

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, BELAGAVI – 590018



A MINI PROJECT REPORT

ON

“GYM MANAGEMENT SYSTEM”

Submitted in partial fulfillment of requirements for the *course*
DBMS Laboratory with Mini Project [18CSL58] of *Fifth Semester*
of *Bachelor of Engineering in Computer Science & Engineering*
during the academic year 2021-22.

Submitted By

“Shaz Ahammed”

“4MH19CS092”

“Roopali Singh R L”

“4MH19CS081”

Under the Guidance of

“Prof. Pratap M S”

Assistant Professor,
Dept. of CS&E,



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

2021 - 2022

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the mini project work entitled “**GYM MANAGEMENT SYSTEM**” is a bonafide work carried out by **Shaz Ahammed [4MH19CS092]** and **Roopali Singh R L [4MH19CS081]** in partial fulfillment for the **DBMS Laboratory with Mini Project (18CSL58)** prescribed by the Visvesvaraya Technological University, Belagavi during the year 2021-2022 for the fifth semester B.E in Computer Science and Engineering. The mini project report has been approved as it satisfies the academic requirements.

Signature of Guide

(Prof. Pratap M S)

Assistant Professor, Dept. of CS&E
MIT Mysore

Signature of HOD

(Dr. Shivamurthy R C)

Professor & Head, Dept. of CS&E
MIT Mysore

Name of the Examiners	Signature with date
1.....
2.....

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. B.G. Naresh Kumar, Principal, Maharaja Institute of Technology Mysore**, for having supported us in our academic endeavors.

We are extremely thankful to **Dr. Shivamurthy R C, Professor & Head, Department of Computer Science and Engineering**, for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide **Prof. Pratap M S, Assistant Professor, Department of Computer Science and Engineering**, for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members of Department of Computer Science and Engineering**, for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

Shaz Ahammed (4MH19CS092)

Roopali Singh R L (4MH19CS081)

ABSTRACT

The gym is a great place to benefit a person both physically. Gym not only makes you fit physically but also mentally. In current busy life schedule people are neglecting the most important asset of their life, that is their own health. Going to the gym on routine basis enforces you to eat clean and avoid fast food, alcohol, smoking etc. Our aim is to make it easier for the different stack holders of a gym for the institutions smooth run.

Our project, The Gym Management System helps the administration in speeding up the tasks and also reduce the complexity. The purpose or objective of this system is to digitalize and create an automated system. Our project has separate login option for the admin, trainer and the member. Our project lets a gym member track his improvement and guide him to set up a well scheduled gym session. Any member can easily login to track his performance. Our project also lets the admin create or delete any member or trainer. We believe that our project will help any gym institution for its easy and smooth run.

~~~~~ TABLE OF CONTENTS ~~~~~

1. INTRODUCTION	01
1.1 Aim of the Project	01
1.2 Overview of the Project.....	01
1.3 Outcome of the Project.....	02
1.4 Software Requirements.....	02
 2. DESIGN	 04
2.1 Schema Diagram	04
2.2 E-R Diagram	05
2.3 Use Case Diagram	05
2.4 Data Flow Diagram	06
2.5 Sequence Diagram	07
 3. IMPLEMENTATION	 09
3.1 Description of Tables	09
3.2 Constraint on Tables	10
3.3 Back End Implementations	11
3.4 Front End Implementations	14
 4. RESULT ANALYSIS	 29
4.1 Snap Shots	29
4.2 Discussion	35
4.3 Testing	36
 5. CONCLUSION AND FUTURE WORK	 38
4.1 Conclusion	38
4.2 Future Enhancement	38
 6. REFERENCES	 39

Chapter 01

INTRODUCTION

1.1 Aim of The Project

As a total man power based system is currently running for the gym procedures, designing a new system which makes the whole process online demands a deep knowledge about the existing system. Throughout the project we focus on presenting information and commands in an easy and intelligible manner. The purpose of our GYM MANAGEMENT SYSTEM is to provide a leading technological tool for the ease of gym functions such as adding and deleting trainer or member, tracking trainee performance and deciding on his daily routine and for tracking the trainee and trainer details. It will reduce considerably the difficulties faced by the existing system, with minimum error and difficulties.

1.2 Overview of The Project

Our project provides a smooth way of running a gym institute and makes things easier for the admin (owner), trainer and trainee (member). Now there can be minimal communication between the different entities as the whole process is transparent and visible. Own project allows three types of log-in, one for the admin, one for the trainer and then for the trainee. The three types of log-in gives three different sets of permission for the user, each requires different user id and password which is set by the admin. The admin can manipulate the trainee and trainer while the trainer can only update on the performance of trainees under him. The trainees can only track their performance.

1.4 Outcome of The Project

Our Gym Management System allows the trainer to easily update the weight and fat content of trainees under him/her. The trainees can now easily view their improvement with just a click. The manual work of recording these on paper and storing them is out of question. The admin can now easily decide who should train whom and who all can access the gym resources.

1.4 Software Requirement

Front End: HTML, CSS, Java Script

Back End: MySQL

Language: php

The system is implemented as 3-tier approach with a backend database handled by the system administrator and a web browser as the front-end client. This document will discuss each of the underlying technologies used to create and implement gym management website. To implement this we have used PHP, which is platform independent and therefore, can be run on all major operating systems. PHP provides support to all major servers like Apache and databases like MySQL. Since it uses its own memory, the loading time is decreased and processing time is increased. Next, we have used HTML, JavaScript and CSS for front-end implementation. They provide a front-end development framework to create fully responsive web pages and define proper styles and presentation of the document. Lastly, MySQL is used

as the back-end database since it is one of the most popular open-source databases, and it provides fast data access, easy installation and simplicity

Chapter 02

DESIGN

2.1 Schema Diagram

We have four different tables admin, trainer, trainee and trainer-trainee. The table trainer stores the details of all trainer, details such as trainer name, trainer id, phone and salary. The table trainee has details of the trainee such as trainee name, trainee id, phone and monthly fee. The table trainer-trainee has the performance details of the trainee such as the weight of the trainee, fat content of the trainee and his plan details. This table is connected with the trainee table using the trainee id.

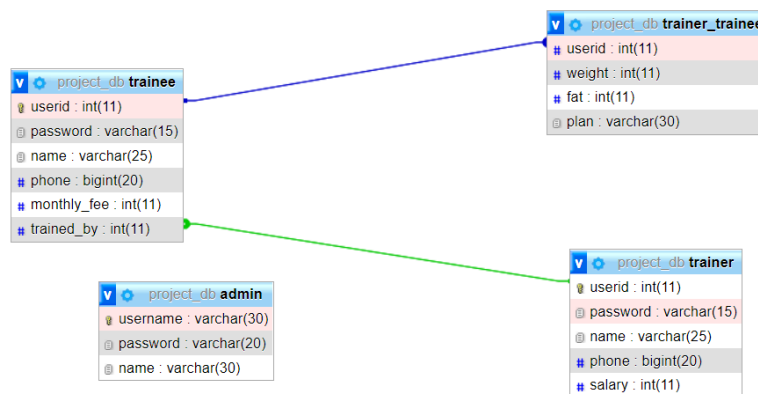


Fig 2.1: Schema Diagram

2.2 E-R Diagram

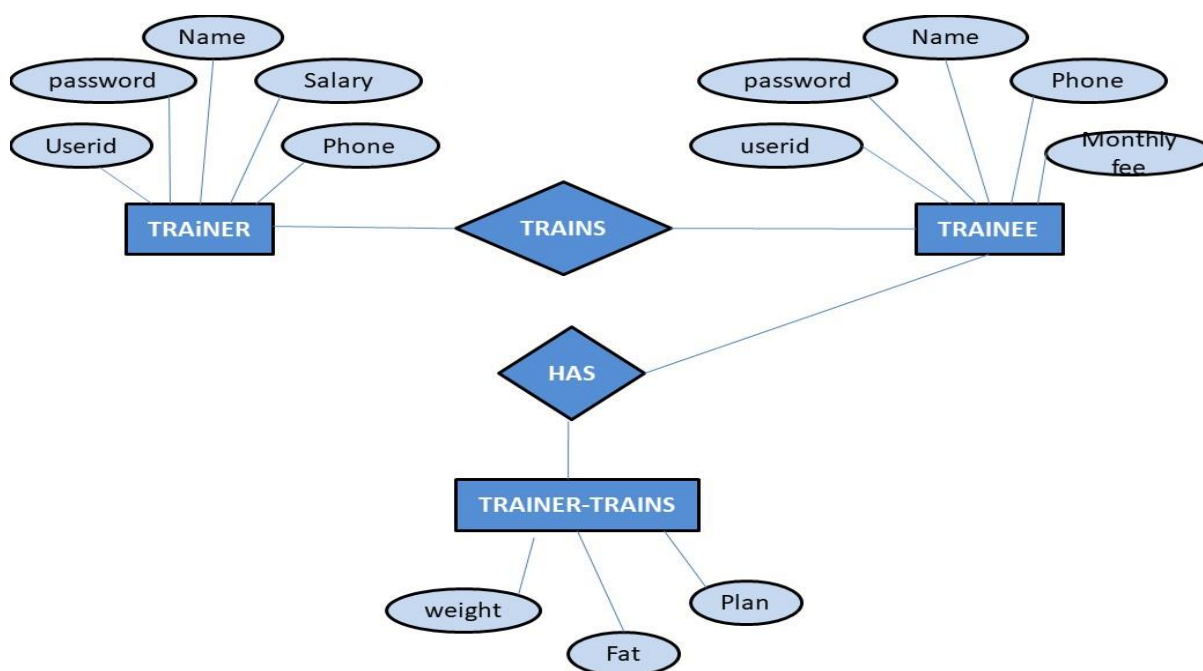


Fig 2.2: E R Diagram

In our schema diagram the table trainer is connected to the trainee table using the trainer id to mention the trainee's trainer. Also the tables trainer-trainee is an extension of the trainee table which mentions the trainee's performance details. The schema diagram further provides the detailed relationship between the relations.

2.3 Use Case Diagram

In our project Gym management system, we have three types of users – The Admin, the Trainer and the Trainee. The admin has the ability to add, view or remove the details of both trainee and trainer. The trainer has the ability to view the trainees under him and update only their weight and fat that is their performance. The trainee can view his performance details

and also view his trainer's name and phone number. The exercise that the trainee is supposed to do is also mentioned.

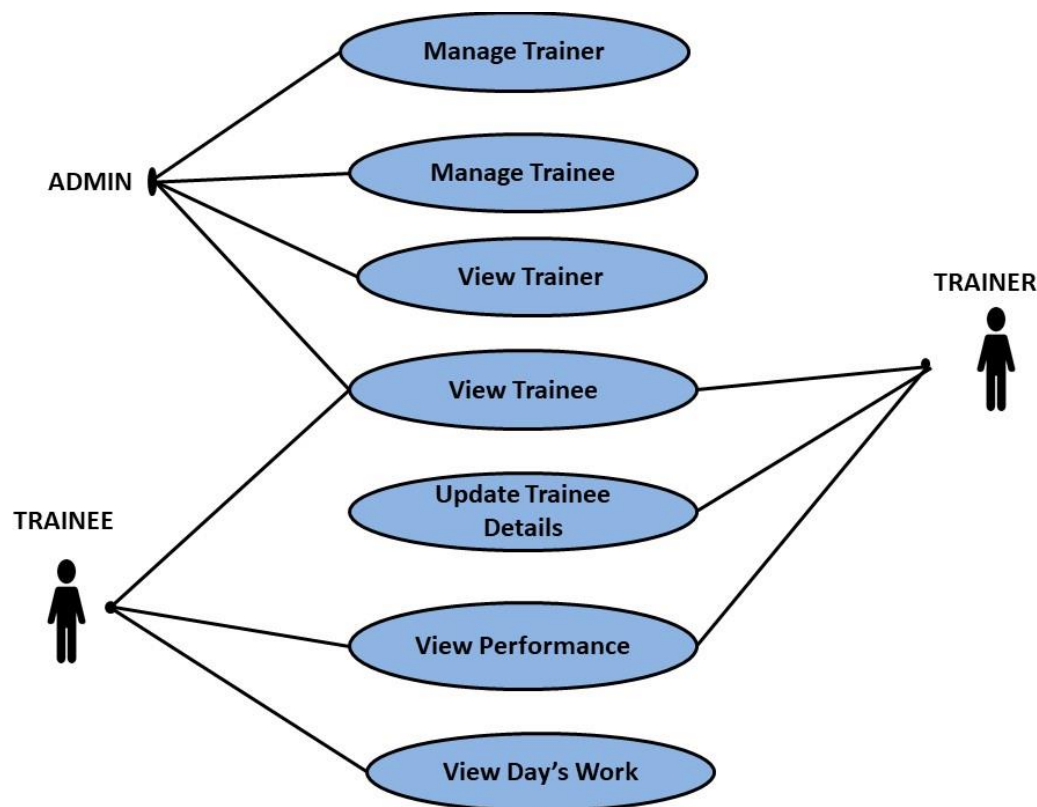


Fig 2.3: Use Case Diagram

2.4 Data Flow Diagram

We have three login options in our web-based application.

- The Admin
- The Trainer
- The Trainee

After successful login the admin has the ability to add, view or remove the details of both trainee and trainer. The trainer has the ability to view the trainees under him and update only

their weight and fat that is their performance. The trainee can view his performance details and also view his trainer's name and phone number. The exercise that the trainee is supposed to do is also mentioned.

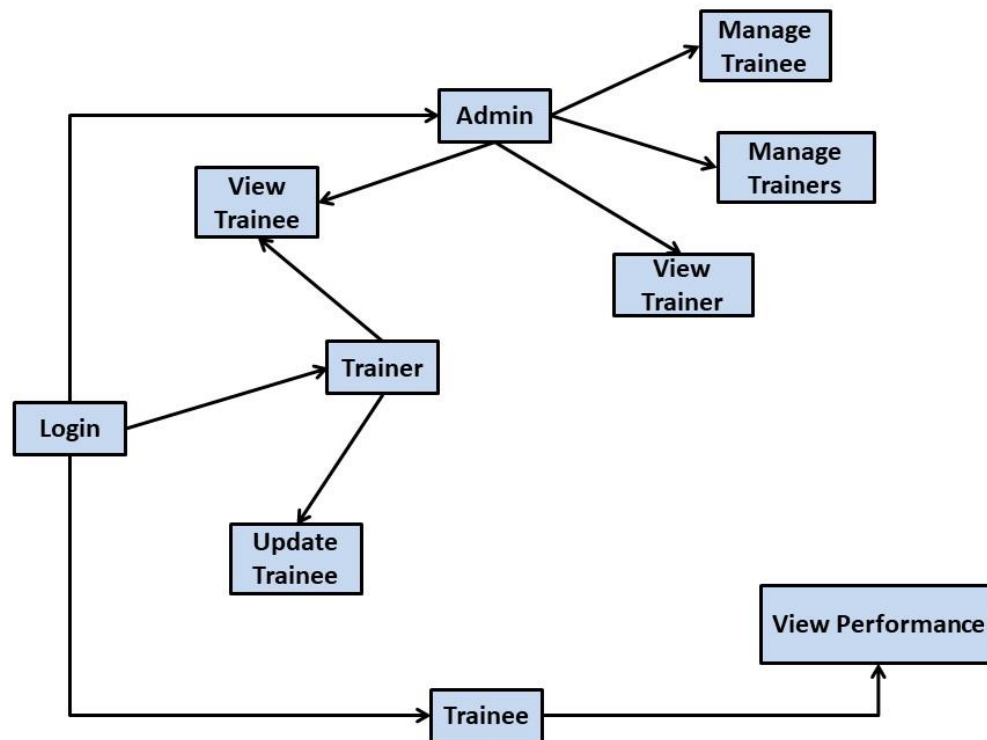


Fig 2.4: Data Flow Diagram

2.5 Sequence Diagram

Our project, the gym management system is a secure web-based computer application that only with thorough validation it lets all the three entities that is the admin, trainer and the trainee in to access the data and modify them. Figure below shows the sequence diagram of a secure login system by accessing the database from the backend.

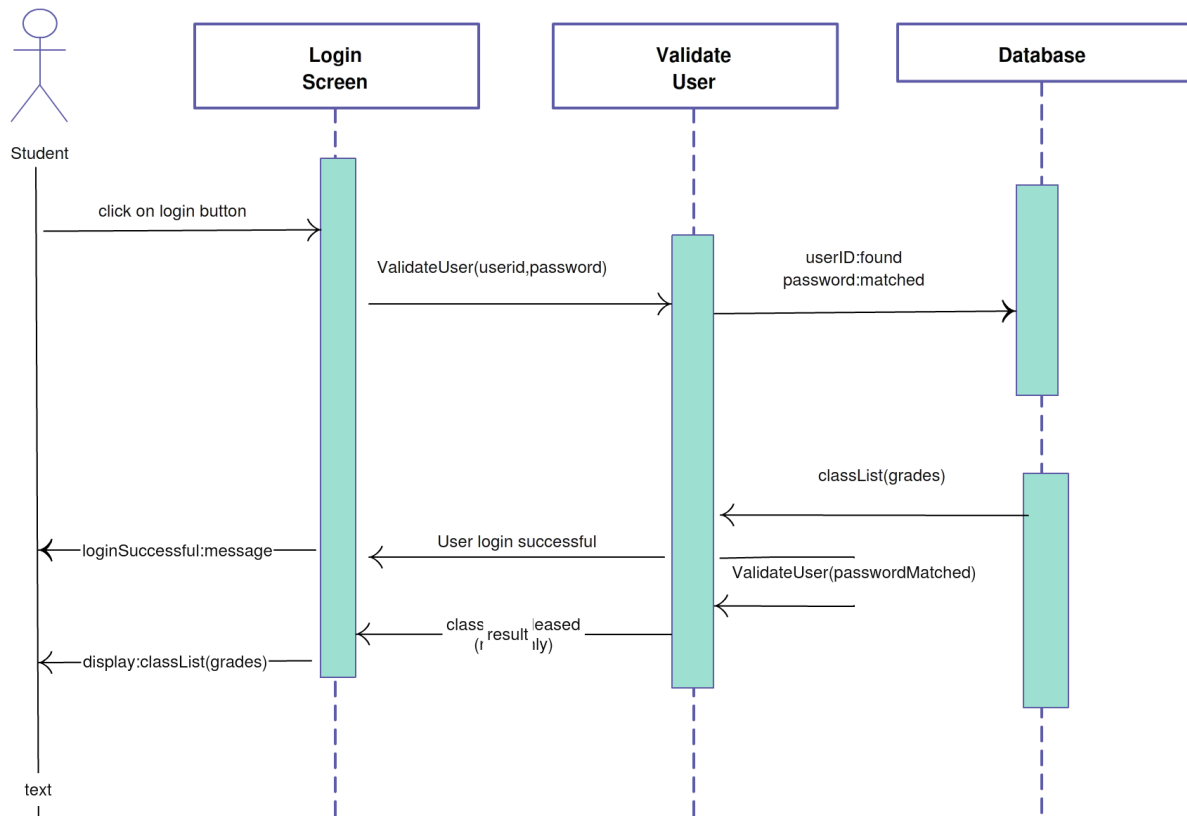


Fig 2.5 Sequence Diagram

Chapter 03

IMPLEMENTATION

3.1 Table Description

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	userid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	password	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	name	varchar(25)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	phone	bigint(20)			No	None			Change Drop More
<input type="checkbox"/> 5	salary	int(11)			No	None			Change Drop More

Fig 3.1.1: Trainer Table

Trainer Table - The table contains details of the trainers at the gym. It contains five attributes. User ID (userid), password, Trainer Name (name), Trainer's phone number (phone), and his salary (salary).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	userid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	password	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	name	varchar(25)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	phone	bigint(20)			No	None			Change Drop More
<input type="checkbox"/> 5	monthly_fee	int(11)			No	None			Change Drop More
<input type="checkbox"/> 6	trained_by	int(11)			No	None			Change Drop More

Fig 3.1.2: Trainee Table

Trainee Table - The table contains details of the trainees at the gym. It contains five attributes. User ID (userid), password, Trainee Name (name), Trainee's phone number (phone), the monthly fee that they are paying (monthly_fee) and his trainer's id (trained_by). The trained_by attribute is referenced from the trainer attribute to mention the trainer of each trainee.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	userid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	weight	int(11)			No	None			Change Drop More
<input type="checkbox"/> 3	fat	int(11)			No	None			Change Drop More
<input type="checkbox"/> 4	plan	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More

Fig 3.1.3: Trainee details table

Trainer_trainee - This table contains performance details of the trainee. It has the trainee's user id (userid) which is referenced from the trainee table, the current weight of the trainee(weight), current body fat content of the trainee (fat) and finally the plan that they opted for (plan).

3.2 Constraints on Table

In all the table the User ID (userid) or user name (username) in admin table is considered primary key. This key helps us access the tuples of the table easily. In the table trainee the trainee trainer's id (trained_by) is a foreign key constraint which is referenced from the trainer table (userid). This is used to know the trainer of each trainee.

In the table trainer-trainee where the performance details of the trainee is stored the user id of the trainee is a foreign key constraint which is directly referenced from the trainee table. This is to retrieve the performance details of each trainee with their user id.

The primary helps us uniquely identify the tuple or the row, the primary can neither be repeated nor kept null whereas the foreign key helps us to maintain the relation between two tables. Foreign helps us access data of one table from another.

3.3 Back End Implementation

Creating database---

```
CREATE DATABASE 'project_db';
```

```
USE 'project_db';
```

Table structure for table trainer---

```
CREATE TABLE 'trainer'(  
  
    'userid' INT PRIMARY KEY NOT NULL,  
  
    'password' VARCHAR(15) NOT NULL,  
  
    'name' VARCHAR(25) NOT NULL,  
  
    'phone' BIGINT NOT NULL,
```


'salary' INT NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Dumping data into the 'trainer' table---

INSERT INTO 'trainer' VALUES

(101100,'gary@!','Gary',845621789,12000)

(101101,'mandy@!','Mandy',9856321478,12000)

(101102,'sam@!','Sam Hunter',8485246789,15000)

(101106,'saina@!','Saina Pramod',985621789,14000)

(101109,'larry@!','Larry Stylinson',9874563789,13000);

Table structure for table trainee---

CREATE TABLE 'trainee' (

'userid' INT PRIMARY KEY NOT NULL,

'password' VARCHAR(15) NOT NULL,

'name' VARCHAR(25) NOT NULL,

'phone' BIGINT NOT NULL,

'monthly_fee' INT NOT NULL,

'trained_by' INT,

FOREIGN KEY ('trained_by') REFERENCES 'trainer'('userid')

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data into table 'trainee'---

```
INSERT INTO TABLE 'trainee' VALUES (  
    (201100,'sid@!','Sidharth',845621789,900)  
    (201101,'simran@!','Simran',9856321478,800)  
    (201102,'joy@!','Joy Pearce',8485246789,1200)  
    (201108,'saniya@!','Saniya Mira',985621789,1500)  
    (201110,'harry@!','Harry Linson',9874563789,900),  
    (201169,'louis@!','Louis Linson',7894563789,1000);
```

Table structure for table trainer-trainee---

```
CREATE TABLE 'trainer-trainee' (  
    'userid' INT NOT NULL,  
    'weight' INT NOT NULL,  
    'fat' INT NOT NULL,  
    'plan' VARCHAR(30) NOT NULL,  
    FOREIGN KEY (userid) REFERENCES 'trainee'('userid')  
)  
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data into table 'trainer-trainee'---

```
INSERT INTO 'trainer-trainee' VALUES
    (201100,75,28,'6 month'),
    (201101,59,25,'1 year'),
    (201102,70,28,'5 month'),
    (201108,62,22,'2 years'),
    (201110,69,28,'4 month'),
    (201169,70,30,'3 month');
```

3.4 Front End Implementation

- LOGIN.php

```
<?php
```

```
// define database related variables
```

```
$db_name = 'project_db';
```

```
$host = 'localhost';
```

```
$user = 'root';
```

```
$password = "";
```

```
$db = mysqli_connect($host, $user, $password, $db_name);
```

```
if(mysqli_connect_errno()) {
```

```
    die("Failed to connect with MySQL: ". mysqli_connect_error());
```

```
}
```

```
session_start();
```

```
$error="";
```

```
if($_SERVER["REQUEST_METHOD"] == "POST") {  
  
    if( $_POST['user'] == "Admin"){  
  
        $myusername = mysqli_real_escape_string($db,$_POST['username']);  
  
        $_SESSION['username']=$myusername;  
  
        $mypassword = mysqli_real_escape_string($db,$_POST['psw']);  
  
        $sql = " SELECT * from admin where username = '$myusername' and password =  
'$mypassword'";  
  
        $result = mysqli_query($db,$sql) ;//or die( mysqli_error($db));  
  
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);  
  
        $count = mysqli_num_rows($result);  
  
  
        if($count == 1) {  
  
            header("location: admin_home.php");  
  
        }else {  
  
            $error = "Invalid Username or Password";  
  
        }  
  
    }  
  
    elseif( $_POST['user'] == "Trainer"){  
  
        $myusername = mysqli_real_escape_string($db,$_POST['username']);  
  
        $_SESSION['username']=$myusername;  
  
        $mypassword = mysqli_real_escape_string($db,$_POST['psw']);  
  
        $_SESSION['password']=$mypassword;  
  
        $sql = " SELECT * FROM trainer WHERE userid = '$myusername' and password =  
'$mypassword'";
```

```
$result = mysqli_query($db,$sql) ;//or die( mysqli_error($db));

$row = mysqli_fetch_array($result,MYSQLI_ASSOC);

$count = mysqli_num_rows($result);

if($count == 1) {

    header("location: trainer_home.php");

}else {

    $error = "Invalid Username or Password";

}

}

elseif( $_POST['user'] == "Trainee")

{

    $myusername = mysqli_real_escape_string($db,$_POST['username']);

    $_SESSION['username']=$myusername;

    $mypassword = mysqli_real_escape_string($db,$_POST['psw']);

    $_SESSION['password']=$mypassword;

    $sql = " SELECT * FROM trainee WHERE userid = '$myusername' and password =

'$mypassword'";

    $result = mysqli_query($db,$sql) ;

    $row = mysqli_fetch_array($result,MYSQLI_ASSOC);

    $count = mysqli_num_rows($result);

    if($count == 1) {

        header("location: trainee.php");

    }else {

        $error = "Invalid Username or Password";

    }

}
```

```
}  
  
}  
  
?>
```

- ADMIN_HOME.php

```
<?php  
  
include('admin.php');  
  
$sql1="SELECT count(*) from trainee";  
  
$result = mysqli_query($db,$sql1);  
  
$row= mysqli_fetch_assoc($result);  
  
$trainee_count=$row['count(*)'];  
  
$sql2="SELECT count(*) from trainer";  
  
$result1 = mysqli_query($db,$sql2);  
  
$row= mysqli_fetch_assoc($result1);  
  
$trainer_count=$row['count(*)'];  
  
$total=$trainee_count+$trainer_count;  
  
?>
```

- ADD_TRAINER.php

```
<?php  
  
include('admin.php');  
  
?>  
  
<?php  
  
if($_SERVER["REQUEST_METHOD"] == "POST") {  
  
    $flag=0;
```

```
$myusername = mysqli_real_escape_string($db,$_POST['username']);

$password = mysqli_real_escape_string($db,$_POST['password']);

$name = mysqli_real_escape_string($db,$_POST['name']);

$myphone = mysqli_real_escape_string($db,$_POST['phone']);

$mysalary = mysqli_real_escape_string($db,$_POST['salary']);

$sql="INSERT INTO trainer
values('$myusername','$password','$name','$myphone','$mysalary')";

$result = mysqli_query($db,$sql) ;

if($result){

    $flag=1;

}

}
```

?>

- ADD_TRAINEE.php

```
<?php

include('admin.php');

?>

<?php

if($_SERVER["REQUEST_METHOD"] == "POST") {

    $flag=0;

    $myusername = mysqli_real_escape_string($db,$_POST['username']);

    $password = mysqli_real_escape_string($db,$_POST['password']);

    $name = mysqli_real_escape_string($db,$_POST['name']);

    $myphone = mysqli_real_escape_string($db,$_POST['phone']);
```

```
$myfee = mysqli_real_escape_string($db,$_POST['fee']);

$mytrainer = mysqli_real_escape_string($db,$_POST['trainer']);

$myweight = mysqli_real_escape_string($db,$_POST['weight']);

$myfat = mysqli_real_escape_string($db,$_POST['fat']);

$myplan = mysqli_real_escape_string($db,$_POST['plan']);

$sql="INSERT INTO trainee

values('$myusername','$mypassword','$myname','$myphone','$myfee','$mytrainer')";

$result = mysqli_query($db,$sql) ;

$sql1="INSERT INTO trainer_trainee

values('$myusername','$myweight','$myfat','$myplan')";

$result1 = mysqli_query($db,$sql1) ;

if($result && $result1){

    $flag=1;

}

}
```

?>

- ALL_TRAINEE.php

```
<?php

include('admin.php');

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">
```



```
<title>Trainer</title>

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

</head>

<body>

    <div class=view>

        <table>

            <tr>

                <th>User ID</th>

                <th>Name</th>

                <th>Phone</th>

                <th>Monthly Fee</th>

                <th>Plan</th>

            </tr>

            <?php

                $sql = " SELECT T.userid , T.name , T.phone , T.monthly_fee ,M.plan FROM trainee
T , trainer_trainee M where T.userid=M.userid;";

                $result = mysqli_query($db,$sql) ;

                while($row= mysqli_fetch_assoc($result)){

                    echo "<tr>";

                    echo "<td>" . $row['userid'] . "</td>";

                    echo "<td>" . $row['name'] . "</td>";

                    echo "<td>" . $row['phone'] . "</td>";

                    echo "<td>" . $row['monthly_fee'] . "</td>";

                    echo "<td>" . $row['plan'] . "</td>";

                    echo "</tr>";
```

```
}  
  
echo "</table>";  
?></div>
```

```
</body>
```

```
</html>
```

- ALL_TRAINER.php

```
<?php  
  
include('admin.php');  
  
?>  
  
<!DOCTYPE html>  
  
<html lang="en">  
  
  <head>  
  
    <meta charset="utf-8">  
  
    <title>Trainer</title>  
  
    <link rel="stylesheet" href="stylesheet/trainer.css">  
  
  </head>  
  
  <body>  
  
    <div class=view>  
  
      <table>  
  
        <tr>  
  
          <th>User ID</th>  
  
          <th>Name</th>
```

```
<th>Phone</th>

<th>Salary</th>

<th>Trainees</th>

</tr>

<?php

    $sql = " SELECT T.userid , T.name , T.phone , T.salary ,count(M.userid) as trainee
FROM trainer T , trainee M where T.userid=M.trained_by group by trained_by;";

    $result = mysqli_query($db,$sql) ;

while($row= mysqli_fetch_assoc($result)){

    echo "<tr>";

    echo "<td>" . $row['userid'] . "</td>";

    echo "<td>" . $row['name'] . "</td>";

    echo "<td>" . $row['phone'] . "</td>";

    echo "<td>" . $row['salary'] . "</td>";

    echo "<td>" . $row['trainee'] . "</td>";

    echo "</tr>";

}

echo "</table>";

?>

</div>

</body>

</html>
```

- DELETE_MEMBER.php

<?php

```
include('admin.php');
```

```
if($_SERVER["REQUEST_METHOD"] == "POST")
```

```
{
```

```
    $flag1=0;
```

```
    $myuserid = mysqli_real_escape_string($db,$_POST['username']);
```

```
    $sq=" SELECT * FROM trainer T, trainee M where T.userid='$myuserid' OR  
M.userid='$myuserid'";
```

```
    $result=mysqli_query($db,$sq);
```

```
    if(mysqli_num_rows($result) >= 1)
```

```
    {
```

```
        if( $_POST['user'] == "Trainer")
```

```
        {
```

```
            $sql=" DELETE FROM trainer where userid='$myuserid' ";
```

```
            if(mysqli_query($db,$sql)){
```

```
                $flag1=1;
```

```
            }
```

```
        }
```

```
        if( $_POST['user'] == "Trainee"){
```

```
            $sql1="DELETE FROM trainee where userid='$myuserid' ";
```

```
            if(mysqli_query($db,$sql1)){
```

```
                $flag1=1;
```

```
            }
```

```
        }
```

```
}  
  
}  
  
?>
```

- TRAINER.php

```
<?php  
  
// define database related variables  
  
$db_name = 'project_db';  
  
$host = 'localhost';  
  
$user = 'root';  
  
$password = '';  
  
$db = mysqli_connect($host, $user, $password, $db_name);  
  
if(mysqli_connect_errno()) {  
  
    die("Failed to connect with MySQL: ". mysqli_connect_error());  
  
}  
  
session_start();  
  
?>
```

- TRAINER_VIEW.php

```
<?php  
  
include('trainer.php');  
  
?>  
  
<!DOCTYPE html>  
  
<html lang="en">  
  
    <head>
```

```
<meta charset="utf-8">

<title>Trainer</title>

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

</head>

<body>

  <div class=view>

    <table>

      <tr>

        <th>Name</th>

        <th>Phone</th>

        <th>Plan</th>

        <th>Weight</th>

        <th>Fat</th>

      </tr>

      <?php

        $myusername=$_SESSION['username'];

        $sql = " SELECT M.name, M.phone, T.plan, T.weight, T.fat FROM trainee M,
trainer_trainee T WHERE M.userid=T.userid and trained_by = '$myusername' ";

        $result = mysqli_query($db,$sql) ;

        while($row= mysqli_fetch_assoc($result)){

          echo "<tr>";

          echo "<td>" . $row['name'] . "</td>";

          echo "<td>" . $row['phone'] . "</td>";

          echo "<td>" . $row['plan'] . "</td>";

          echo "<td>" . $row['weight'] . "</td>";
```

```
echo "<td>" . $row['fat'] . "</td>";

echo "</tr>";

}

echo "</table>";

?>

</div>

</body>

</html>
```

- TRAINER_UPDATE.php

```
<?php

include('trainer.php');

?>

<?php

$myusername=$_SESSION['username'];

$sqlList = " SELECT * FROM trainee where trained_by = '$myusername'";

$listResult = mysqli_query($db,$sqlList);

if($_SERVER["REQUEST_METHOD"] == "POST") {

    $myuserid = mysqli_real_escape_string($db,$_POST['username']);

    $myweight = mysqli_real_escape_string($db,$_POST['weight']);

    $myfat = mysqli_real_escape_string($db,$_POST['fat']);

    $sql = " UPDATE trainer_trainee SET weight='$myweight', fat='$myfat' where userid =

    '$myuserid'";
```

```
$result = mysqli_query($db,$sql) ;//or die( mysqli_error($db));

if($result){

    $var=1;

}

else{

    $var=0;

}

}

?>
```

- TRAINEE.php

```
<?php

// define database related variables

$db_name = 'project_db';

$host = 'localhost';

$user = 'root';

$password = "";

$db = mysqli_connect($host, $user, $password, $db_name);

if(mysqli_connect_errno()) {

    die("Failed to connect with MySQL: ". mysqli_connect_error());

}

session_start();

$myusername=$_SESSION['username'];

$mypassword=$_SESSION['password'];
```


	1	2	3	4	5
1	1				
2		1			
3			1		
4				1	
5					1

Table 1

Chapter 04

RESULT ANALYSIS

4.1 Snap Shots

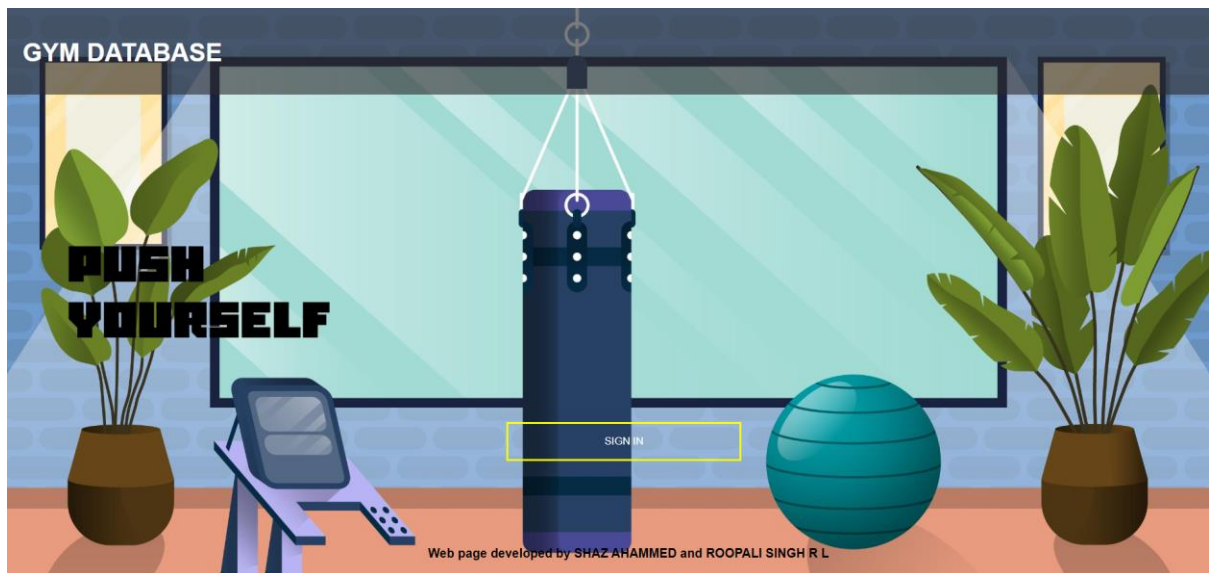


Fig 4.1.1: Home Page

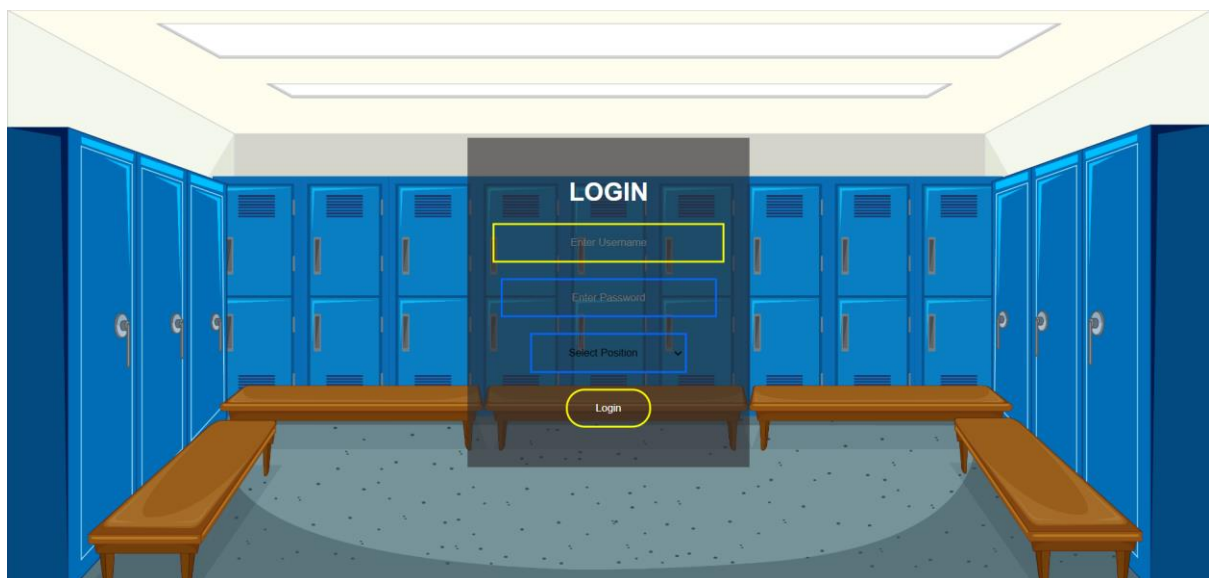


Fig 4.1.2: Login Page

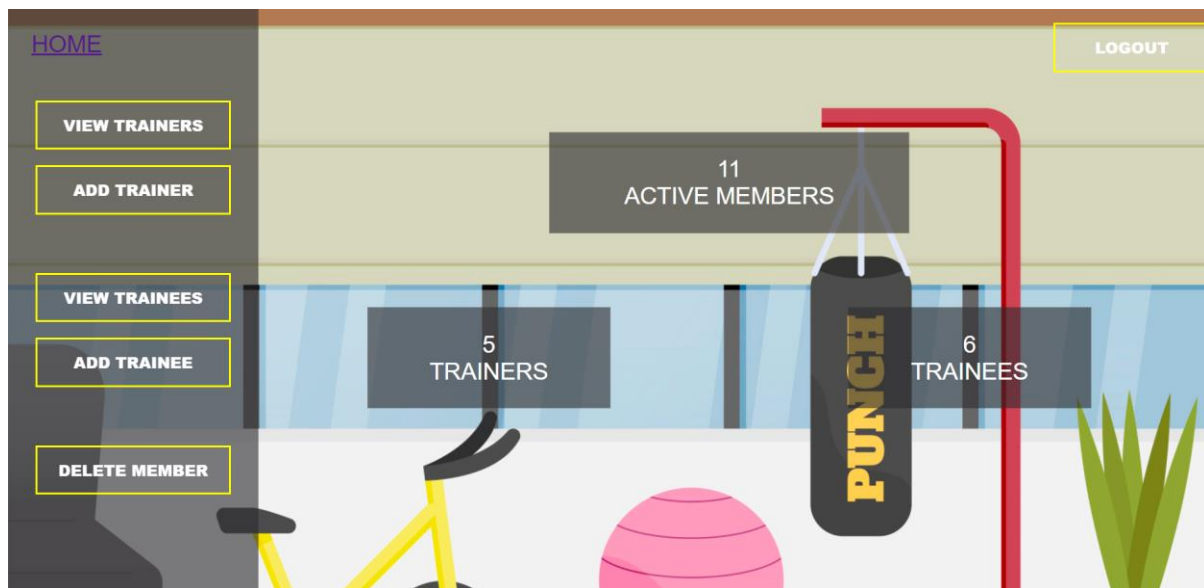


Fig 4.1.3: Admin Home Page

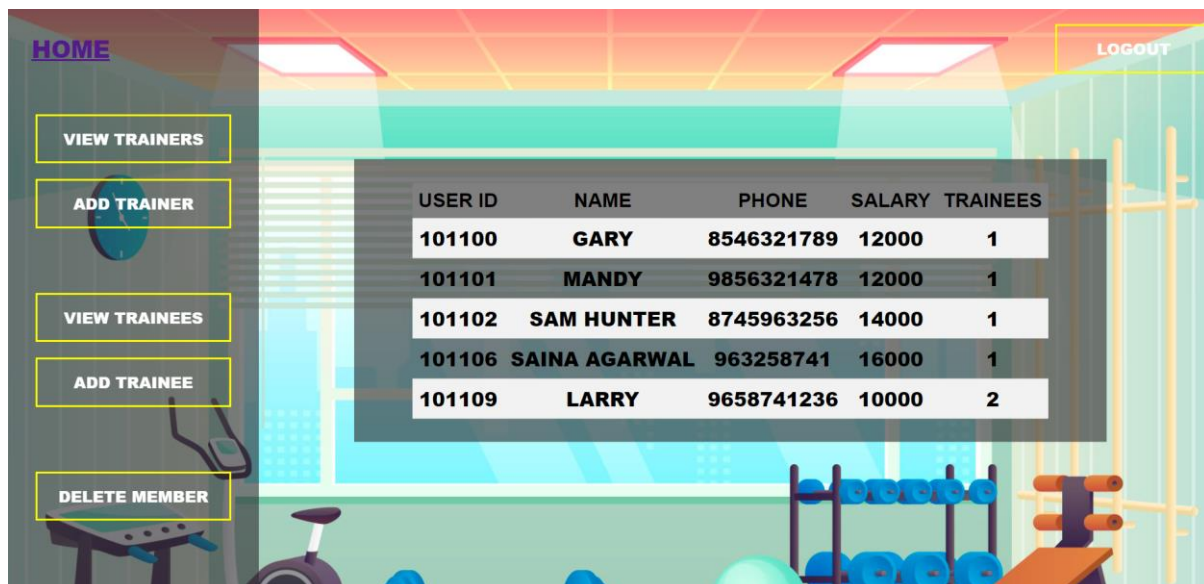
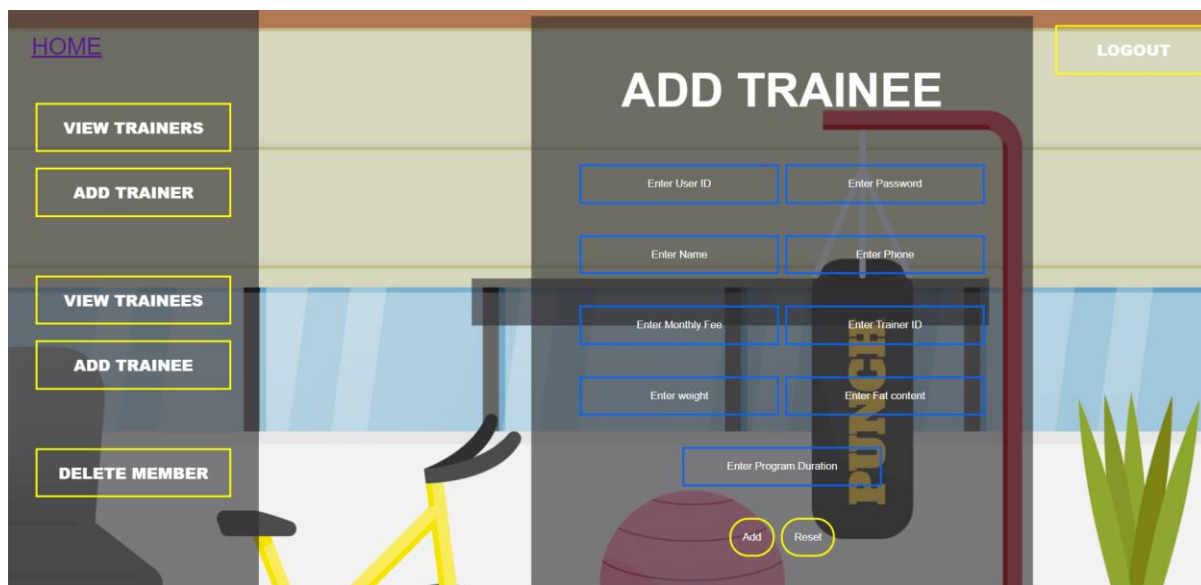


Fig 4.1.4: Trainer list Page

Fig 4.1.5: Add Trainer Page

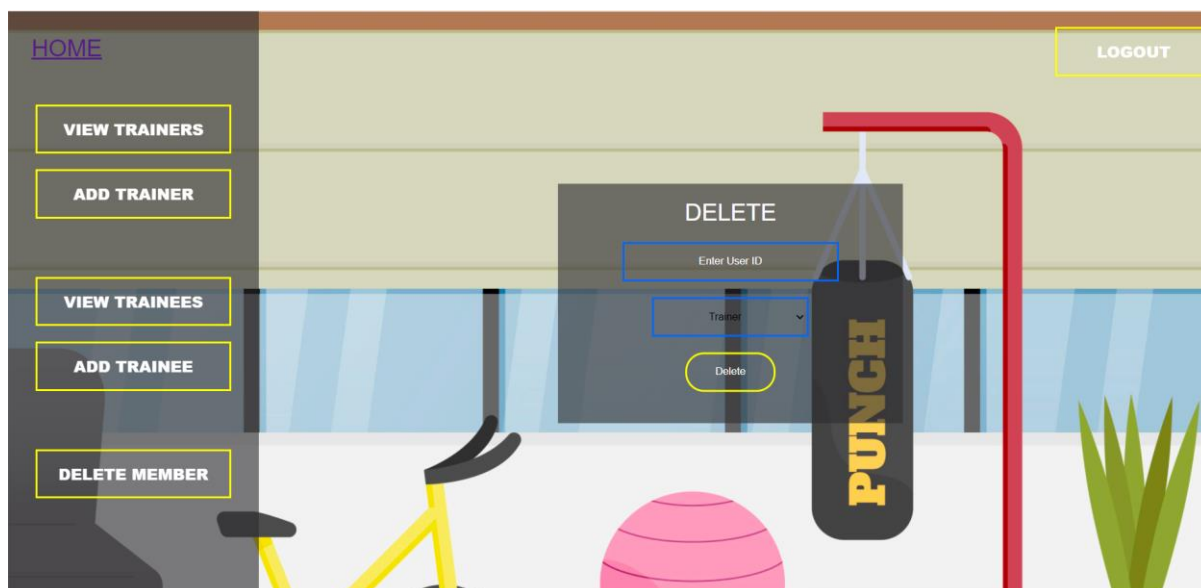
USER ID	NAME	PHONE	MONTHLY FEE	PLAN
201100	SIDHARTH	7856321498	900	6 MONTH
201101	SIMRAN	8246712548	800	1 YEAR
201102	JOY PEARCE	8524569317	1200	5 MONTHS
201108	SANIYA MIRA	8745632198	1500	2 YEARS
201110	HARRY LINSON	7896541258	900	4 MONTHS
201169	LOUIS LINSON	7896548796	1000	3 MONTHS

Fig 4.1.6: Trainee list Page



The 'ADD TRAINEE' page features a sidebar on the left with navigation links: HOME, VIEW TRAINERS, ADD TRAINER, VIEW TRAINEES, ADD TRAINEE, and DELETE MEMBER. A LOGOUT button is located in the top right corner. The main content area is titled 'ADD TRAINEE' and contains a form with the following fields: Enter User ID, Enter Password, Enter Name, Enter Phone, Enter Monthly Fee, Enter Trainer ID, Enter weight, Enter Fat content, and Enter Program Duration. At the bottom of the form are 'Add' and 'Reset' buttons. The background of the page is a gym-themed illustration with a yellow bicycle, a pink exercise ball, a black punching bag labeled 'PUNCH', and a green plant.

Fig 4.1.6: Add Trainee Page



The 'DELETE' page features the same sidebar and navigation elements as the 'ADD TRAINEE' page. The main content area is titled 'DELETE' and contains a form with the following fields: Enter User ID, a Trainer dropdown menu, and a Delete button. The background is the same gym-themed illustration as the previous page.

Fig 4.1.7: Delete Page

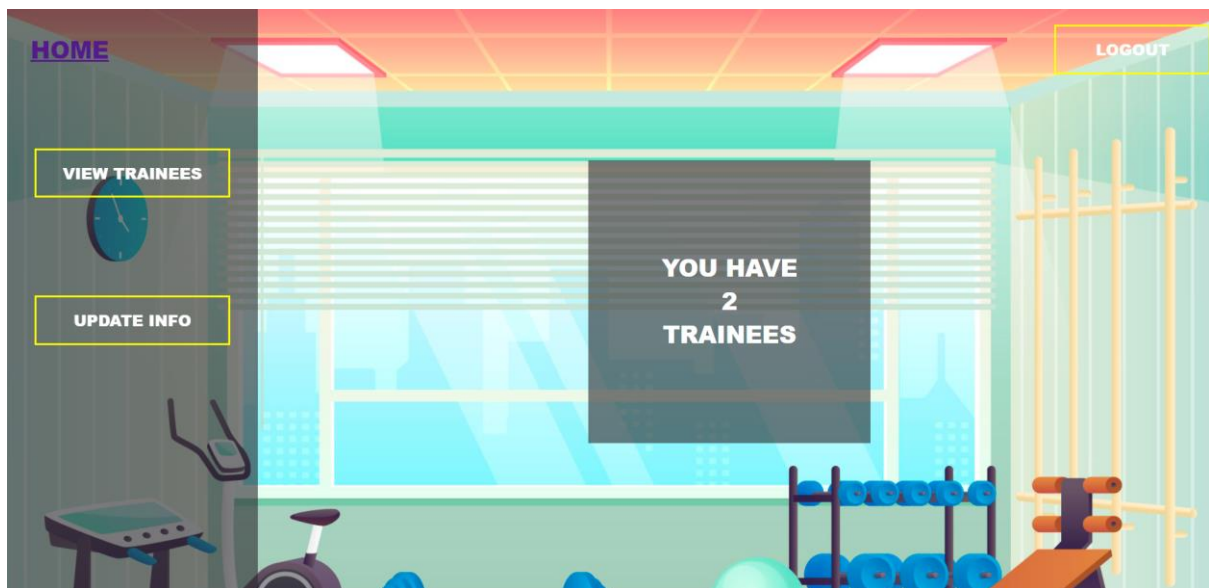


Fig 4.1.8: Trainer Home Page



Fig 4.1.9: Trainee Detail Page

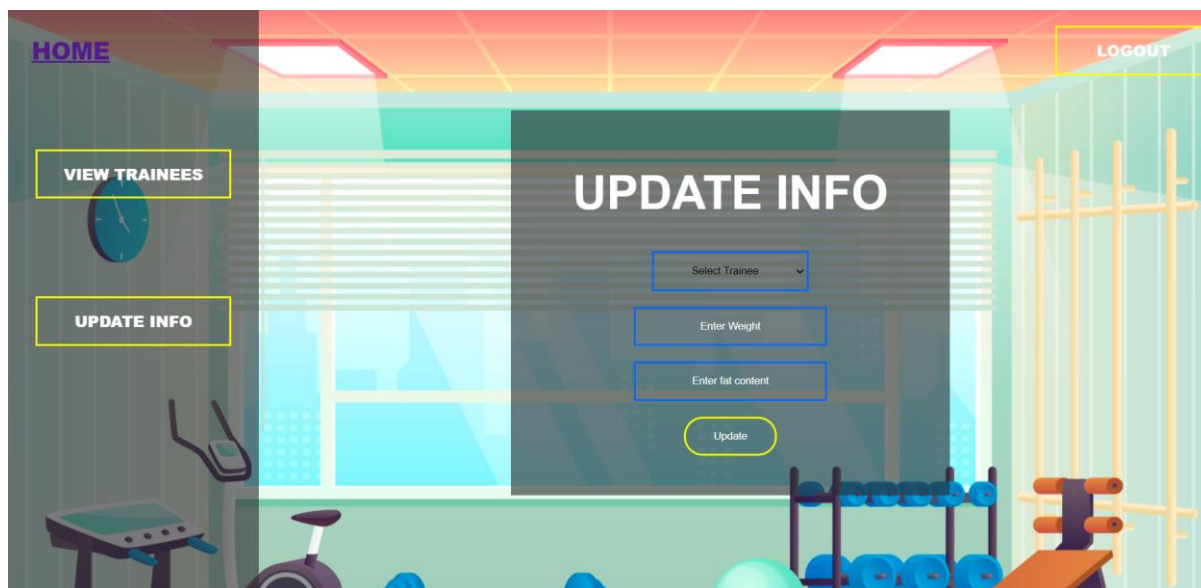


Fig 4.1.5: Trainee Update Page

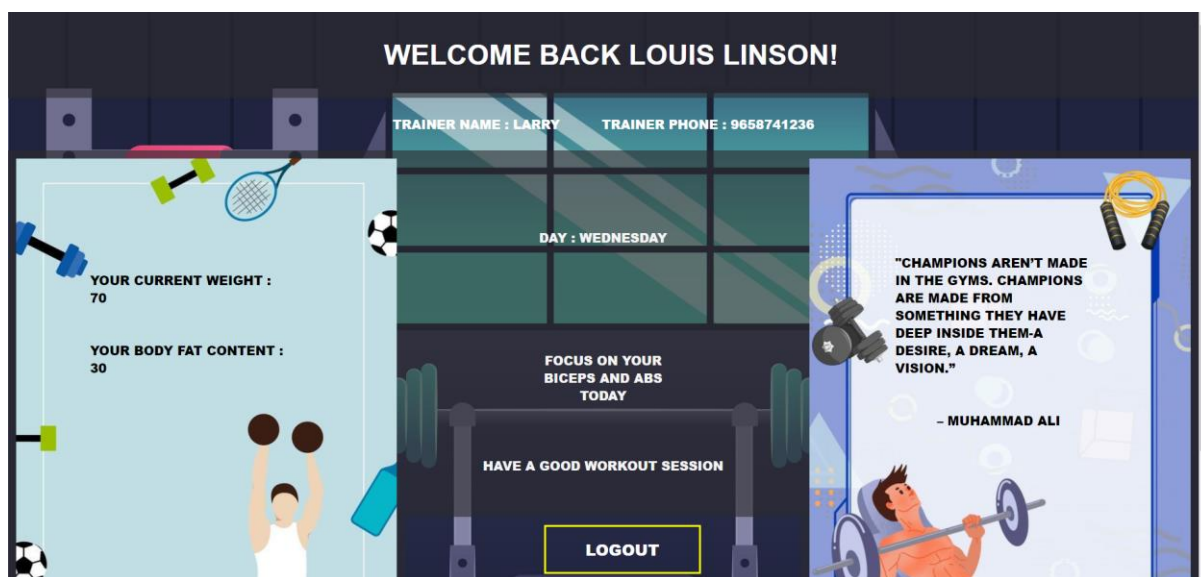


Fig 4.1.5: Trainee Home Page

4.2 Discussion

- Login Page

The login page lets the admin, trainer and trainee to login to their respective home pages for further details and processing.

- Admin Home Page

The admin home page gives the number of trainers, trainees and the total number of active members on the gym (trainer + trainee).

- Trainer list Page

The trainer list page shows the basic details of all trainers such as user id, name, phone, salary.

- Add Trainer Page

The add trainer page lets the admin add a new trainer into the database. All the basic details of the trainer should be provided in order to enter the trainer into the database successfully.

- Trainee list Page

The trainee list page shows the basic details of all trainees such as user id, name, phone, monthly fee.

- Add Trainee Page

The add trainer page lets the admin add a new trainee into the database. All the basic details of the trainer should be provided in order to enter the trainee into the database successfully.

- Trainer Home Page

The trainer home page gives the number of trainees that are under the particular trainer who logged in.

- Trainee Details Page

The trainee details page shows the performance details such as weight, fat and the plan of all trainees who is under him.

- Trainee Update Page

The trainee update page lets any trainer to update the performance of trainees if the trainee is under him.

- Trainee Home Page

The trainee home page shows the trainee several details such as his trainer's name and phone number, his performance details and so on.

4.3 Testing

System testing is actually a series of different trails whose primary purpose is to fully exercise the computer-based system. Software testing is critical element of software quality

assurance and represent the ultimate review of the specification, design and coding. System testing makes a logical assumption that all the part of the system is correct; the goal will be successfully achieved. Testing is the final verification and validation activity within the organization itself. During testing the major activities are concerned on the examinations and modification of the source code.

SL NO	DESCRIPTION	INPUT	OUTPUT	RESULT
1	Login	Correct username and password	Login successful	Pass
2	Login	Wrong username and password	Login unsuccessful	Pass
3	Add Trainee	All valid details	Trainee details added successfully	Pass
4	Add Trainee	Invalid details	There was an unexpected error	Pass
5	Add Trainer	All valid details	Trainer details added successfully	Pass
6	Add Trainer	Invalid details	There was an unexpected error	Pass
7	Delete Trainer or Trainee	Valid user ID	Deleted successfully	Pass
8	Delete Trainer or Trainee	Invalid user ID	There was an unexpected	Pass
9	Update trainee Information	Valid user ID	Update successful	Pass
10	Update trainee Information	Invalid user ID	Update not successful	Pass

Table 4.3 Test Cases

Chapter 05

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The application is designed in such a way that any further enhancements can be done with ease. The system has the capability for easy integration with other systems. New modules can be added to the existing system with less effort. The system has six classes. Each of these classes has various procedures and functions. In future a new function or procedure can be easily added in the system through these classes. Or even a new class can be added. The system generates only a limited number of reports. If more detailed reports are required the system can be directed. Even though the system has well communication facility, it's not enough. The mail service can be enhanced with features bcc, cc etc. the system has full security but the account information for the customer credit information. Thus by adding this module the system transaction will be improved.

5.2 Future Enhancement

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

Chapter 06

REFERENCES

Learn HTML5, CSS3, PHP, JAVA Script from W3Schools an online web tutorial.

- Website link: <https://www.w3schools.com>

Referred some web designing video tutorials from YouTube.

- <https://www.youtube.com/watch?v=wHFflWvii3M&t=140s>
- <https://www.youtube.com/watch?v=IBfshkPIMW8&t=1974s>