# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



**An Internship Project Report**

# GYM MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

**Submitted by**

|  |
| --- |
| **VARSHA PS** |
| **1RN18IS118** |

|  |
| --- |
| **Faculty In charge** |
| **Mr. T.S.Bhagavath Singh** |
| **Associate Professor** |



**Department of Information Science and Engineering RNS Institute of Technology**

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post Bengaluru – 560 098

**2021 – 2022**

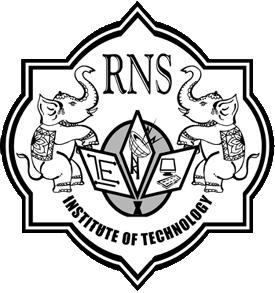
**RNS INSTITUTE OF TECHNOLOGY**

Dept. of ISE, RNSIT 2018-2019 Page 2

**Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,**

**Channasandra, Bengaluru - 560098**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**



**CERTIFICATE**

Certified that the Internship work entitled ***GYM MANAGEMENT SYSTEM*** has been successfully completed by **Varsha PS (1RN18IS118)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

|  |  |  |
| --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Mr. T.S Bhagavath Singh** | **Dr. Suresh L** | **Dr. M K Venkatesha** |
| Internship Guide | Professor and HoD | Principal |
| Associate Professor | Department of ISE | RNSIT |
| Department of ISE | RNSIT |  |
|  |  |  |
|  | **External Viva** |  |
| **Name of the Examiners** |  | **Signature with Date** |
| **1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

# ABSTRACT

Gym Management system is the solution for all the fitness centres to manage its members and their respective information in an easier and more convenient way. The administrator is the user in this system. Since it’s a tedious job to maintain the record of each and every member of the gym using paper work, this mini project serves as the best tool for management of data.

This mini project is basically developed to aid the user to perform certain actions like adding a new member to the gym, managing payments, workout routines etc. The user shall be able to add the name, date of birth, contact address of the member, exercise routine and month of joining the gym. It also records the phone numbers, height, weight and calorie count of every member. It also has option for maintaining the payment transactions. This helps the user to know about fee payments and select the type of exercise routines like monthly, quarterly, 6 months or annually depending on the choice of the member. It also holds the receipt number and the amount of fees paid. The health status of each and every member is maintained and updated frequently. Admin can be managing, adding and updating the information of the members of the gym by overcoming the limitations and drawbacks of the conventional paperwork.

**ACKNOWLEDGMENT**

I, **VARSHA PS [USN: 1RN18IS118]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled  ***GYM MANAGEMENT SYSTEM*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

**VARSHA PS**

**(1RN18IS118)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **TABLE OF CONTENTS** |  |
| **CERTIFICATE** | | | | |
| [**ABSTRACT**](#_bookmark0) | | |  | [**i**](#_bookmark0) |
| [**ACKNOWLEDGEMENT**](#_bookmark1) | | | | [**ii**](#_bookmark1) |
| [**TABLE OF CONTENTS**](#_bookmark2) | | | | [**iii**](#_bookmark2) |
| [**LIST OF FIGURES**](#_bookmark3) | | | | **vi** |
| [**LIST OF TABLES**](#_bookmark4) | | | | **vii** |
| **ABBREVIATIONS** | | | | **viii** |
| [**1**](#_bookmark5) | [**INTRODUCTION**](#_bookmark5) | | | [**1**](#_bookmark5) |
|  | [1.1](#_bookmark5) | [Background](#_bookmark5) | | [1](#_bookmark5) |
|  | 1.2 | About the mini project | | 1 |
| [**2**](#_bookmark6) | **LITERATURE REVIEW** | | | **3** |
| [**3**](#_bookmark7) | [**ANALYSIS AND DESIGN OF THE SYSTEM**](#_bookmark7) | | | **6** |
|  | [3.1](#_bookmark7) | [Analysis of the system](#_bookmark7) | | 6 |
|  | 3.2 | Design of the system | | 6 |
|  |  | 3.2.1 Architecture of the system | | 6 |
|  |  | 3.2.2 Use-Case Diagram | | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
| [**4 IMPLEMENTATION**](#_bookmark8) | | | **9** |
| [4.1](#_bookmark8) | [Front-end and back-end used.](#_bookmark8) | | 9 |
|  | [4.1.1](#_bookmark8) | [Features of front-end - PHP.](#_bookmark8) | 9 |
|  | [4.1.2](#_bookmark8) | [Features of backend – MySQL (WAMP server).](#_bookmark8) | 10 |
| [4.2](#_bookmark9) | [Discussion](#_bookmark9) of code segments | | 11 |
|  | [4.2.1](#_bookmark9) | Database Connection | 11 |
|  | 4.2.2 | Admin Login | 11 |
|  | 4.2.3 | Health Status. | 12 |
| 4.3 | Applications of project work. | | 13 |
| **5 TESTING AND RESULTS** | | | **14** |
| 5.1 | Testing |  | 14 |
|  | 5.1.1 | Unit testing. | 14 |
|  |  | 5.1.1.1 Unit test case 1 | 14 |
|  | 5.1.2 | Integration testing | 15 |
|  |  | 5.1.2.1 Integration test case 1 | 15 |
|  |  | 5.1.2.2 Integration test case 2 | 15 |
|  | 5.1.3 | System testing | 16 |
|  |  | 5.1.3.1 System test case 1 | 16 |
| 5.2 | Discussion of results | | 17 |
|  | 5.2.1 Login page | | 17 |
|  | 5.2.2 Dashboard | | 17 |
|  | 5.2.3 Registration page | | 18 |
|  | 5.2.4 Payments page | | 18 |
|  | 5.2.5 Health status page | | 19 |
|  | 5.2.6 Workout routines page | | 19 |

1. INSTALLATION INSTRUCTIONS 20
   1. Installing WAMP Server 20
2. CONCLUSION AND FUTURE ENHANCEMENTS 22

[REFERENCES](#_bookmark10) 23

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| Figure 3.1 | System Data Flow Diagram | 7 |
| Figure 3.2 | System Use Case Diagram | 8 |
| Figure 5.1 | Login page | 17 |
| Figure 5.2 | Dashboard | 17 |
| Figure 5.3 | Registration page | 18 |
| Figure 5.4 | Payments page | 18 |
| Figure 5.5 | Health status page | 19 |
| Figure 5.6 | Workout routines page | 19 |

**Table No. Table Description Page No.**

|  |  |  |
| --- | --- | --- |
| Table 4.1 | Unit test case for Login check | 14 |
| Table 4.2 | Integration test case for Registration check | 15 |
| Table 4.3 | Integration test case for Changing profile check | 16 |
| Table 5.4 | System test case for Exercise Routine | 16 |

# ABBREVIATIONS

|  |  |
| --- | --- |
| CSS | Cascading Style Sheets |
| DFD | Data Flow Diagram |
| HTML5 | Hyper Text Markup Language version 5 |
| HTTP | Hyper Text Transfer Protocol |
| OS | Operating System |
| PHP | Hypertext Pre-processor |
| SRS | Software Requirements Specification |
| SQL | Structured Query Language |
| UCD | Use Case Diagram |
| XAMPP | Windows, Apache, MySQL and PHP |

**Chapter 1**

# INTRODUCTION

The Gym Management System is a web-based application. It provides a user-friendly, interactive interface based on PHP, CSS, JavaScript and HTML5 elements. All data is stored in a MySQL database. The application uses XAMPP Server to communicate with the database.

## Background

The purpose of the “Gym Management System” is to provide a system which handles the information of the people coming into the gym and maintaining their health care. The main motivation behind the selection of this project was to design, develop and implement software which will be a platform for management of all aspects of the gym. Further, to make the application user interactive and user-friendly, Web technologies like PHP, CSS, JavaScript and HTML5 are used.

## About the mini project

The Gym Management System is an interface which maintains the record of all the members of the gym, payments, workout routines, plans and health status. Administrator can log in to the system using the credentials which takes him to the main page. He can add a new member, update their health status which includes their calorie intake, height, weight and remarks. It also includes the workout plans like monthly, quarterly, 6 months and annual plans.

Admin can view, edit/update the users or the inventory along with payment option which specifies joining date, expiry date, workout routine and health status. Trainers need not use any paperwork for maintaining the information about his members, which will be a tedious job and also there are chances of it getting misplaced. This mini project in turn helps him/her to maintain the records in a database which can be updated, managed and changed at any point of time.

New members can be added in the registration column which includes all their basic details along with joining details. Random function is being used to avoid redundancy of the primary key. Workout routines can be viewed as well as edited to the likes of the administrator

and new plans can also be added. Members per month, year and plan can be viewed by selecting the year and month from the drop-down box. All the data entered in the front end will be stored in the backend in their respective tables. The mini project organizes, manages and updates the entire database which eases the work of the administrator.

This Gym Management System is developed to provide the following services:

* + - The Administrator can add new members to the gym by adding their credentials like name, address, phone number, joining date and exercise routine.
    - Health status of each and every member can be stored along with their calorie count, height, weight and remarks from their trainer.
    - Payment status of every member is maintained along with the exercise routine, joining date and expiry date of their plan.
    - Workout routine is included which includes four plans namely Annual, 6 months, Quarterly and Monthly.
    - Admin profile can also be changed by providing the previous password and updating the credentials of the new one.

**Chapter 2**

# LITERATURE REVIEW

* + .Kasliwal Mahima , Raundal Pooja , Wagh Niyati, G.M. Lodha , “Survey Paper on Gym Management System” , Journal of Advancement in Software Engineering and Testing Volume 2 Issue 3. In this paper the authors have mentioned the feature where you can login as admin and manage the customer/user details, which I have implemented in this project also learnt how to add certain categories like the user’s age , weight , height , etc as one of the informations to go into the database.
  + Eric Huong,“ Gym Review”, Scribd Research Volume 13. In this paper the author mentions about how physical fitness is important and how in this era that we live in Gym industry is booming and more and more help is needed in bringing them up for a profitable lifestyle, This lead me to come up with building this website which will be used so much now in the market and will also be benefitial to the fitness market.
  + Sanja Delcev , Drazen Draskovic , “ Modern Javascript Frameworks : A Survey Study” , Date of issue: 30 May 2018 , Added to IEEE on 30th August 2018 , DOI :10.1109/ZINC.2018.8448444 , the author here mentions that JavaScript is used in today's web applications as a client script language and on the server side. The JavaScript language supports the Model View Controller (MVC) architecture that maintains a readable code and clearly separates parts of the program code. The topic of this research is to compare the popular JavaScript frameworks: AngularJS, Ember, Knockout, Backbone. All four frameworks are based on MVC or similar architecture , this sounded as an appropriate framework to use in my project and thus I developed my website using Js as the Backend.
  + A.V Dinesh Kumar , K Bhargav Ram Rayal, M.Saraswathi , “Smart Gym Management System “ , SCSVMV University, Elanathur Kanchipuram – 631561 , I learnt that Increasing efficiency and effectiveness automation, accuracy, user-friendly interface, information availability, communication capacity, maintenance, cost reduction makes our system smarter than the existing system.

**Chapter 3**

# ANALYSIS AND DESIGN OF THE SYSTEM

### Analysis of the system

The Gym Management System includes:

* + - Gym Management: This includes maintaining the information about all the members, their health status, workout plans and payments.
    - PHP Technology used for the development of the application.
    - Web-platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimum.

### Design of the system

**Systems design** is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. It is the application of systems theory to product development. There is some overlap with system analysis, system architecture and system engineering.

#### Architecture of the system

A **Data Flow Diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing) (structured design).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured [flowchart](https://en.wikipedia.org/wiki/Flowchart) which focuses on control flow, or a UML activity workflow diagram, which presents both control and data, flows as a unified model.

Figure 3.1 shows the data flow diagram for the Gym Management System. It illustrates the flow of data from and to the Gym Management in the system. The administrator can add new

member to the database, as well as make payment transactions, maintain health status of each and every member, workout routine management and finally updating or adding a new admin profile to the system. The system uses a MySQL database to store all the required information in the form of relational tables. In response to user input, the system uses the database to store the member information, their health status, payment transactions, workout routines and admin information.

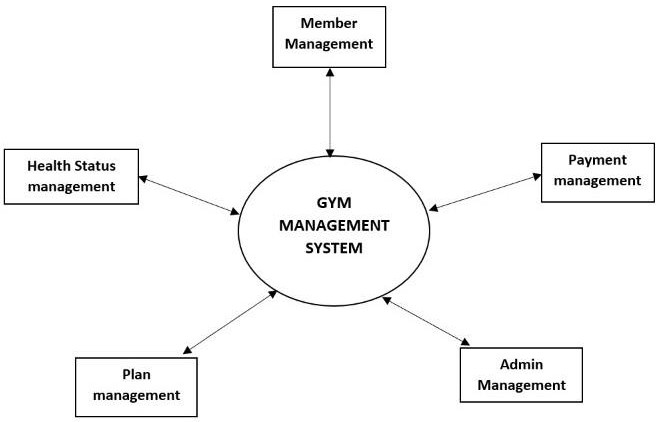


Figure 3.1 System Data Flow Diagram

#### Use-Case Diagram

A **Use-Case Diagram** (**UCD**) at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use-cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use-case diagram can identify the different types of users of a system and the different use-cases and will often be accompanied by other types of diagrams as well.

Figure 3.2 shows the use-case diagram for the Gym Management System. It illustrates the different actions that can be performed by administrators in order to interact with the application. Administrator can add a new member to the database, add new payments, update and add health status of each and every member of the gym, new workout routines, plans and also edit or add new admin profile to the database.

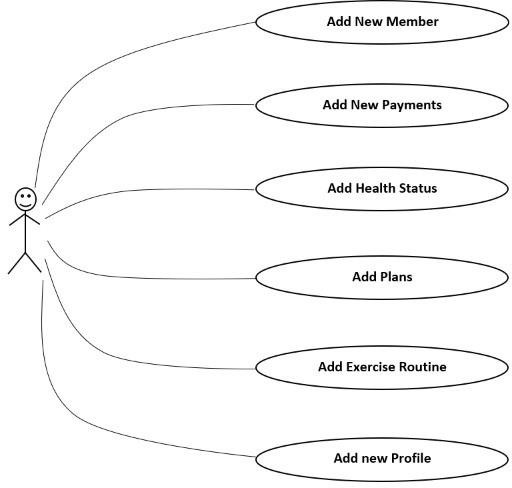


Figure 3.2 System Use-Case Diagram

Internship Project work Gym Management System

**Chapter 4**

# IMPLEMENTATION

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

## Front-end and back-end used

The front-end is everything involved with what the user sees. The back-end, or the "server-side", is basically how the site works, updates and changes. This refers to everything the user can't see in the browser, like [databases](http://pluralsight.com/training/courses/TableOfContents?courseName=intro-sql-server&highlight=dan-sullivan_what-is-relational&what-is-relational) and [servers.](http://pluralsight.com/training/courses/TableOfContents?courseName=sql-server-fundamentals&highlight)

## Features of front-end

HTML5 code along withCSS3 is used for styling while JavaScript, is used for validation at frontend. PHP is a server-side scripting language designed for Web development, but also used as a general-purpose programming language. PHP code is embedded into HTML5.

HTML 5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard, and subsumes XHTML. It includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications For the same reasons, HTML 5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind. The APIs and Document Object Model (DOM) are now fundamental parts of the HTML 5 specification and HTML 5 also better defines the processing for any invalid documents.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity. Its name cascading comes from the specified priority scheme to determine the style to be applied.

Dept. of ISE, RNSIT 2021-2022 Page 9

Internship Project work Gym Management System

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. Unlike CSS 2, which is a large single specification defining various features, CSS 3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility. Work on CSS level 3 started around the time of publication of the original CSS 2 recommendation. The earliest CSS 3 drafts were published in June 1999. Due to the modularization, different modules have different stability and statuses. Each web browser uses a layout engine to render web pages, and support for CSS functionality is not consistent between them. Because browsers do not parse CSS perfectly, multiple coding techniques have been developed to target specific browsers with workarounds (commonly known as CSS hacks or CSS filters). Adoption of new functionality in CSS can be hindered by lack of support in major browsers. For example, Internet Explorer was slow to add support for many CSS 3 features, which slowed adoption of those features and damaged the browser's reputation among developers. In order to ensure a consistent experience for their users, web developers often test their sites across multiple operating systems, browsers, and browser versions, increasing development time and complexity. Tools such as Browser Stack have been built to reduce the complexity of maintaining these environments.

PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical interface.

## Features of back-end

**XamppServer** refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

The **Apache HTTP Server**, colloquially called **Apache**, is a free and open-source cross- platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The Apache HTTP Server is cross-platform with Version 2.0 improved support for non-Unix operating systems such as Microsoft developed Windows operating system.

**MySQL** is an open-source relational database management system (RDBMS). MySQL is a central component of the XAMPP open-source web application software stack (and other "AMP" stacks). MySQL is written in C and C++. MySQL works on many system platforms, including Linux, macOS, Microsoft Windows, etc.

MySQL performs extremely well in the average case, the developer interfaces are there, and the documentation is very, very good. It has also been tested to be a fast, stable and true multi-user, multi-threaded SQL database server.

## Discussion of code segments

This section includes the segments of code used to provide various user functionalities.

## Database connection

The below code shows how to establish connection with database

<?php

define('DB\_HOST','localhost'); define('DB\_USER','root');

define('DB\_PASS',''); define('DB\_NAME','carrental'); try{

$dbh = new PDO("mysql:host=".DB\_HOST.";dbname=".DB\_NAME,DB\_USER, DB\_PASS,array(PDO::MYSQL\_ATTR\_INIT\_COMMAND => "SET NAMES 'utf8'"));

} catch (PDOException $e){ exit("Error: " . $e->getMessage());}

?>

## Admin login

The below code shows how admin logs in to the system by providing the credentials

<?php

session\_start(); include('includes/config.php'); if(isset($\_POST['login'])){

$email=$\_POST['username'];

$password=md5($\_POST['password']);

$sql ="SELECT UserName,Password FROM admin WHERE UserName=:email and Password=:password";

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

$query->bindParam(':email', $email, PDO::PARAM\_STR);

$query->bindParam(':password', $password, PDO::PARAM\_STR);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ); If($query->rowCount() > 0){

$\_SESSION['alogin']=$\_POST['username'];

echo "<script type='text/javascript'>document.location = 'dashboard.php';

</script>";

} else{

?>

echo "<script>alert('Invalid Details');</script>";}

}

## Health status

The below code shows the health status of the members

<?php if(isset($\_POST['submit'])){

$calorie=$\_POST['calorie'];

$height=$\_POST['height'];

$weight=$\_POST['weight'];

$fat=$\_POST['fat'];

$remarks=$\_POST['remarks'];

$userid=$\_POST['usrid'];

$query="update health\_status set calorie='".$calorie."', height='".$height."',weight='".$weight."',fat='".$fat."',remarks='".$remarks."' where uid='".$userid."'";

if(mysqli\_query($con,$query)){

echo "<head><script>alert('Health Status Added ');</script></head></html>"; echo "<meta http-equiv='refresh' content='0; url=new\_health\_status.php'>";

}

else{

echo "<head><script>alert('NOT SUCCESSFUL, Check Again');</script></head></html>";

echo "error".mysqli\_error($con);

echo "<meta http-equiv='refresh' content='0; url=new\_health\_status.php'>";

} }

else{

$uid=$\_POST['uid'];

$uname=$\_POST['uname'];

$udob=$\_POST['udob'];

$ujoin=$\_POST['ujoin'];

$ugender=$\_POST['ugender'];

$sql="select \* from health\_status where uid='".$uid."'";

$result=mysqli\_query($con,$sql); if($result){

$row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC);

$cal=$row['calorie'];

$hei=$row['height'];

$wei=$row['weight'];

$fa=$row['fat'];

$remar=$row['remarks'];

}

}

?>

## Applications of project work

* Connected platform
* Building a community
* End to end conversation
* Temporary group creation
* Secured conversations
* Sharing of confidential information

Internship Project work Gym Management System

**Chapter 5**

# TESTING AND RESULTS

## Testing

**Software testing** is conducted to provide stakeholders with information about the [quality](https://en.wikipedia.org/wiki/Software_quality) of the [software](https://en.wikipedia.org/wiki/Software) product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

## Unit testing

**Unit testing** is a [software testing](https://en.wikipedia.org/wiki/Software_testing) method by which individual units of [source code](https://en.wikipedia.org/wiki/Source_code), sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

It is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software.

## Unit test case 1

This test case checks to see if signed up users are able to login or not, as shown in Table 5.1.

Table 5.1 Unit test case for Login check

|  |  |
| --- | --- |
| SI No. of test case: | 1 |
| Name of test: | Login check |
| Item / Feature being tested: | Login |
| Sample Input: | Enter username and password fields and click on Login button. |
| Expected output: | Message 'Login success, Welcome user' is displayed. |
| Actual output: | Message ‘Login success, Welcome user' is displayed. |
| Remarks: | Test succeeded |

## Integration testing

**Integration testing** is the phase in [software testing](https://en.wikipedia.org/wiki/Software_testing) in which individual software modules are combined and tested as a group. It occurs after [unit testing](https://en.wikipedia.org/wiki/Unit_testing) and before [validation testing](https://en.wikipedia.org/wiki/Software_verification_and_validation). The purpose of this level of testing is to expose faults in the interaction between integrated

units. Test drivers and test stubs are used to assist in Integration Testing.

It is the phase in software testing in which individual software modules are combined and tested as a group. It is conducted to evaluate the compliance of a system or a component with specified functional requirements. It occurs after unit testing and before validation testing.

## Integration test case 1

This test checks to see if the details of the new member entered in the registration page is stored in the database properly when the submit button is pressed, as shown in Table 5.2.

Table 5.2 Integration test case for Registration check

|  |  |
| --- | --- |
| SI No. of test case: | 1 |
| Name of test: | Registration check |
| Item / Feature being tested: | Details being registered is checked |
| Sample Input: | Enter all the details of the new member to be added and press submit button |
| Expected output: | Message ‘Member added successfully’ is to be displayed |
| Actual output: | Message ‘Member added successfully’ is to be displayed |
| Remarks: | Test succeeded |

## Integration test case 2

Admin can add new members to the database, edit and update the existing ones. He/she also can add payments, add new workout routines. The profile of the admin can also be changed by entering the updated credentials. This is tested in the Table 5.3.

Table 5.3 Integration test case for changing profile

|  |  |
| --- | --- |
| SI No. of test case: | 2 |
| Name of test: | Profile change |
| Item / Feature being  tested: | Admin profile changing check |
| Sample Input: | Enter the profile details of the admin |
| Expected output: | After entering details, press submit. Message ‘changed successfully’ is to be displayed |
| Actual output: | After entering details, press submit. Message ‘changed successfully’ is to be displayed. |
| Remarks: | Test succeeded. |

## System testing

**System testing** is testing conducted on a complete integrated system to evaluate the system's compliance with its specified [requirements](https://en.wikipedia.org/wiki/Requirements).

## System test case 1

This checks to see if the new exercise routine is properly added, as shown in Table 5.4

Table 5.4 System test case for Exercise routine

|  |  |
| --- | --- |
| SI No. of test case: | 1 |
| Name of test: | Exercise routine updating |
| Items / Features being tested: | To check whether exercise routine is being updated |
| Sample Input: | Enter the new exercise routine details |
| Expected output: | After pressing submit button, ‘new routine added’ message is to be displayed |
| Actual output: | After pressing submit button, ‘new routine added’ message is to be displayed |

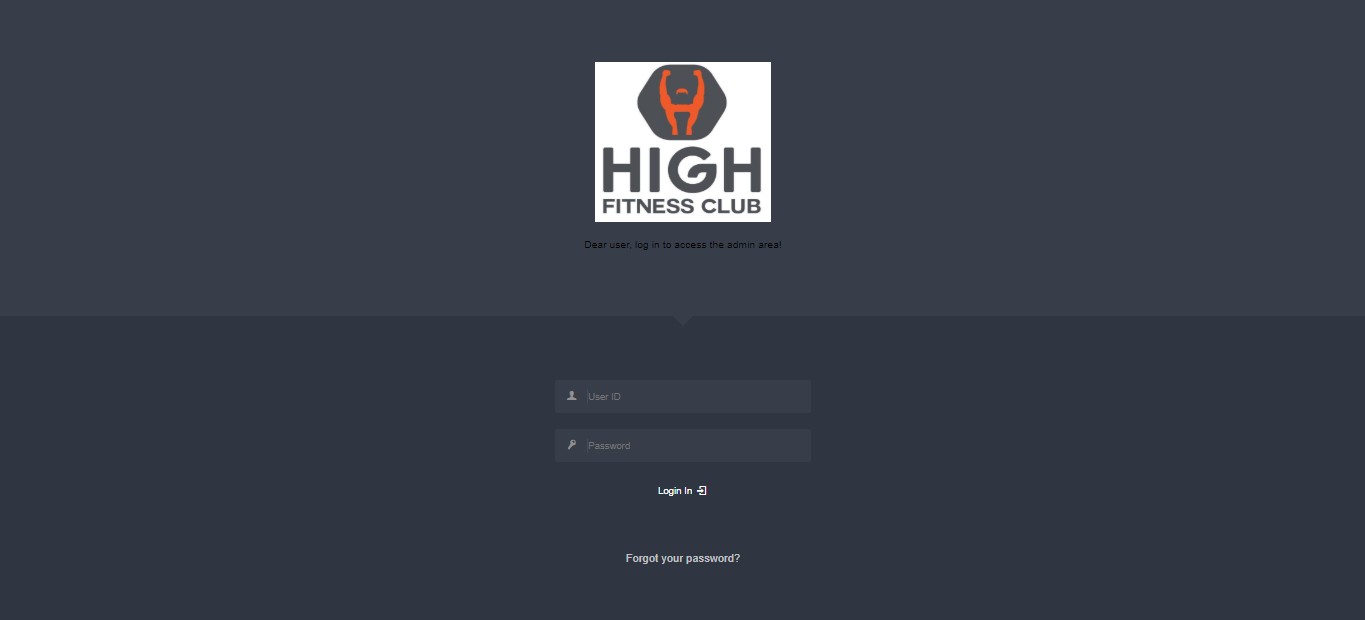
|  |  |
| --- | --- |
| Remarks: | Test succeeded. |

## Discussion of results

The outcomes of test results for a variety of user interactions are discussed in the following sections of the chapter.

## Login page

Figure 5.1 shows the Homepage of the Gym Management System displayed before user’s login.



## Dashboard

Figure 5.1 Login page

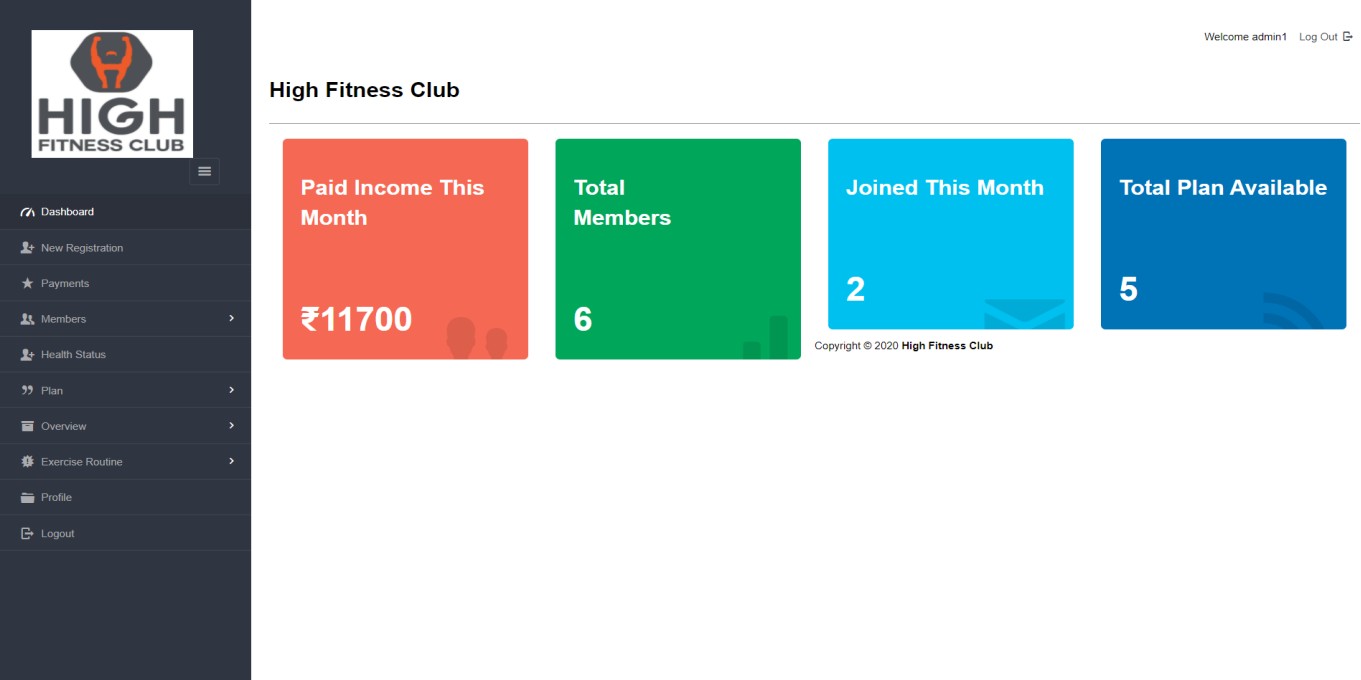
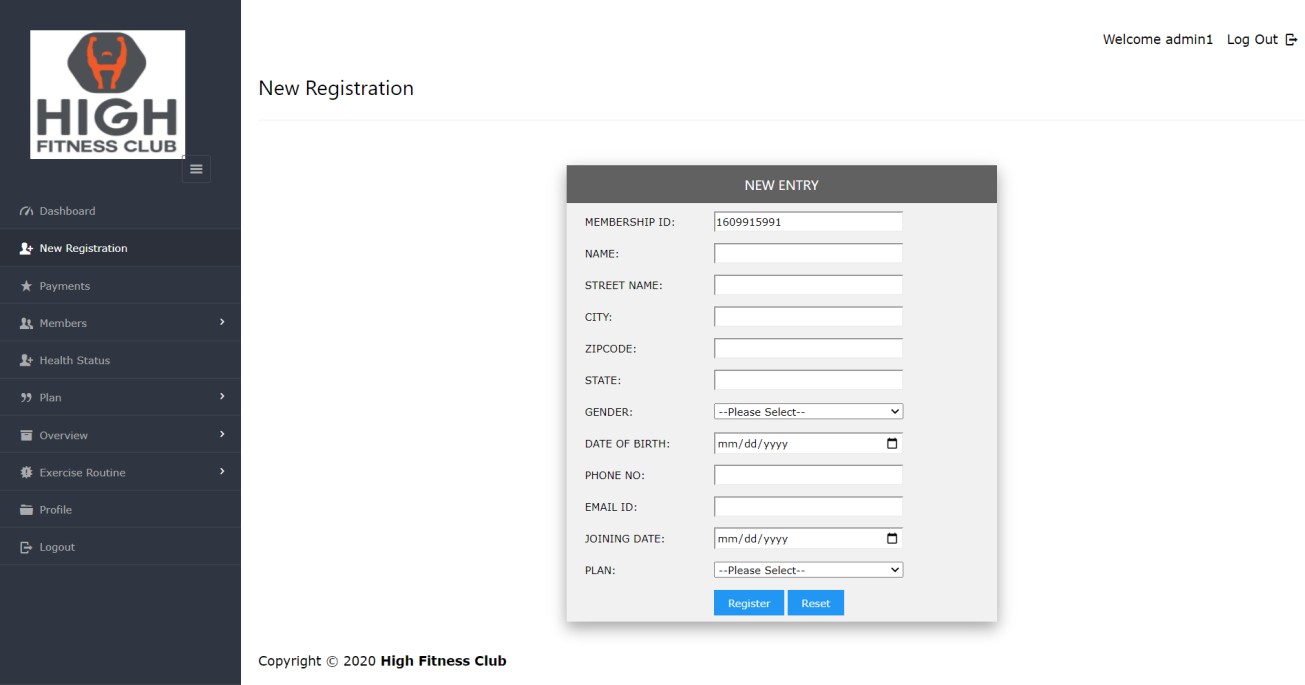
Figure 5.2 shows the Home page which is displayed after the user logs in to the system

Figure 5.2 Dashboard

## Registration page

Figure 5.3 shows the Registration page where the admin enters all the details if the user.



## Payments page

Figure 5.3 Registration page

Figure 5.4 shows the Payments page which consists of all the transactions on the members

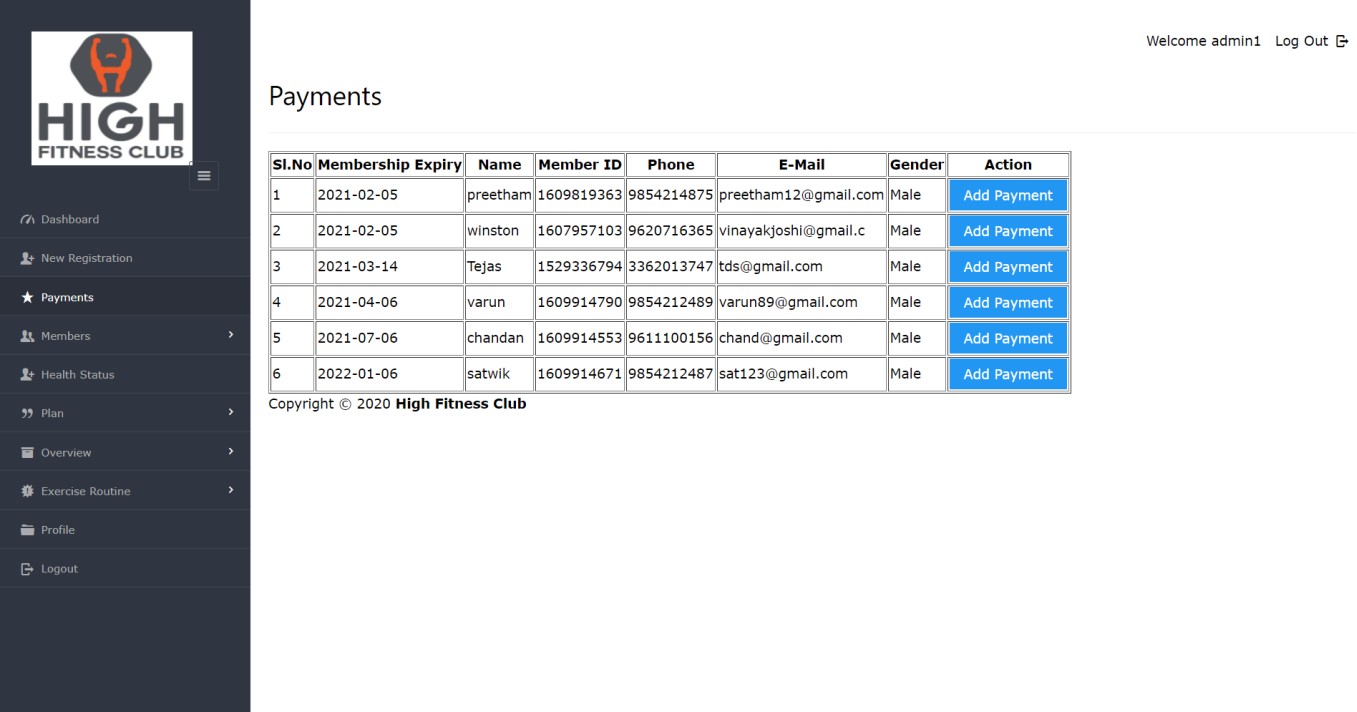


Figure 5.4 Payments page

## Health Status page

Figure 5.5 shows the Health status of the members which is maintained by the administrator

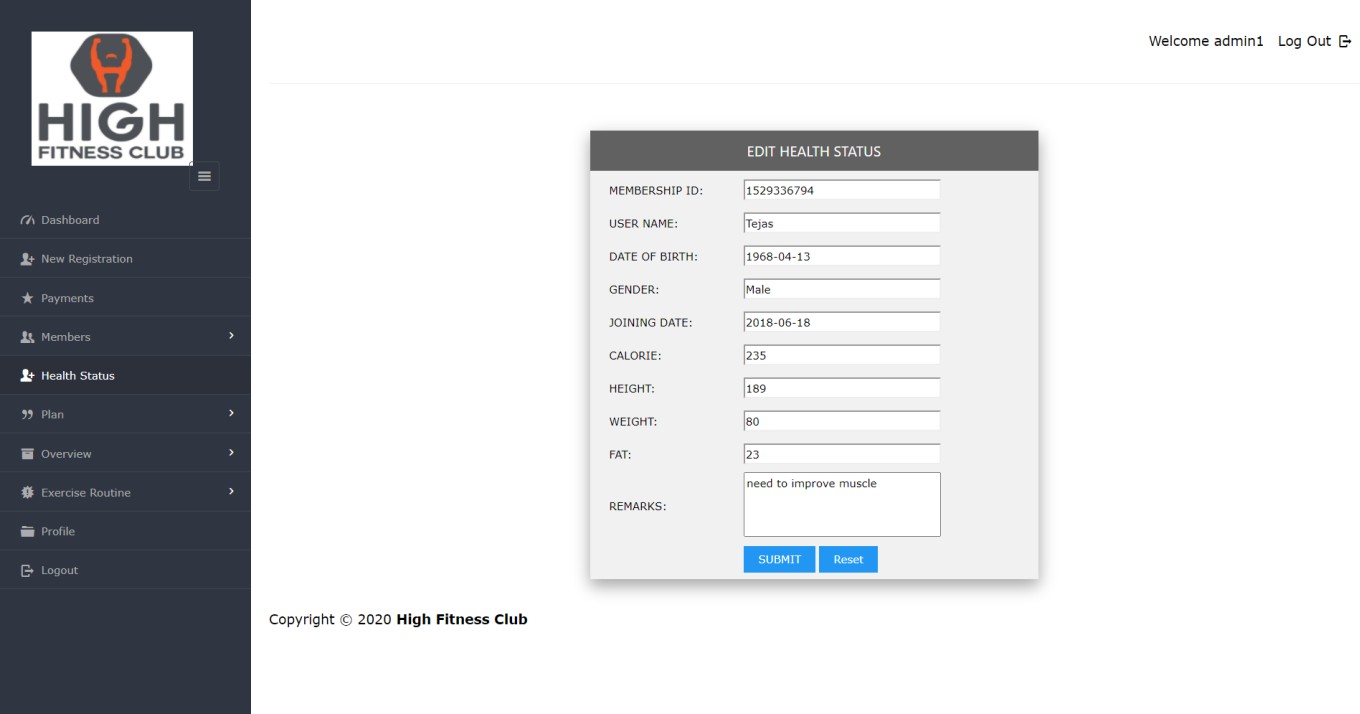


Figure 5.5 Health status page

## Workout routines page

Figure 5.6 shows the Workout routines which can be added, updated by the administrator

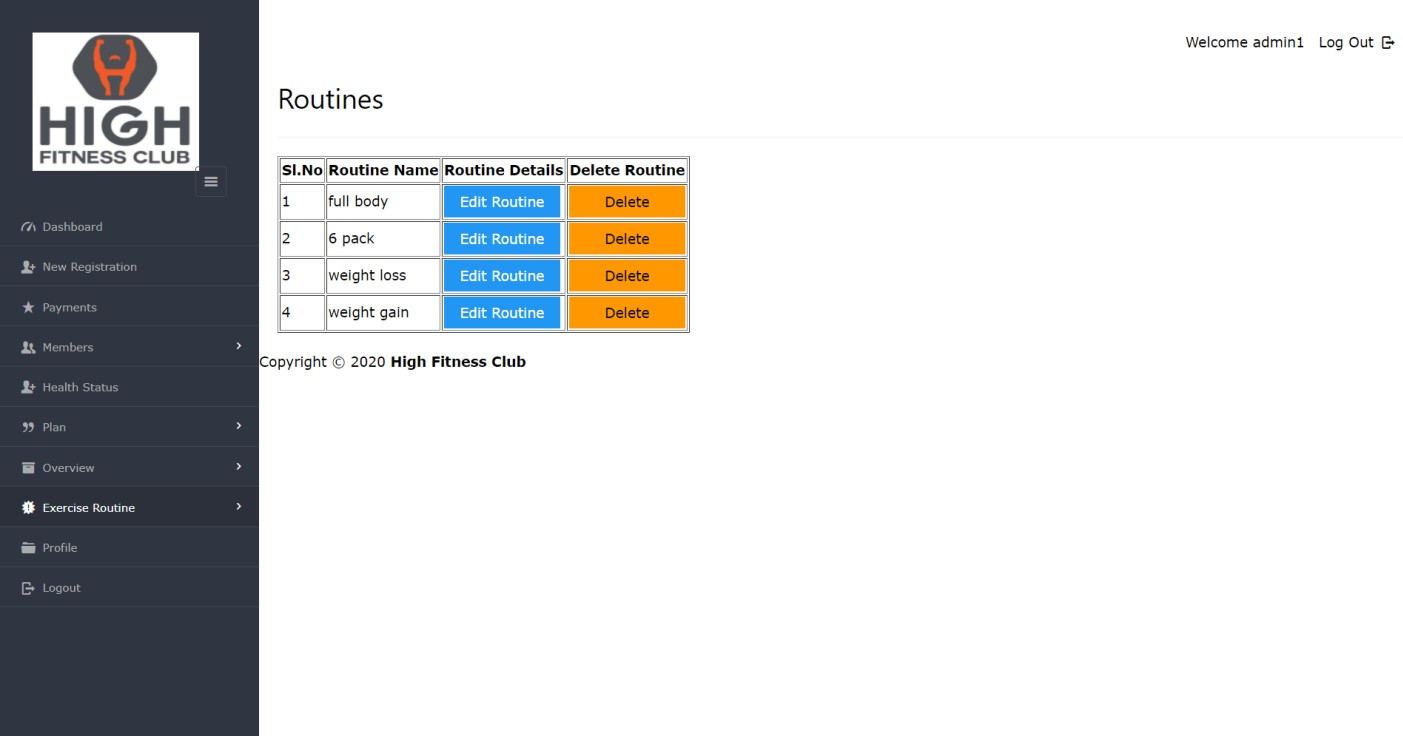


Figure 5.6 Workout routines page

## Chapter 6

**CONCLUSION AND FUTURE ENHANCEMENTS**

The system was mainly designed to reduce the manual work of registering every time a new member registers to the gym and maintain his/her health status. It also provides flexible and powerful reports regarding members, payments, workout routines, plans, health status.It organizes, maintains and updates the entire database which eases the work of the administrator.

The future enhancements that can be made to this mini project are:

* We can add an extra user login page which directs the user to the pages where he/she can manage their credentials.
* A way to register for members and subsequently login to their workout routine.
* Workout videos, inspirational quotes when the members login to their account which can be customized by them based on their likes and dislikes
* A shopping option for the required supplements, material that can be added to their cart, bought and delivered to their mentioned address.

# REFERENCES

1. Raghu Ramakrishnan and Johannes Gehrke , Database Management Systems , McGraw HILL , 3rd Edition.
2. Rameez Elmasri and Shamkant B. Navathe , Fundamentals of Database Systems , Pearson

, 7th Edition.

1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development”, 1stEdition, Pearson Education India.
2. Robin Nixon, “Learning PHP, MySQL &JavaScript with jQuery, CSS andHTML5”, 4th Edition, O’Reilly Publications, 2015.
3. Luke Welling, Laura Thomson, “PHP and MySQL Web Development”, 5th Edition, Pearson Education, 2016.
4. Nicholas C Zakas, “Professional JavaScript for Web Developers”, 3rd Edition, Wrox/Wiley India, 2012.
5. [www.php.net](http://www.php.net/)
6. [www.javascript.com](http://www.javascript.com/)