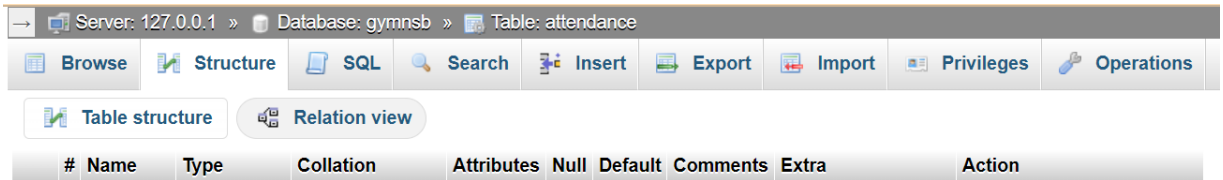
Members (User_id , Name , Password , Username)

Equipment (E_id, Amount, Quantity, Vendor)



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
3	password	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Fig 4.5 Admin Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	curr_date	text	latin1_swedish_ci		No	None			Change Drop More
4	curr_time	text	latin1_swedish_ci		No	None			Change Drop More
5	present	tinyint(4)			No	None			Change Drop More

Fig 4.6 Attendance Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	amount	int(100)			No	None			Change Drop More
4	quantity	int(100)			No	None			Change Drop More
5	vendor	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
6	description	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
7	address	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
8	contact	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
9	date	date			No	None			Change Drop More

Fig 4.7 Equipment Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**user_id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**fullname**	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
3	**username**	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
4	**password**	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
5	**gender**	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
6	**dor**	date			No	*None*			Change Drop More
7	**services**	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More

Fig 4.8 Members Table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	**name**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
3	**charge**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More

Fig 4.9 Rates Table

Chapter 5

Results and Discussion

Sign-Up Page:

Through this page user, admin and staff can sign up to the gym management system

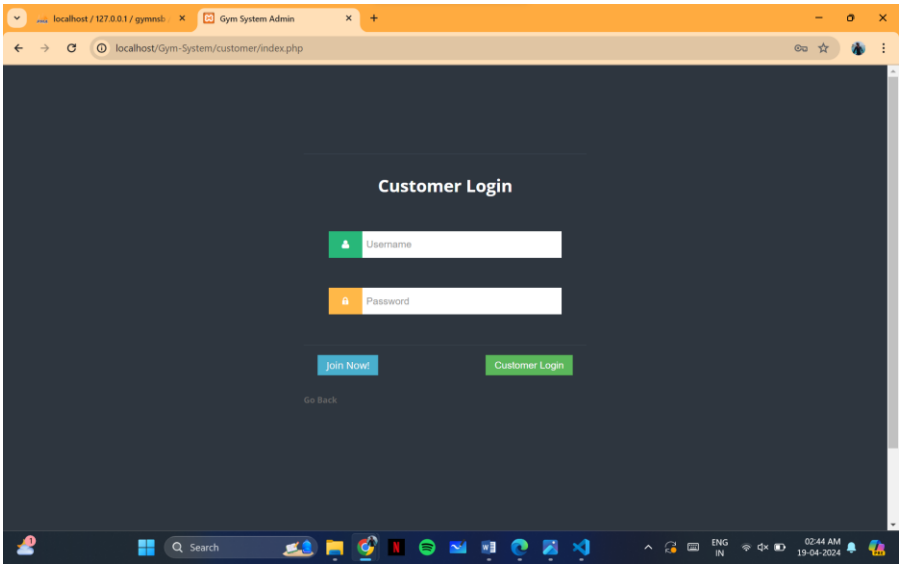


Fig 5.1: Home Page

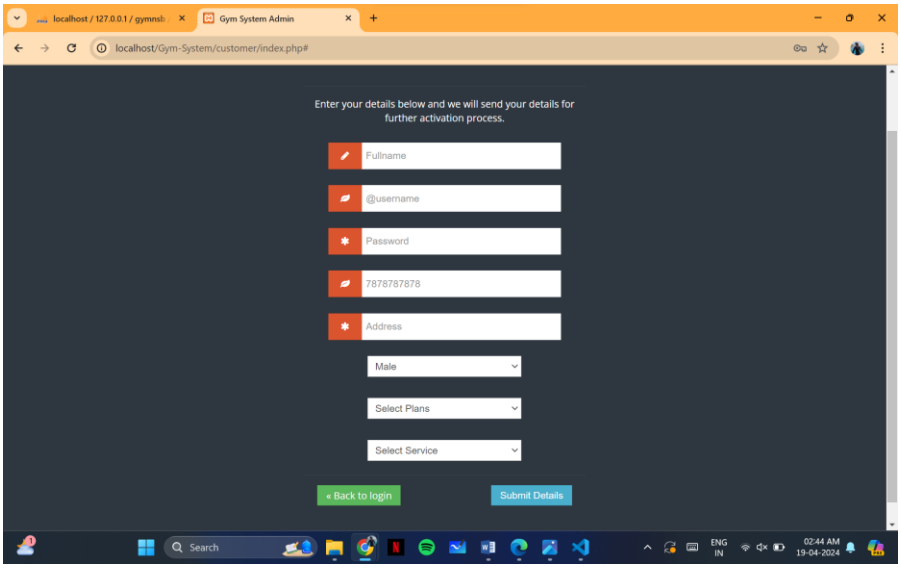


Fig 5.2: Customer Sign-up Page

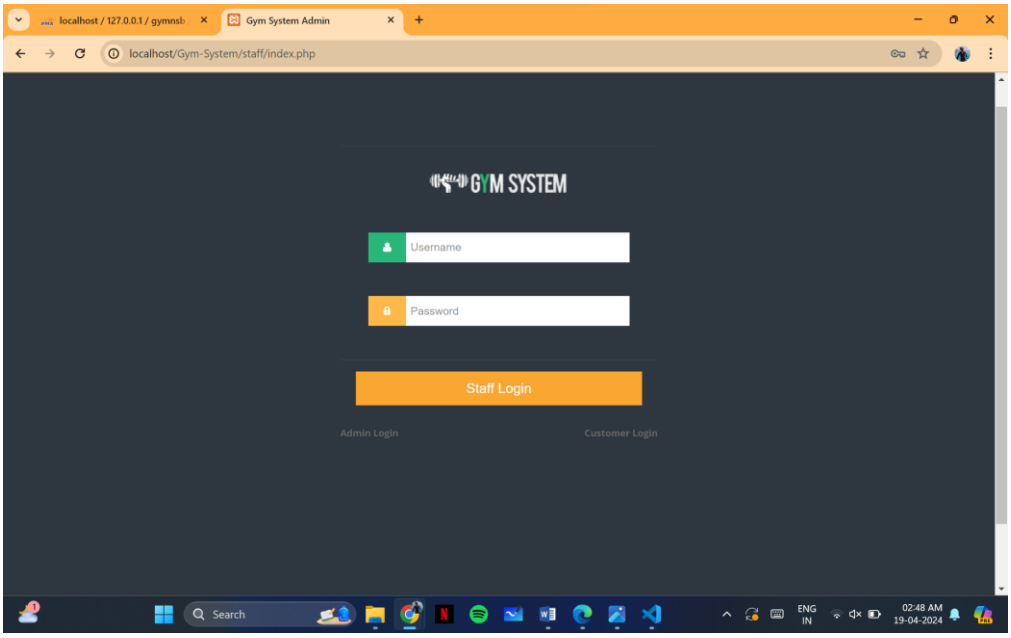


Fig 5.3: Staff Sign-Up Page

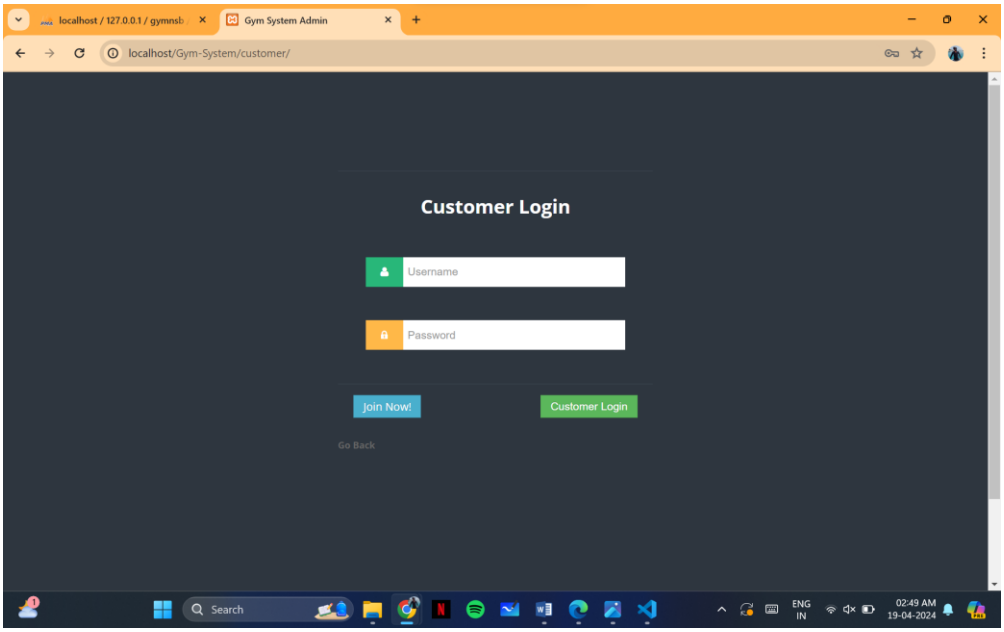


Fig 5.4: Customer Sign-Up Page

Sign-In Page:

Through this page user, admin and volunteer can sign in to the food donation website if they already have an account in it.

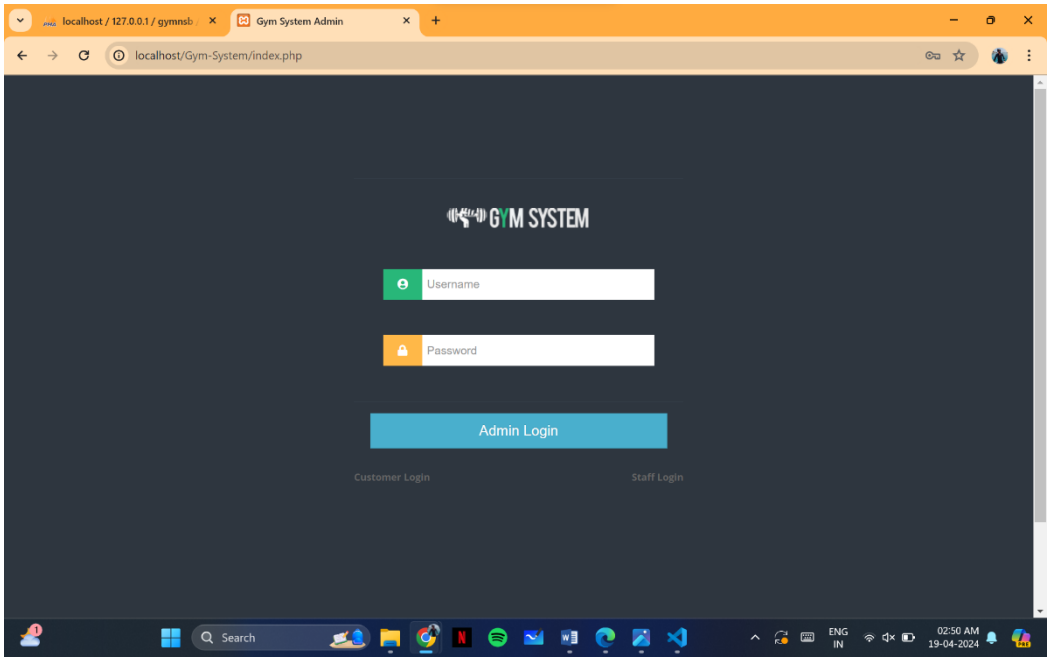


Fig: 5.5 Admin Sign-In Page

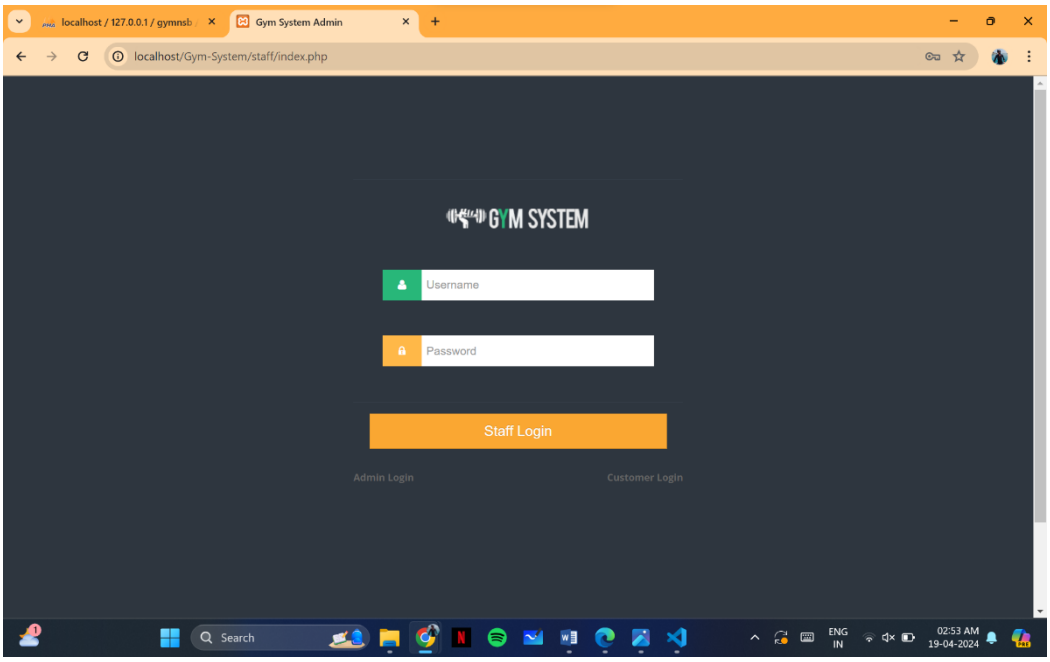


Fig: 5.6 Staff Sign-In Page

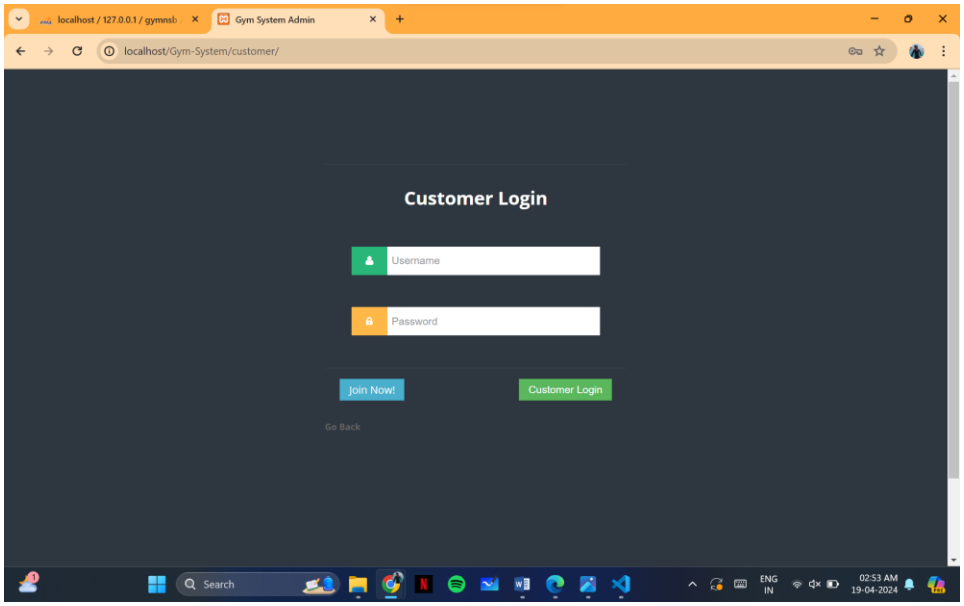


Fig 5.7: Customer Sign-In Page

User Homepage:

In it, admin can manage various things, he or she can give feedback to the admin and has catalogue of all the members and equipment in the gym.

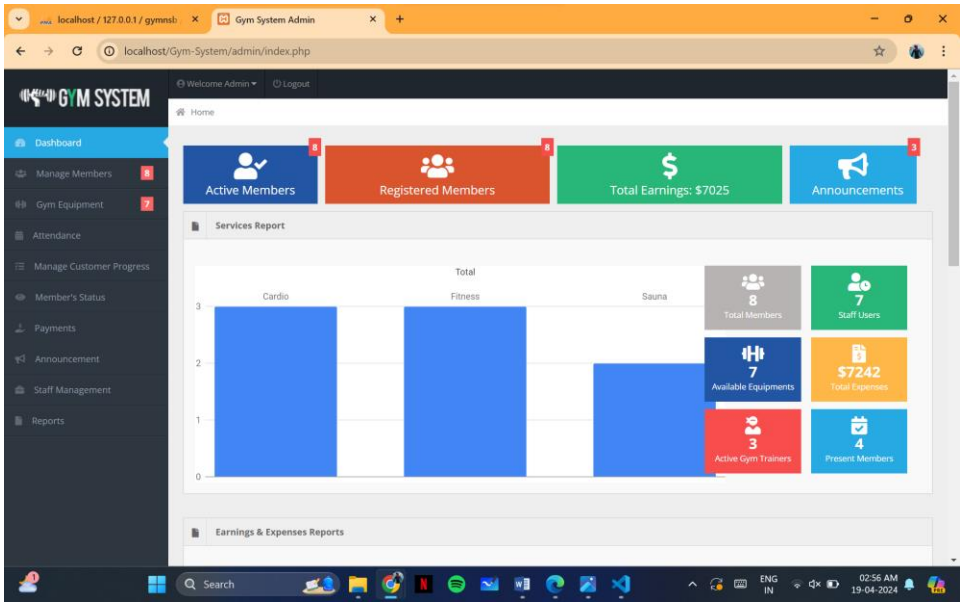


Fig 5.8: Admin Home Page

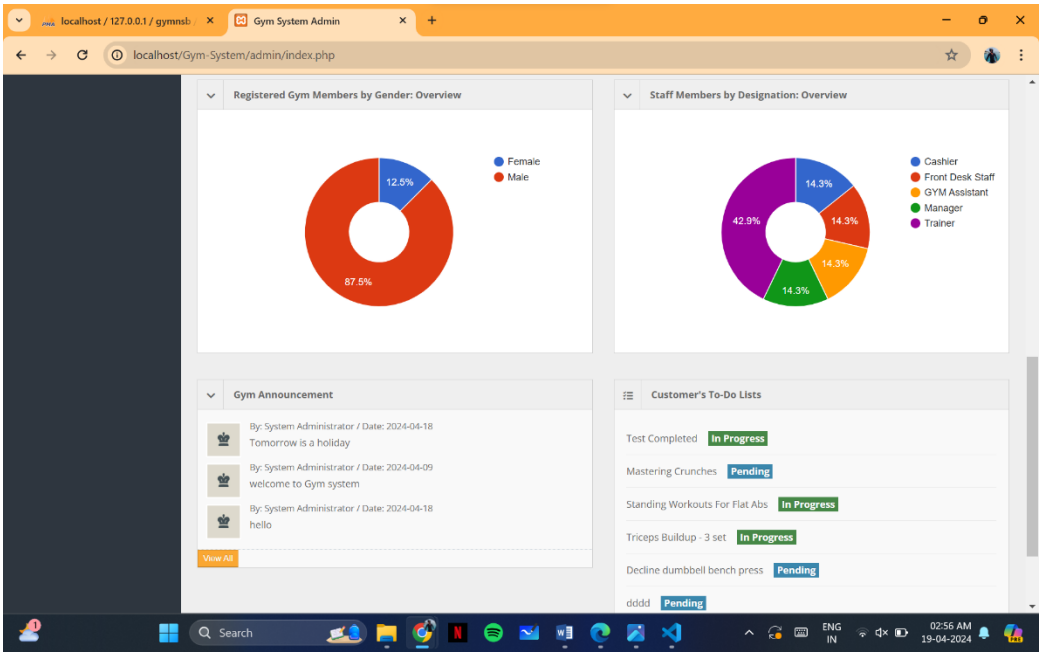


Fig 5.9: Graphs

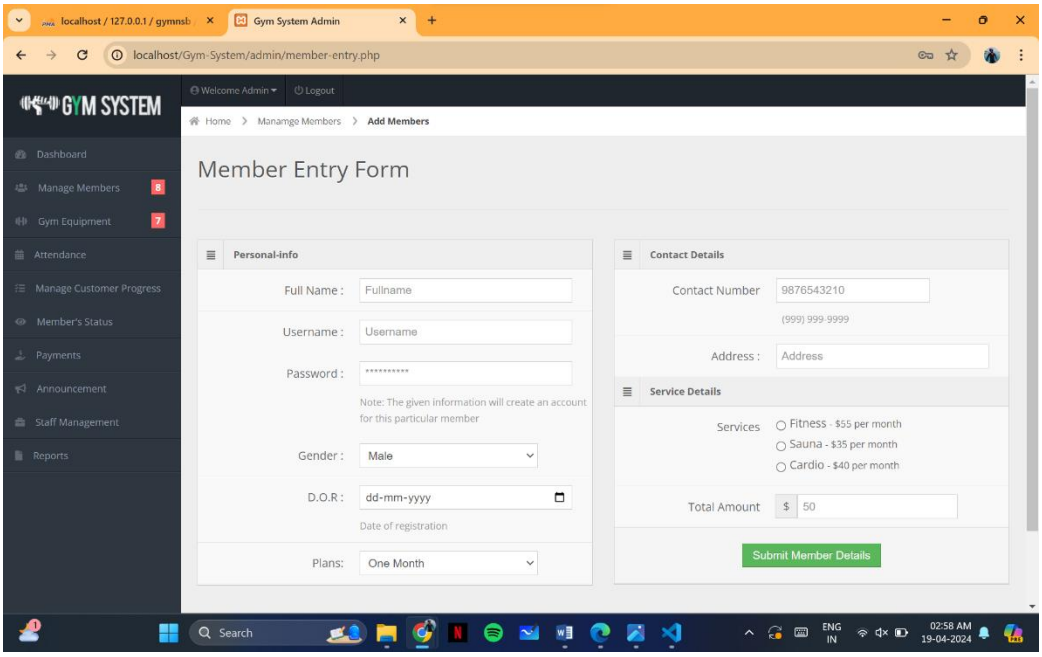


Fig 5.10: Member Entry Form

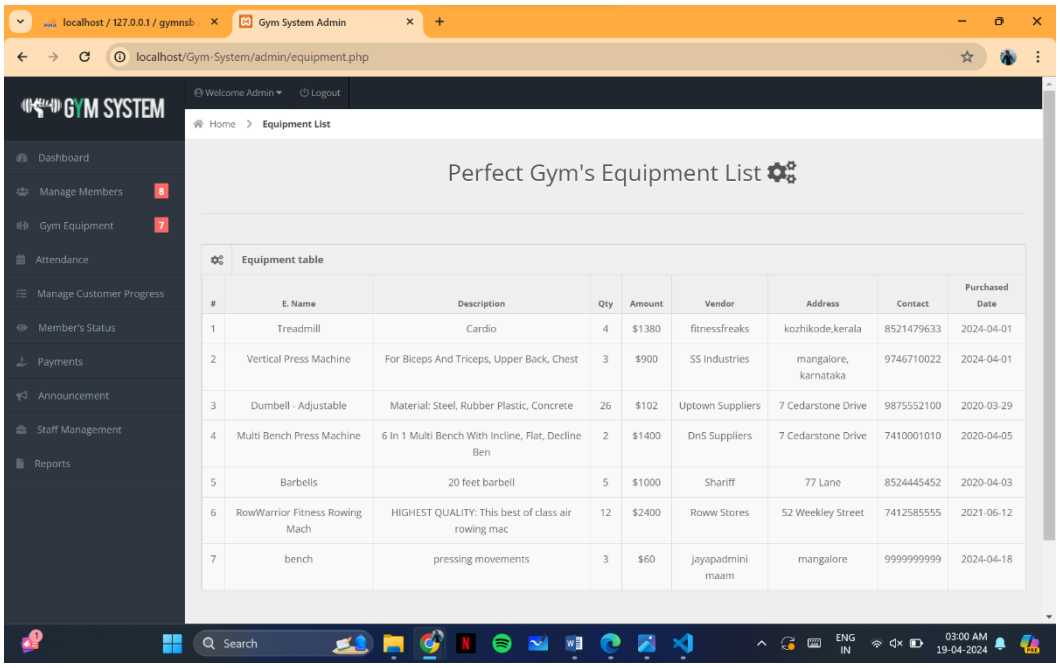


Fig 5.11: Equipment List

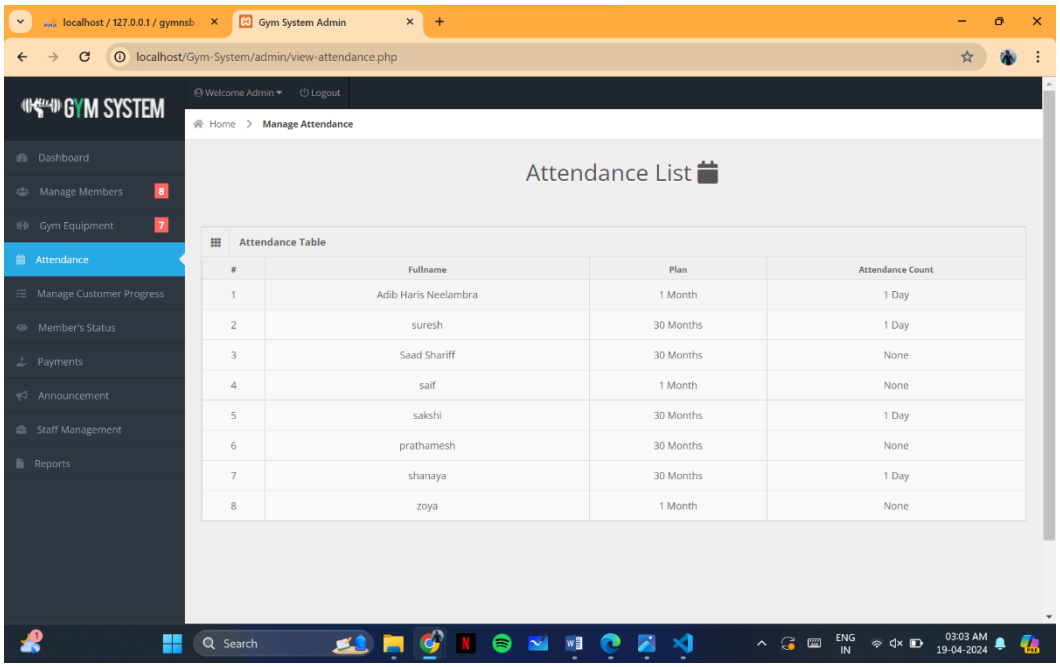


Fig 5.12: Attendance List

Chapter 6

Conclusion and Future work

The gym management system, as depicted by the ERD, offers a comprehensive solution for efficiently managing gym operations. By structuring data into separate entities and establishing relationships between them, the system facilitates user management, announcements, attendance tracking, equipment management, membership rates, reminders, and to-do lists. Through effective data organization and relationship establishment, the system aims to streamline administrative tasks, improve data integrity, and enhance overall operational efficiency within the gym environment.

To further enhance the system, future development efforts could focus on three key areas. Firstly, User Interface Development is crucial to designing user-friendly interfaces tailored for administrators, staff, and members. Secondly, investing in Enhanced Reporting functionalities would enable the generation of insightful reports on various aspects of gym operations. Thirdly, integrating Automation and Notifications could automate routine tasks and facilitate timely communication with stakeholders. Additionally, Payment Integration capabilities, Mobile Application Development, and other enhancements could be explored to extend system functionality and improve user experience.

.

References

[1] Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, Pearson.

[2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

[3] Silberschatz Korth and Sudharshan: Database System Concepts, 6th Edition, McGraw Hill, 2013.

[4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

[5] https://www.w3schools.com/

[6] https://www.wikipedia.org/

[7] https://images.google.com/

[8] https://github.com/topics/gym-management-system/

[9] https://youtu.be/y7H_nkdtbz4?si=cwMl0SQJ-4cHn2BY

[10] https://codeastro.com/gym-management-system-in-php-mysql/