ABSTRACT

The project Sports Club Management System creates a software that stores and manages all the data needed to describe the personal data of the members and their framework within an organization. It includes definition of various levels of hierarchy in an organization, the price structure pertaining to every element in this hierarchy, the description of every member functioning in the club and the overall sports club database which integrates all the elements mentioned above.

It has a database administration that has access to the entire database, in regards with viewing and update of information. The exclusive right is implemented using authorized access. Also viewing all data and editing of personal data can be done by any admin, this also using authorized access.

ACKNOWLEDGEMENTS

We express our humble pranamams to his holiness **Jagadguru Sri Sri Sri Shivarathri Deshikendra Mahaswamiji** who has showered their blessings on us for framing our career successfully.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide **Mr. Rohithaksha K,** Assistant Professor, Department of Computer Science and Engineering who has helped us in every aspect of our project work.

We are also indebted to **Dr Naveen N C,** Head of Department of Computer Science and Engineering for the facilities and support extended towards us.

We express our sincere thanks to our beloved principal**, Dr. Mrityunjaya V Latte** for having supported us in our academic endeavors.

And last but not the least, we would be very pleased to express our heart full thanks to all the teaching and non-teaching staff of CSE department and our friends who have rendered their help, motivation and support.

SHREYANSH GUPTA

Table of Contents

List Of Figures

List Of Tables

Chapter 1

Preamble

1.1 Introduction

The Sports Club Management System project deals with registering new members, plans, payments, routine and managing the members for the club. The project has complete access for the crud operations that are to create, read, update and delete the database entries. At first you need to login as this system is totally controlled by the admin/owner and then register the members for the club and check their health status and view the total income per month. Now you can assign different routine to different members and also check the health status which can be viewed and edited too and finally check the payments according to the plan they have chosen.

It has a database administration that has access to the entire database, in regards with viewing , updating and deleting the information.

1.1.1 History of Database Management System

Following the technology progress in the areas of processors, computer memory, computer storage, and computer networks, the sizes, capabilities, and performance of databases and their respective DBMSs have grown in orders of magnitude. The development of database technology can be divided into three eras based on data model or structure: navigational, SQL/relational, and post-relational. The two main early navigational data models were the hierarchical model, epitomized by IBM's IMS system, and the CODASYL model (network model), implemented in a number of products such as IDMS.

The relational model employs sets of ledger-style tables, each used for a different type of entity. Only in the mid-1980s did computing hardware become powerful enough to allow the wide deployment of relational systems (DBMSs plus applications). By the early 1990s, however, relational systems dominated in all large-scale data processing applications, and as of 2015 they remain dominant: IBM DB2, Oracle, MySQL, and Microsoft SQL Server are the top DBMS. The dominant database language, standardized SQL for the relational model, has influenced database languages for other data models.

1.1.2 MySQL

MySQL is an open-source relational database Management System (RDBMS). MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including Linux, macOS, Microsoft Windows, NetBSD. MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features that are available in MySQL are a broad subset of ANSI SQL 99,as well as extensions, Cross-platform support, Stored procedures, using a procedural language that closely adheres to SQL/PSM, Triggers, Cursors, Updatable views, Online DDL when using the InnoDB Storage Engine. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio and the JDBC driver for Java. In addition, an ODBC interface called MySQL Connector/ODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion.

1.1.3 PHP

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page,but it now stands for the recursive initialism PHP: Hypertext Preprocessor.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content Management Systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML

code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

The standard PHP interpreter, powered by the send Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

As of September 2019, over 60% of sites on the web using PHP are still on discontinued/"EOLed" version 5.6 or older;versions prior to 7.1 are no longer officially supported by The PHP Development Team, but security support is provided by third parties, such as Debian.

1.1.4 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a XAMPP stack can be installed quickly and simply on an operating system by a developer, with the advantage a number of common add-in applications such as Wordpress and Joomla! can also be installed with similar ease using Bitnami .

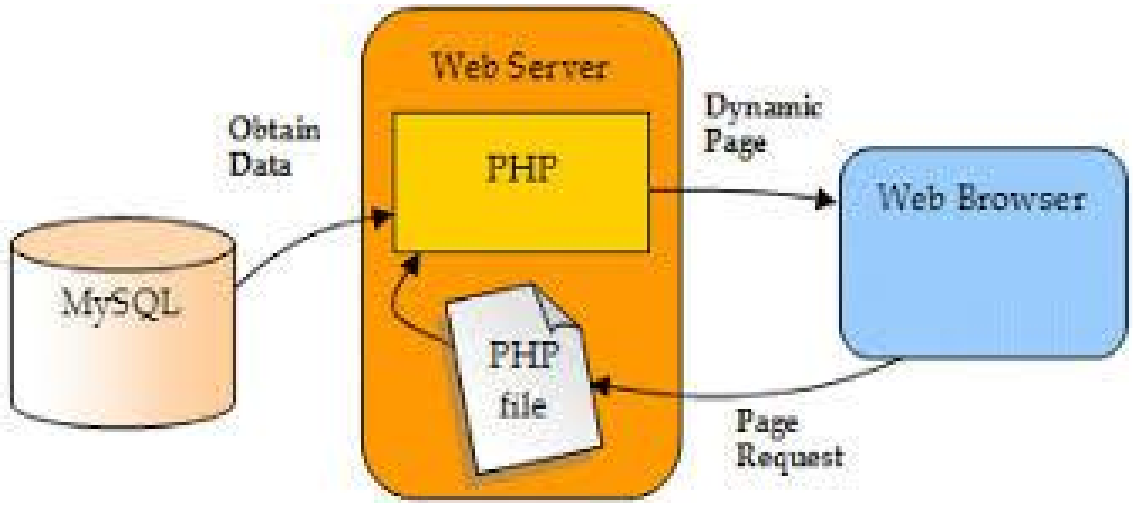


Figure 1.1: **Web Server**

1.1.5 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. To overcome these anomalies we need to normalize the data. There are 4 basic types of normalizations. They are:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce & Codd normal form (BCNF)

A table is said to be in 2NF if the two conditions stated are satisfied. The table is in First normal form and all the non-prime attribute are dependent on the proper subset of any candidate key of table. The attribute that is not part of any candidate key are known as non-prime attribute.

A table design is said to be in 3NF if the table is in 2NF and Transitive functional dependency of non-prime attribute on any super key are removed.

A table design is said to be in BCNF if there is only one super key

1.2 OBJECTIVES

The Sports Club Management software is very user friendly and appealing. The Human objective of the system is to maintainand retrieve information about the members and the sports they will play on which day of the week and at what time in the sports club. The system is fairly simple in design and implementation.

The main objectives of this system can be summarized as follows:

- Design of a GUI portal for managing theSports Club for the main admin/owner.
- Insertion of member's data, plans and managing of payment and health status.
- Monthly Payment for monthly plan and yearly payment of yearly plan.
- Computerized manipulation and management of member data and id.

Easy management of databases of various sections covering key aspects.

1.3 Organization of the Report

Chapter 1 provides the information about the basics of MySQL, ,. In Chapter 2, all the requirement specifications are described. Chapter 3 gives the idea of the project and its actual implementation. Chapter 4 discusses about the results of the program. Chapter 5 concludes by giving the direction for future enhancement.

1.4 Summary

The chapter discussed before is an overview about the PHP Application and MySQL DBMS and its history. It even includes the PHP Application block diagram. The scope of study and objectives of the project are mentioned clearly. The organization of the report is been pictured to increase the readability. Further, coming up chapters depicts the use of various queries to implement various changes like insert, update, delete and also triggers to perform various functions.

Chapter 2

REQUIREMENTS

A high-level requirements specification is required. The purpose of the requirements analysis is to identify requirements for the proposed system. The emphasis is on the discovery of user requirements. Each requirement (or problem) must be defined and documented in the requirements catalogue. Each requirement is recorded in the requirements catalogue on a requirements catalogue entry form. A copy of the form is in the appendix section of the standards manual. The form should be completed as follows:

• Project/System - the proposed system name or an abbreviation of it. • Analyst - your name as the analyst on this project.

• Date - the date the entry was made in the catalogue, whether it is a new entry or an amended entry, i.e., another version of an existing entry.

• Version - a version number is assigned to a requirement. The initial version of a requirement is number one. However, the requirement may need to be updated in the course of the development of the project, so then the requirements catalogue entry will be replaced with the updated version of the

• requirement and the updated version number will reflect this change.

• Status - the status of the requirement will be either ongoing or complete. When the status is ongoing the status box will be empty, and when the status is complete the status box will contain a tick (().

• Page - page numbering will be maintained within the catalogue.

• Source/Origin - is the originator of the requirement; the person with the responsibility for negotiation about the requirement.

• Requirement Number - a unique requirement number is assigned to each requirement. The requirement number is an incremental number starting with one.

• Priority - a priority is assigned to the requirement. The priority given is agreed between the originator and the analyst. There are three priority levels: high (H), medium (M) and low (L).

High priority is assigned when the requirement is mandatory. Medium priority is assigned when the requirement is desirable.

• Low priority is assigned when the requirement is optional.

• Functional Requirements and Non-Functional Requirements - see next page. Human Resource Management and Performance Evaluation System.

• Related Documents - reference to any related documents, eg: user documentation, data flow diagrams.

• Proposed Solution - any possible solution or general comments.

2.1 SOFTWARE SPECIFICATION

- Operating System: Windows 2000/XP/Vista
- Front End : HTML & CSS
- Database : MySQL
- Server: XAMPP
- Design Tool: PHP

2.2 HARDWARE SPECIFICATION

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or grater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical.

2.3 USER CHARACTERISTICS

Every user:

- Should be comfortable with basic working of the computer
- Must have been knowledge of English
- Must carry a login ID and password used for authentication
- The GUI is restricted to English
- Login ID and password used for identification of administrator. There is no facility for a guest login.

Chapter 3

System Design and Implementation

3.1 Introduction

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

This Project is implemented using XAMPP, which is proven to be a very efficient tool in the field of PHP programming. It is done under Windows10 platform. . PHP programming language is used to implement the entire code. Interface to the program is provided with the help of MySQL.

The Sports Club management project deals the adding new Sports Clubplans and managing the members for the Sports Club. The project has complete access for the crud operations that are to create, read, update and delete the database entries. . At first you need to login as this system is totally controlled by the admin/owner andthen registerthe members for the club andcheck their health status and view the total income per month . Now you can assign different routine to different members and also check the health status which can be viewed and edited too and finally check the payments according to the plan they have chosen.It has a database administration that has access to the entire database, in regards with viewing and deletion of information.

Existing System:

Here the existing system is nothing but a manual system in which the admin has to fill the member's Sports Club details in an excel sheet and send it to their supervisor then the supervisor has to merge all the member information details and arrange them in to a single sheet. Maintaining a clean record of all the members is a tedious job in this process.

Drawback:

- Paperback records are hard to maintain.
- Searching for a member's information can be gruesome task.
- Doesn't provide Security.
- Difficulty in updating the records.
- More manual hours is needed to maintain the records.
- Updating the member on deadline for payment is difficult.

One way to overcome all these difficulties is so store all the information in database. The computerization helps mitigate a lot of drawback and streamlines the process.

Proposed System:

Performance Management System is to replace the existing manual system with a software solution. It allows all the employees in different sections of the club to work together and manage a single record.
Different areas of the Sports Club can be managed in different tabs by different people.

Other Computerized System advantages:

- Faster processing
- Centralized database helps in avoiding conflicts.
- Easy to use GUI that does not require specific training.

3.2 ER Diagram.

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

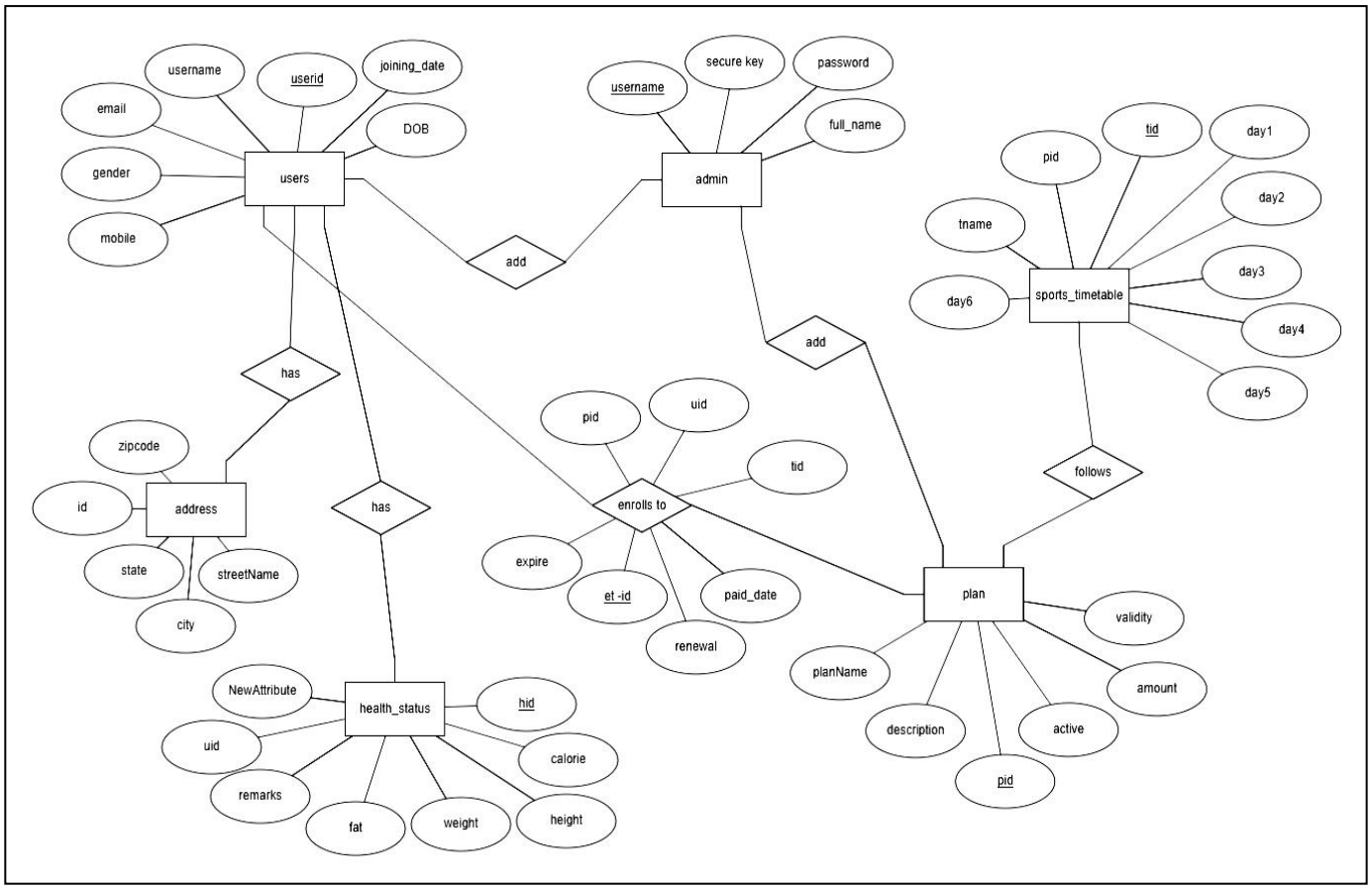


Figure 3.2: **Entity Relationship Diagram**

Figure 3.1 refers to the Entity Relationship diagram utilized in our project

3.3 Schema Diagram.

The database schema of a database system is its structure described in a formal language supported by the database Management System (DBMS). The term "schema" refers to the organization 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 database schema is a set of formulas (sentences) called integrity constraints imposed on a database.

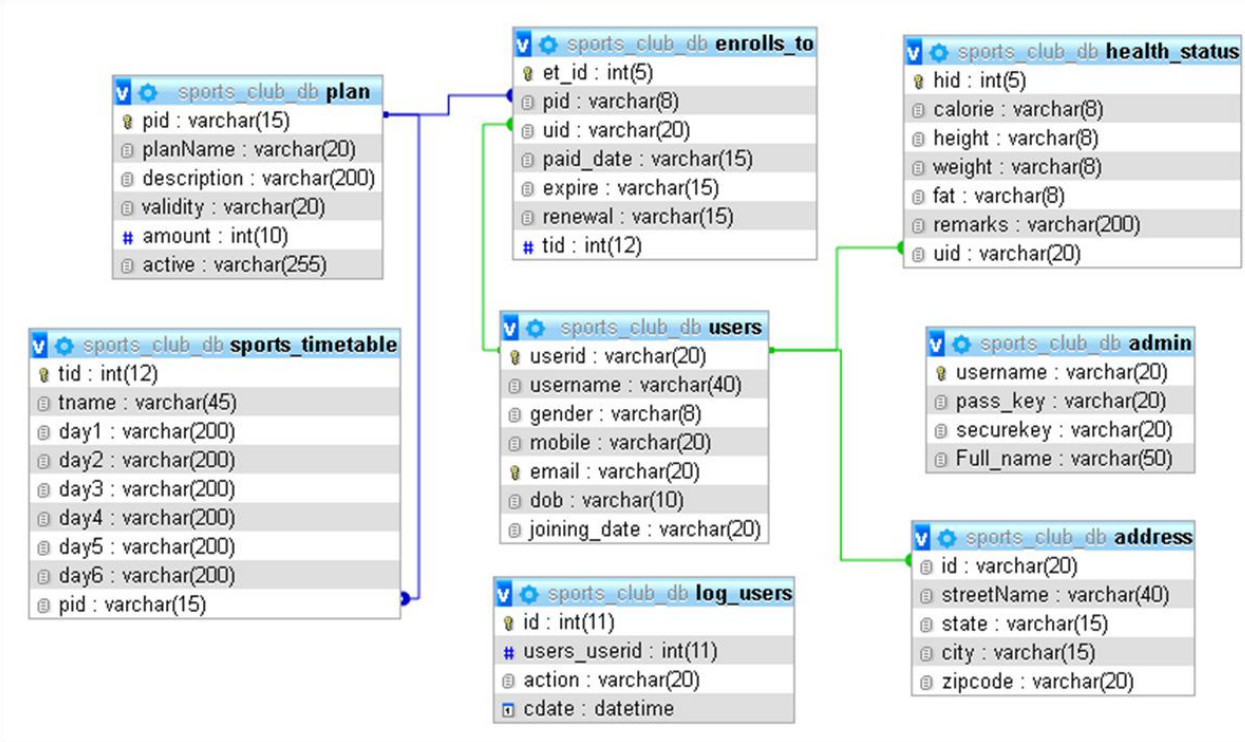


Figure 3.3: Schema Diagram

3.4 Data Tables

Table 3.4.1 : ADDRESS TABLE

FIELD NAME	DATA TYPE
id (Primary key)	VARCHAR(20)
streetName	VARCHAR(40)
state	VARCHAR(15)
city	VARCHAR(15)
zipcode	VARCHAR(20)

Table 3.4.2 : ADMIN TABLE

FIELD NAME	DATA TYPE
username(Primary key)	VARCHAR(20)
pass_key	VARCHAR(20)
securekey	VARCHAR(20)
Full_name	VARCHAR(50)

Table 3.4.3 : ENROLLS_TO TABLE

FIELD NAME	DATA TYPE
et_id (Primary key)	INT(5)
pid	VARCHAR(8)
uid	VARCHAR(20)
paid_date	VARCHAR(15)
expire	VARCHAR(15)
renewal	VARCHAR(15)

Table 3.4.4 : HEALTH_STATUS TABLE

FIELD NAME	DATA TYPE
hid(Primary key)	INT(5)
calories	VARCHAR(8)
height	VARCHAR(8)
weight	VARCHAR(8)
fat	VARCHAR(8)
remarks	VARCHAR(200)
uid	VARCHAR(20)

Table3.4.5 : LOG_USERS TABLE

FIELD NAME	DATA TYPE
id (Primary key)	INT(11)
users_userid	INT(11)
action	VARCHAR(20)
cdate	DATETIME

Table 3.4.6 : PLAN TABLE

FIELD NAME	DATA TYPE
pid (Primary key)	VARCHAR(15)
planName	VARCHAR(20)
description	VARCHAR(200)
validity	VARCHAR(20)
amount	INT(10)
active	VARCHAR(255)

Table 3.4.7 : SPORTS_TIMETABLE TABLE

FIELD NAME	DATA TYPE
tid (Primary key)	INT(12)

tname	VARCHAR(45)
day1	VARCHAR(200)
day2	VARCHAR(200)
day3	VARCHAR(200)
day4	VARCHAR(200)
day5	VARCHAR(200)
day6	VARCHAR(200)
pid	VARCHAR(15)

Table 3.4.8 :USERS TABLE

FIELD NAME	DATA TYPE
userid (Primary key)	VARCHAR(20)
username	VARCHAR(40)
gender	VARCHAR(8)
mobile	VARCHAR(20)
email	VARCHAR(20)
dob	VARCHAR(10)
joining_date	VARCHAR(20)

3.5 Pseudo codes for Sports Club Management

3.5.1 Algorithm for login

Step 1: BEGIN

Step 2: Enter username and password

Step 3: Verify the credentials entered

Step 4: If Credentials match, then proceed to the next

Else show login failed

Step 5: End if

Step 6: END

3.5.2 Algorithm to Add member

Step 1: BEGIN

Step 2: Add the member details in the new registration Column

Step 3:END

3.5.3Algorithm to insert sports plan

Step 1: BEGIN

Step 2: Go to plan Column

Step 3: Add the new plan details and the corresponding plan id.

Step 4: Select the plan id.

Step 5:END

3.5.4 Algorithm to insert Sports Routine

Step 1: BEGIN

Step 2: Go to Club Routine Column

Step 3: Add , remove or edit the routine you want the members to follow.

Step 4:END

3.5.5 Algorithm for Payment

Step 1: BEGIN

Step 2: Go to Payments Column and select Add payment.

Step 3: Add the corresponding sports plan the member registered to.

Step 4:click the add payment and the status will be updated

Step 5:END

3.6 SQL tables implemented in database

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description of the query is also provided.

CREATE STATEMENTS

Description: This query is used to create a table called address Table which will store the address of all the Sports Clubmmbers.

Query:
```
CREATE TABLE `address`  (
  `id` varchar(20),
  `streetName` varchar(40),
  `state` varchar(15),
  `city` varchar(15),
  `zipcode` varchar(20),
  INDEX `userID`(`id`) USING BTREE,
```

```
  CONSTRAINT `userID` FOREIGN KEY (`id`) REFERENCES `users` (`userid`) ON
DELETE CASCADE ON UPDATE NO ACTION
);
```

Description: This query is used to create a table called admin Table which will store the credentials of all the Sports Clubadmins.

Query:

```
CREATE TABLE `admin`  (
  `username` varchar(20),
  `pass_key` varchar(20),
  `securekey` varchar(20),
  `Full_name` varchar(50),
  PRIMARY KEY (`username`)
);
```

Description: This query is used to create a table enrolls_to Table which stores the enrollment is and details of themembers

Query :

```
CREATE TABLE `enrolls_to`  (
  `et_id` int(5) NOT NULL AUTO_INCREMENT,
  `pid` varchar(8) ,
  `uid` varchar(20),
  `paid_date` varchar(15),
  `expire` varchar(15),
  `renewal` varchar(15),
  PRIMARY KEY (`et_id`),
  INDEX `user_ID`(`uid`) USING BTREE,
  INDEX `plan_ID_idx`(`pid`) USING BTREE,
  CONSTRAINT `plan_ID` FOREIGN KEY (`pid`) REFERENCES `plan` (`pid`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `user_ID` FOREIGN KEY (`uid`) REFERENCES `users` (`userid`) ON
DELETE CASCADE ON UPDATE NO ACTION
);
```

Description: This query is used to create a table called health_status Table which stores the health status of the members.

Query:
```
CREATE TABLE `health_status` (
  `hid` int(5) NOT NULL AUTO_INCREMENT,
  `calorie` varchar(8),
  `height` varchar(8),
  `weight` varchar(8),
  `fat` varchar(8),
  `remarks` varchar(200),
  `uid` varchar(20),
  PRIMARY KEY (`hid`),
  INDEX `userID_idx`(`uid`) USING BTREE,
  CONSTRAINT `uID` FOREIGN KEY (`uid`) REFERENCES `users` (`userid`) ON
DELETE CASCADE ON UPDATE NO ACTION
);
```

Description: This query is used to create a table called log_users Table for the trigger implementation.

Query:
```
CREATE TABLE `log_users` (
  `id` int(11) NOT NULL,
  `users_userid` int(11) NOT NULL,
  `action` varchar(20) NOT NULL,
  `cdate` datetime NOT NULL
)
```

Description: This query is used to create a table called plan Table which consists of different plans available in the sports club.

Query:
```
CREATE TABLE `plan` (
  `pid` varchar(8),
  `planName` varchar(20),
```

```
  `description` varchar(200),
  `validity` varchar(20),
  `amount` int(10) NOT NULL,
  `active` varchar(255),
  PRIMARY KEY (`pid`),
  INDEX `pid`(`pid`) USING BTREE
);
```

Description: This query is used to create a table called sports_timetable Table which stores the routine of the various sports that are being played in a day.

Query:

```
CREATE TABLE `sports_timetable`  (
  `tid` int(12) NOT NULL AUTO_INCREMENT,
  `tname` varchar(45),
  `day1` varchar(200),
  `day2` varchar(200),
  `day3` varchar(200),
  `day4` varchar(200),
  `day5` varchar(200),
  `day6` varchar(200),
  `pid` varchar(8),
  PRIMARY KEY (`tid`),
  CONSTRAINT `pID` FOREIGN KEY (`pid`) REFERENCES `plan` (`pid`) ON DELETE CASCADE ON UPDATE NO ACTION);
```

Description: This query is used to create a table called users Table which stores the details of the members which they submitted at the time of registration sports club

Query:

```
CREATE TABLE `users`  (
  `userid` varchar(20),
  `username` varchar(40),
  `gender` varchar(8),
  `mobile` varchar(20),
  `email` varchar(20),
  `dob` varchar(10),
```

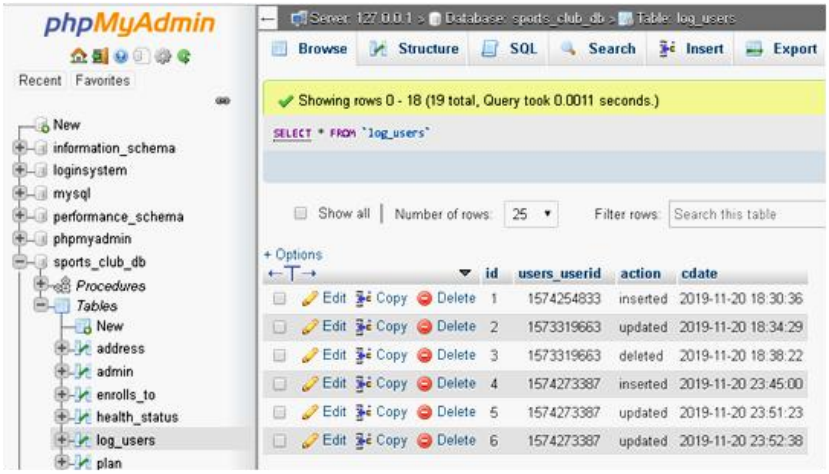
```
  `joining_date` varchar(10) ,
    `tid` int(12)),
  PRIMARY KEY (`userid`) ,
 UNIQUE INDEX `email`(`email`) USING BTREE,
  INDEX `userid`(`userid`) USING BTREE,
   CONSTRAINT `tid` FOREIGN KEY (`tid`) REFERENCES `sports_list` (`tid`) ON DELETE CASCADE ON UPDATE NO ACTION
);
```

3.7 SQL stored procedures and triggers used

TRIGGERS

Trigger:

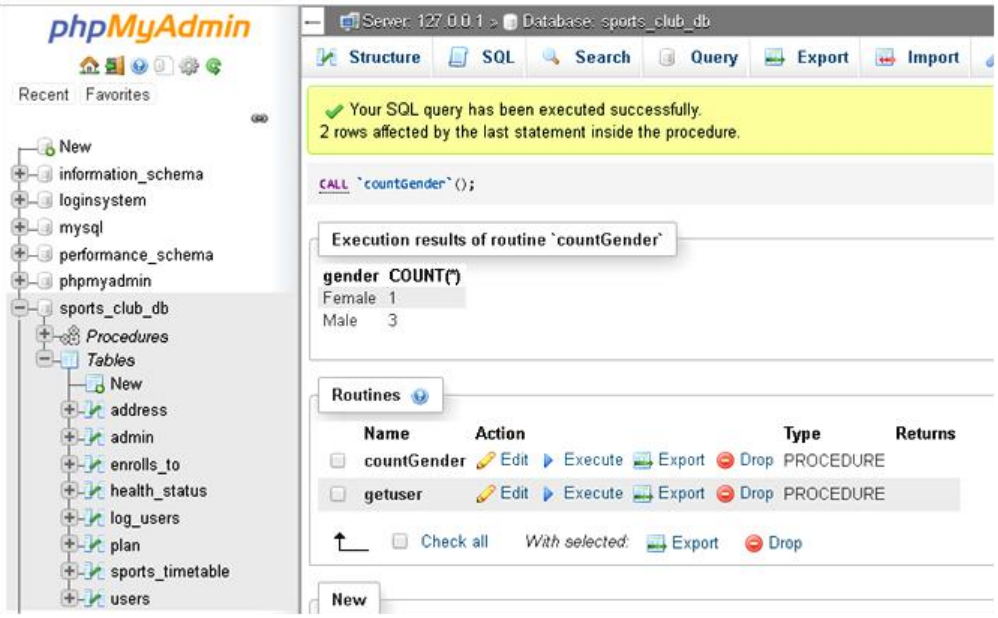
```
CREATE TRIGGER `deletelog` BEFORE DELETE ON `users`
 FOR EACH ROW insert into log_users values(null,old.userid,'deleted',now())

CREATE TRIGGER `insertlog` AFTER INSERT ON `users`
 FOR EACH ROW INSERT INTO log_users VALUES(null,NEW.userid,'inserted',now())

CREATE TRIGGER `updatelog` AFTER UPDATE ON `users`
 FOR EACH ROW insert INTO log_users values(null,new.userid,'updated',now())
```



Description: This trigger is used to display the action performed in the users table i.e. insertion,deletion or update and reflects these changes in log_users table.

STORED PROCEDURES

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `countGender`()
SELECT gender , COUNT(*) from users group by gender$$
DELIMITER ;
```



Description: This procedure will count and display the number of males and females in the users table.

Chapter 4

RESULTS AND DISCUSSIONS

The project is compiled and executed on chrome. Some screen shots are present here to show the working of the application.

ScreenShots:

The below figure refers to the initial bootup page of the application.

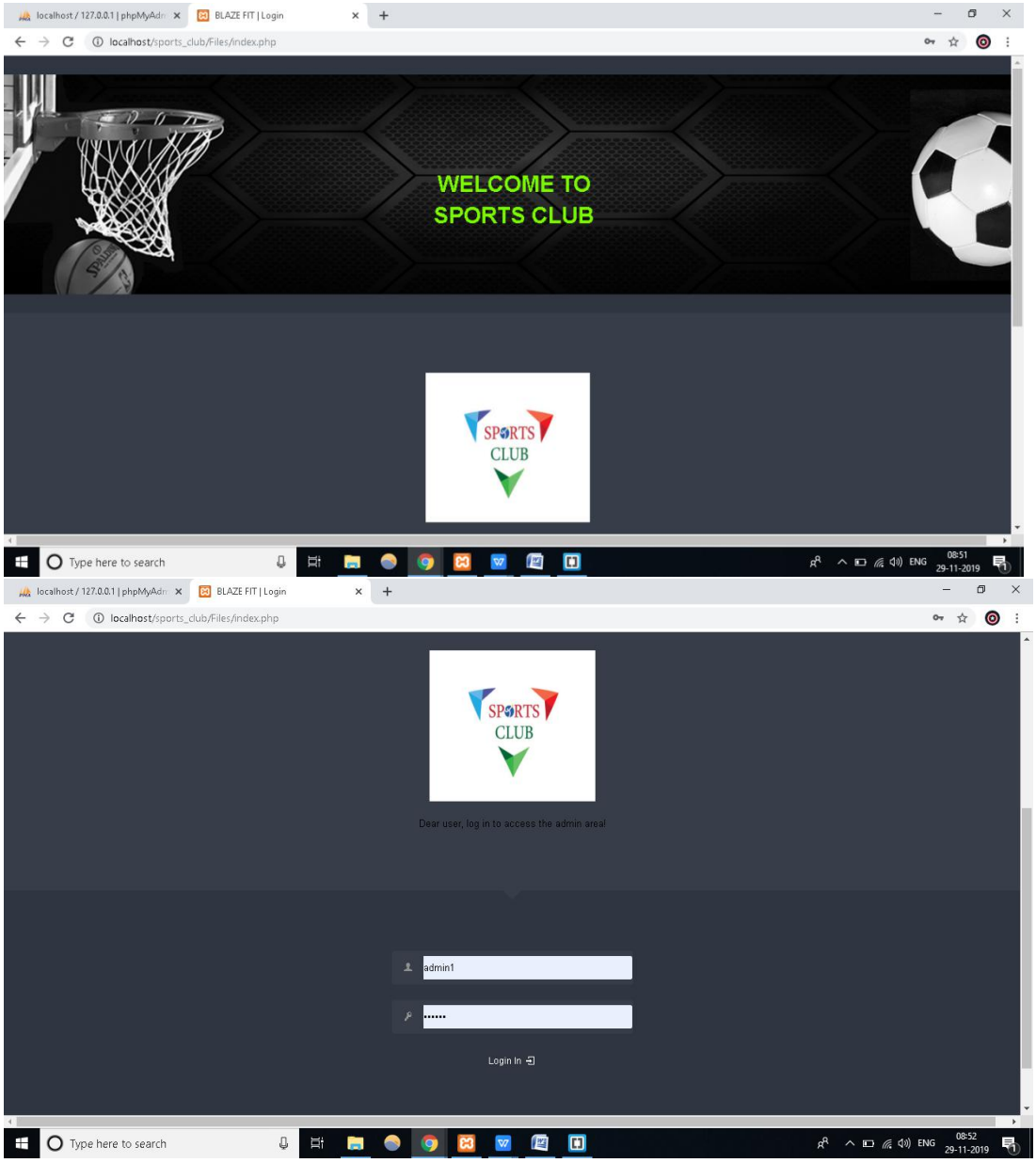


Figure 4.1 Admin Login

The below figure refers to Dashboard page if the login username and password is correct and you get logged in the system.

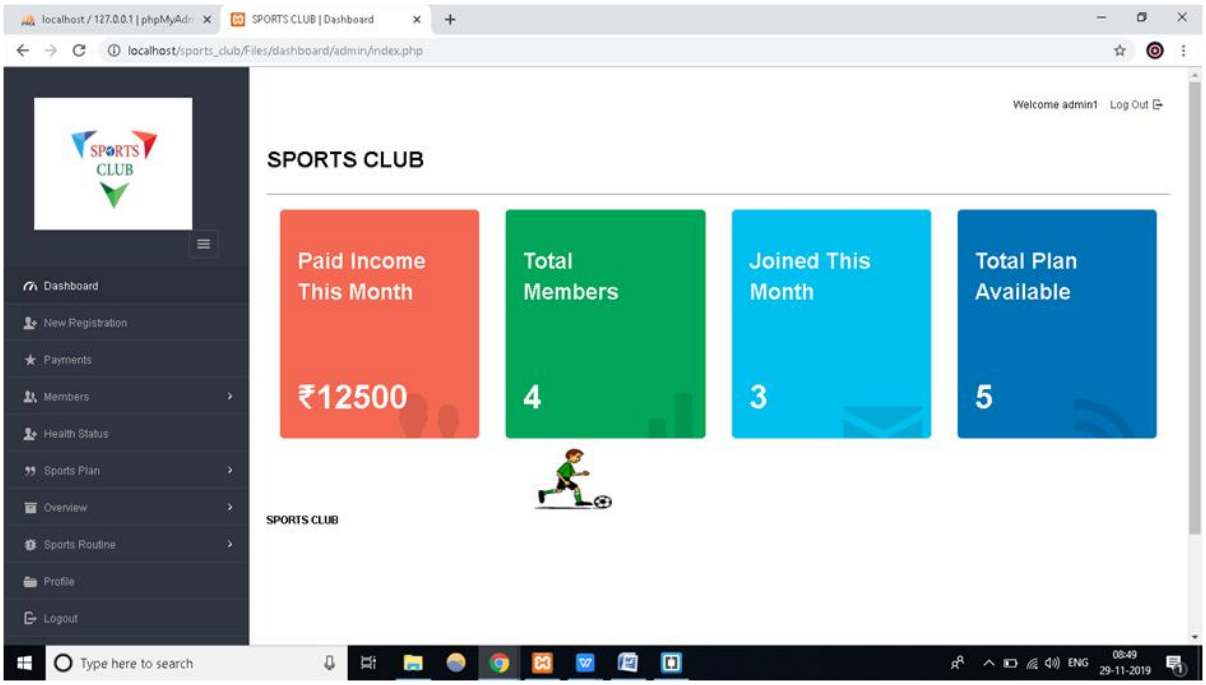


Figure 4.2 Dashboard

The below figure refers to the member registration

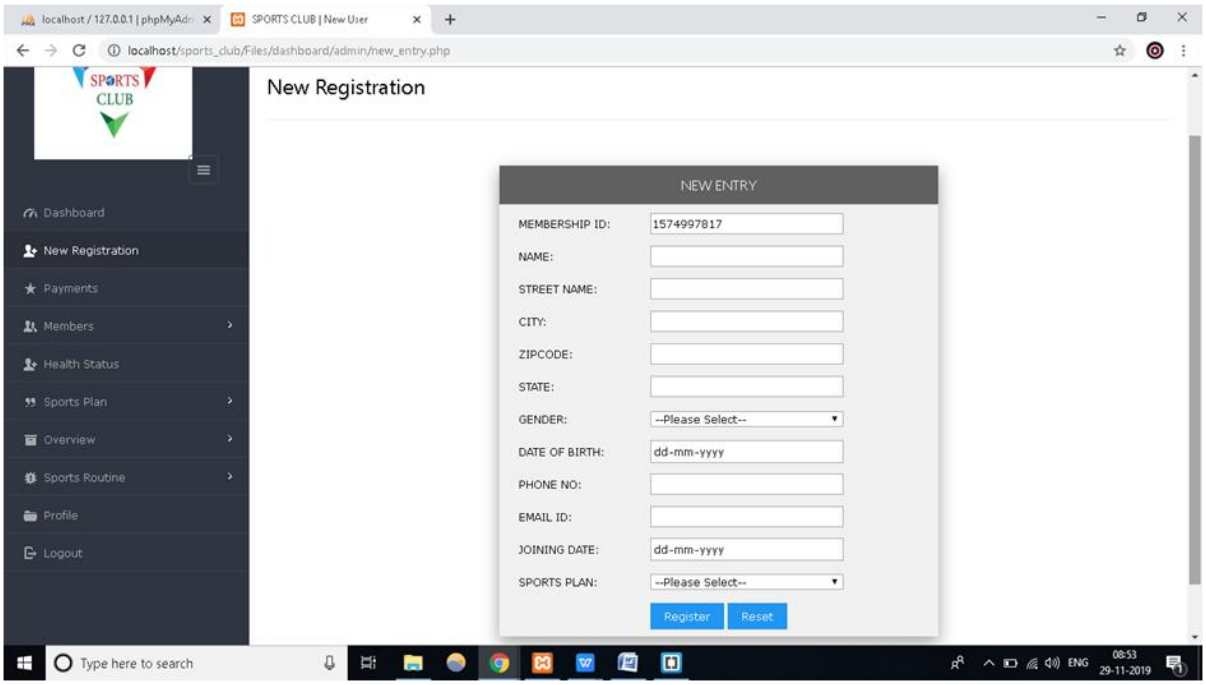
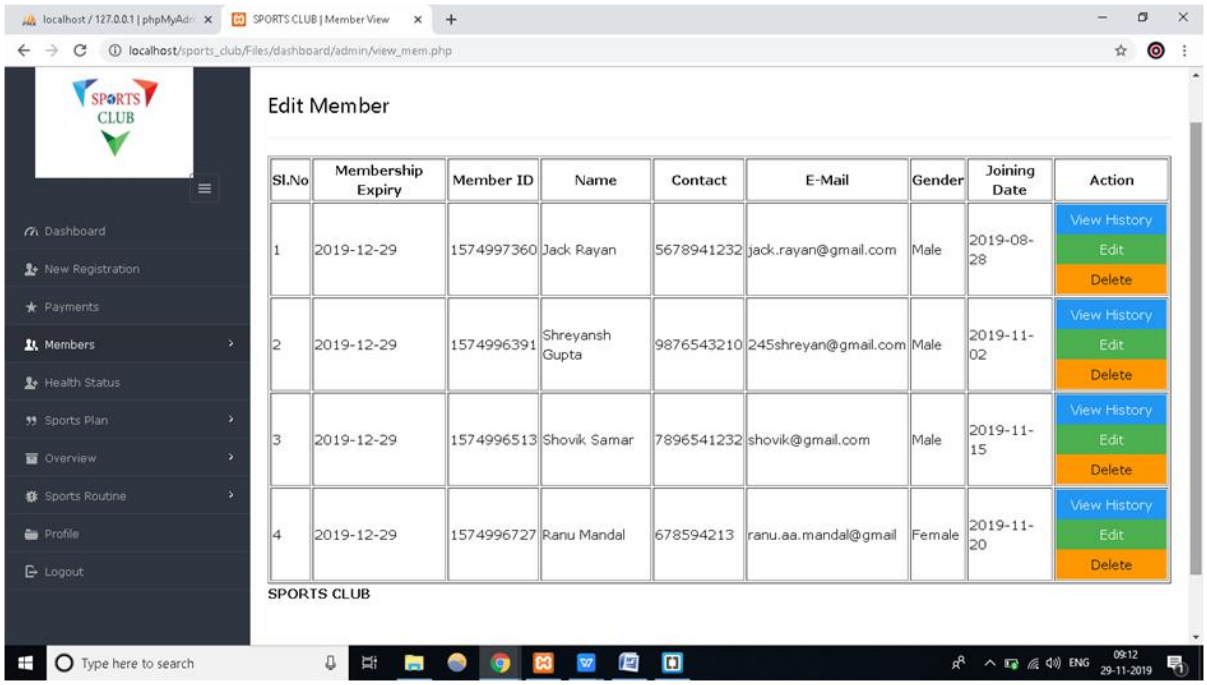


Figure 4.3 Member Registration

The below figure refers to the Member viewing



Sl.No	Membership Expiry	Member ID	Name	Contact	E-Mail	Gender	Joining Date	Action
1	2019-12-29	1574997360	Jack Rayan	5678941232	jack.rayan@gmail.com	Male	2019-08-28	View History / Edit / Delete
2	2019-12-29	1574996391	Shreyansh Gupta	9876543210	245shreyan@gmail.com	Male	2019-11-02	View History / Edit / Delete
3	2019-12-29	1574996513	Shovik Samar	7896541232	shovik@gmail.com	Male	2019-11-15	View History / Edit / Delete
4	2019-12-29	1574996727	Ranu Mandal	678594213	ranu.aa.mandal@gmail	Female	2019-11-20	View History / Edit / Delete

Figure 4.4 Member Viewing and Editing

The figure below refers to the Payments of membership

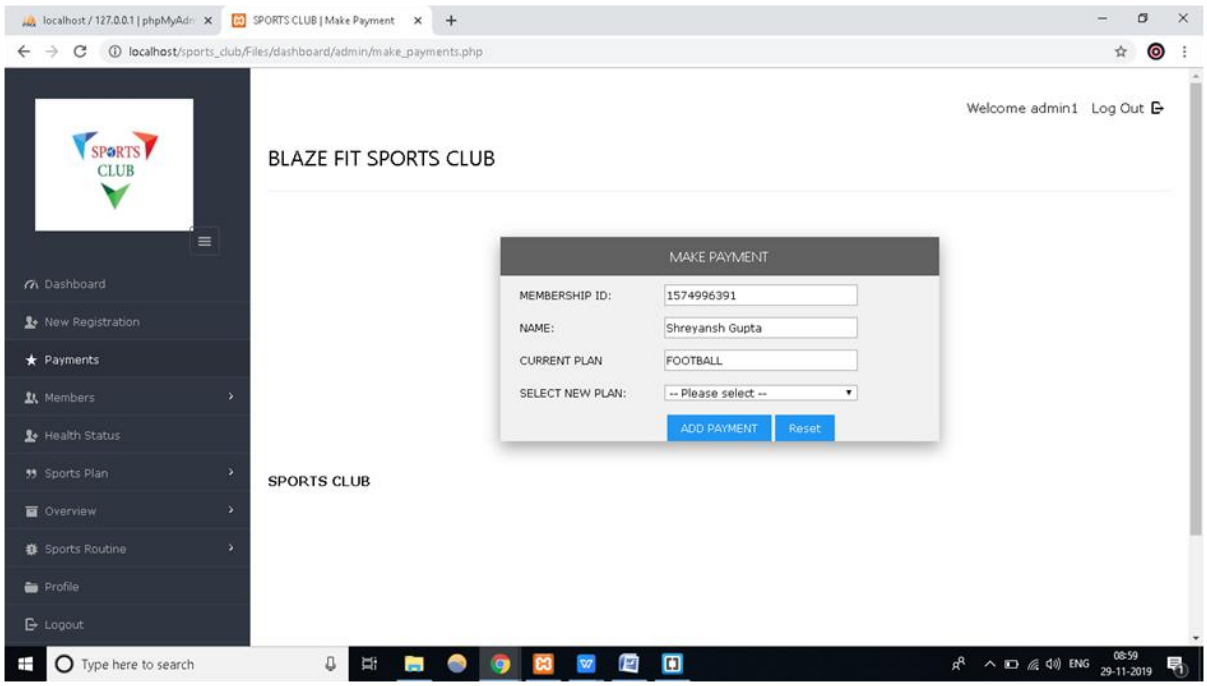


Figure 4.5 Payments

The below figure refers to the Health Status entry for the member

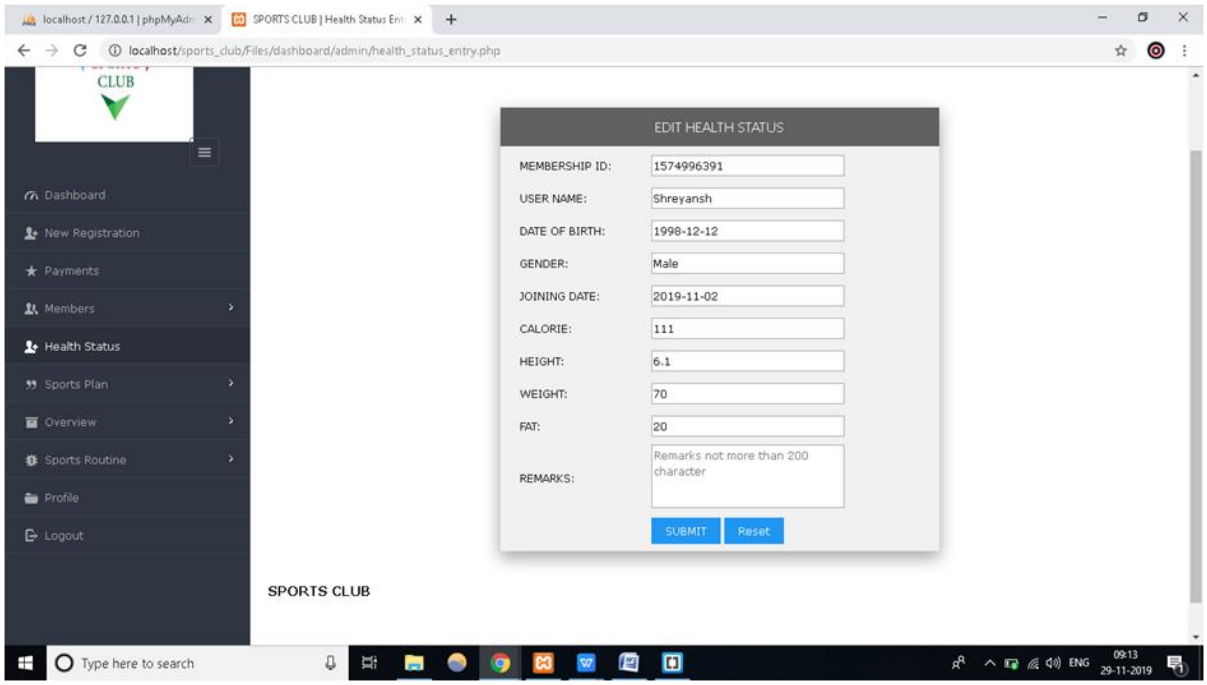


Figure 4.6 Health Status

The below figure refers to the different sports plans available and can be added and edited

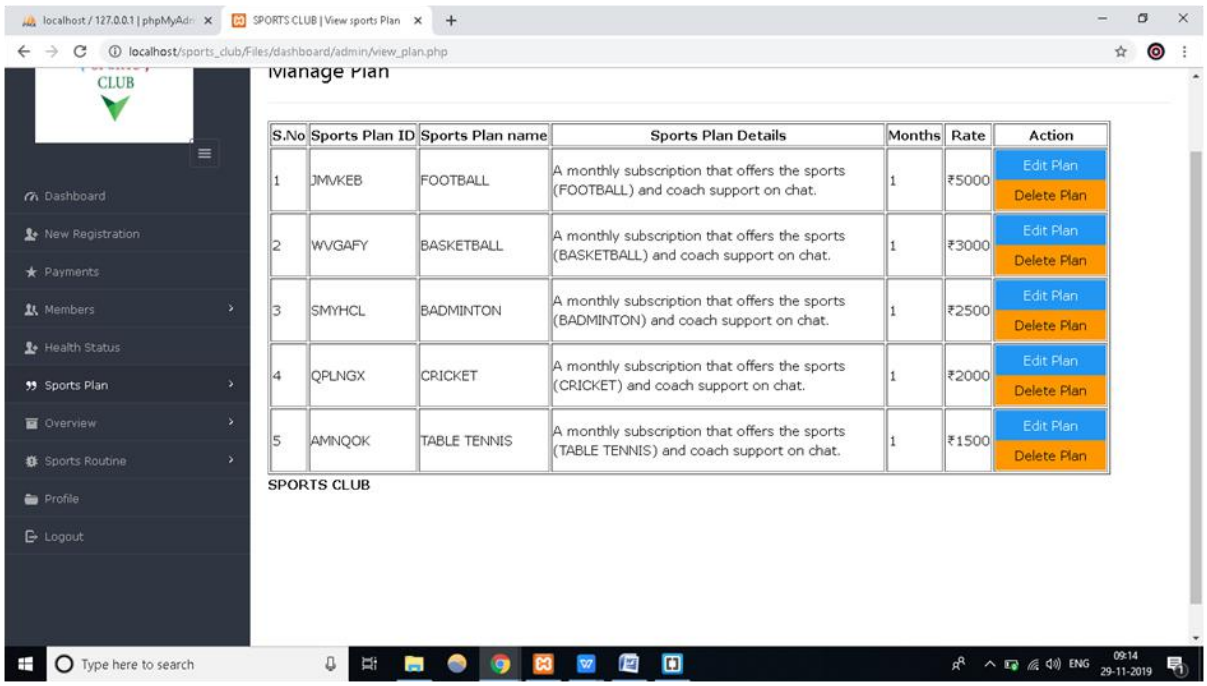


Figure 4.7 Sports Plan

The below figure refers to the members per month and income per month

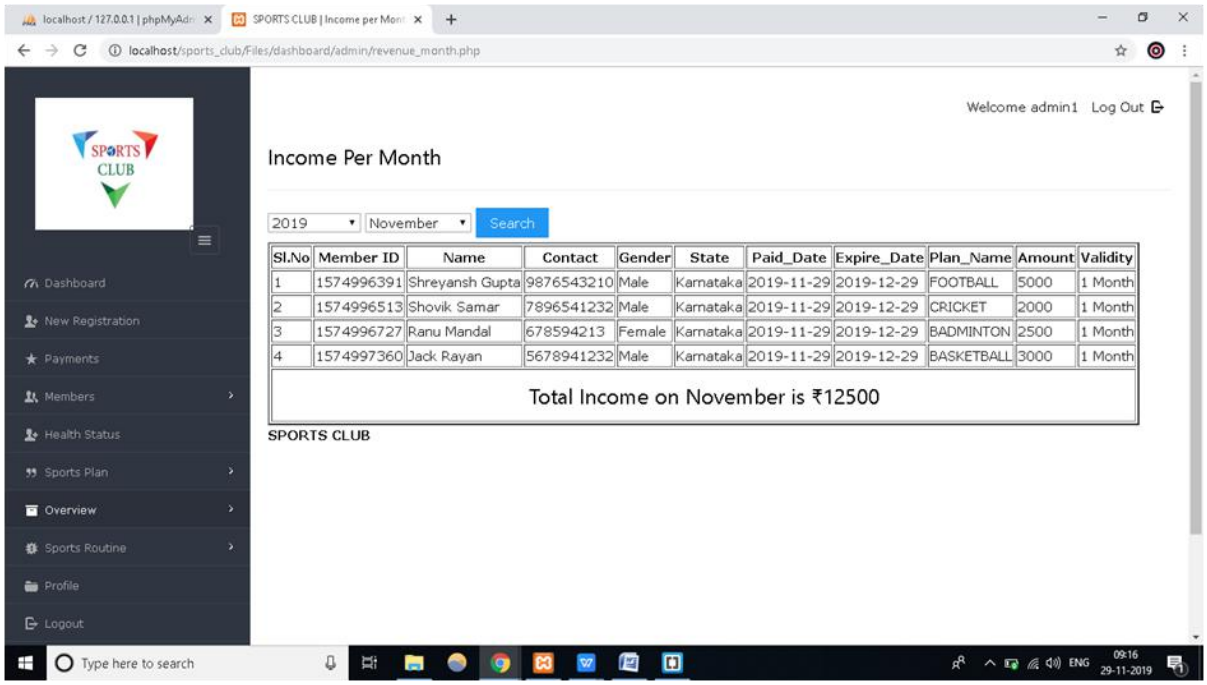


Fig 4.8 Overview income per month

The below figure refers to Sports Routine which is followed every day

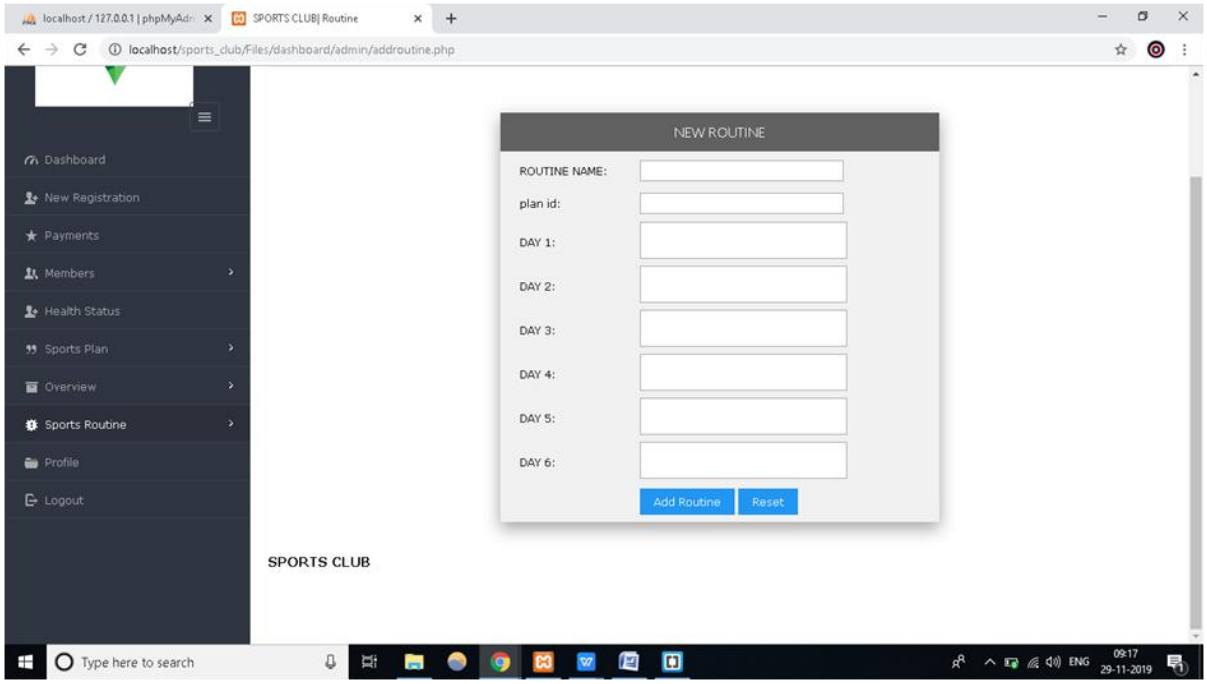


Fig 4.9.1 Add Sports Routine

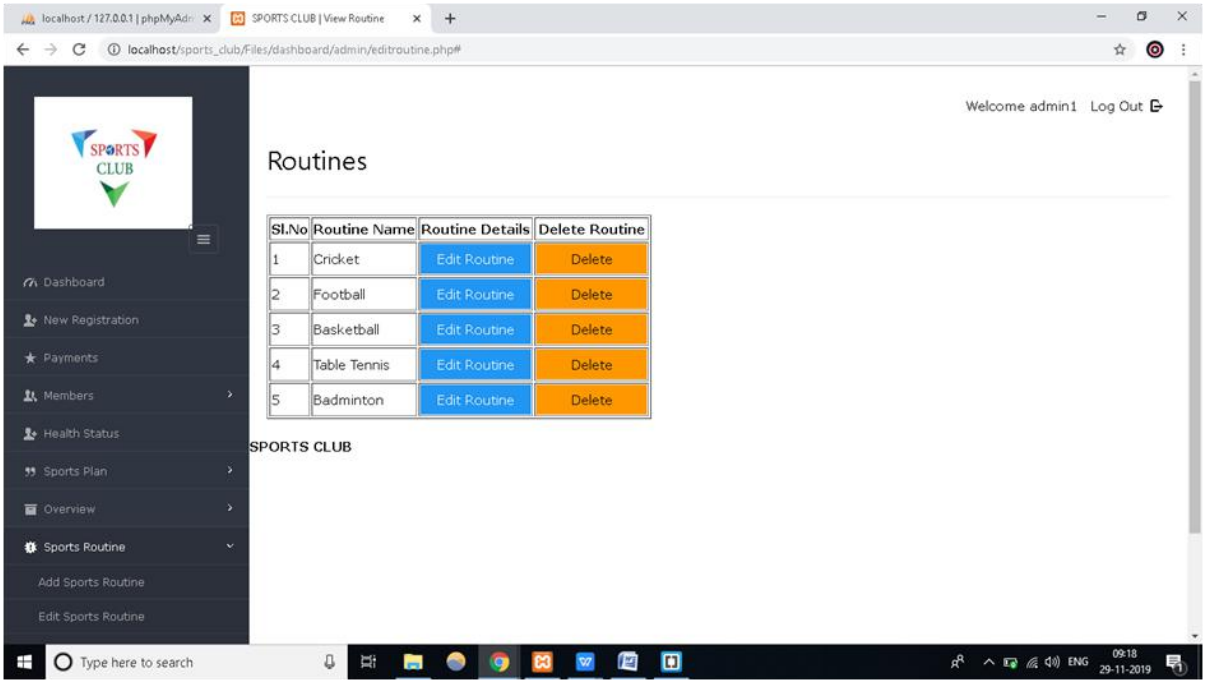


Fig 4.9.2 Edit Sports Routine

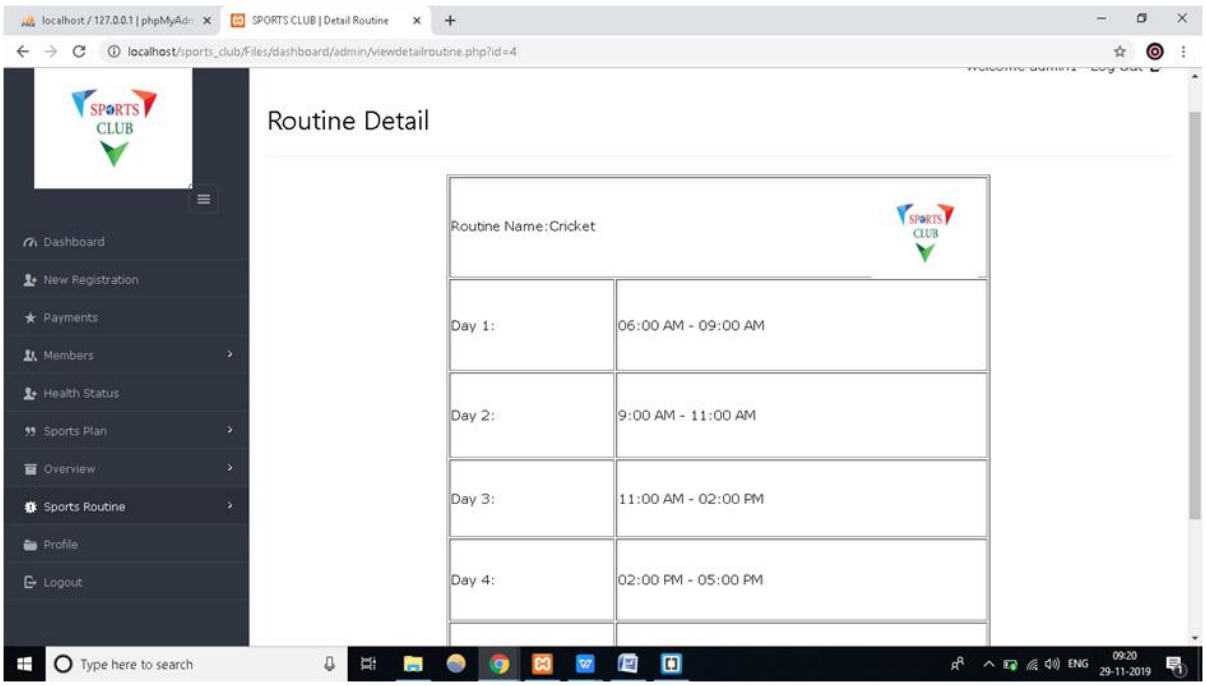


Fig 4.9.3 View Sports Routine

The below figure refers to the Admin login detailed Profile

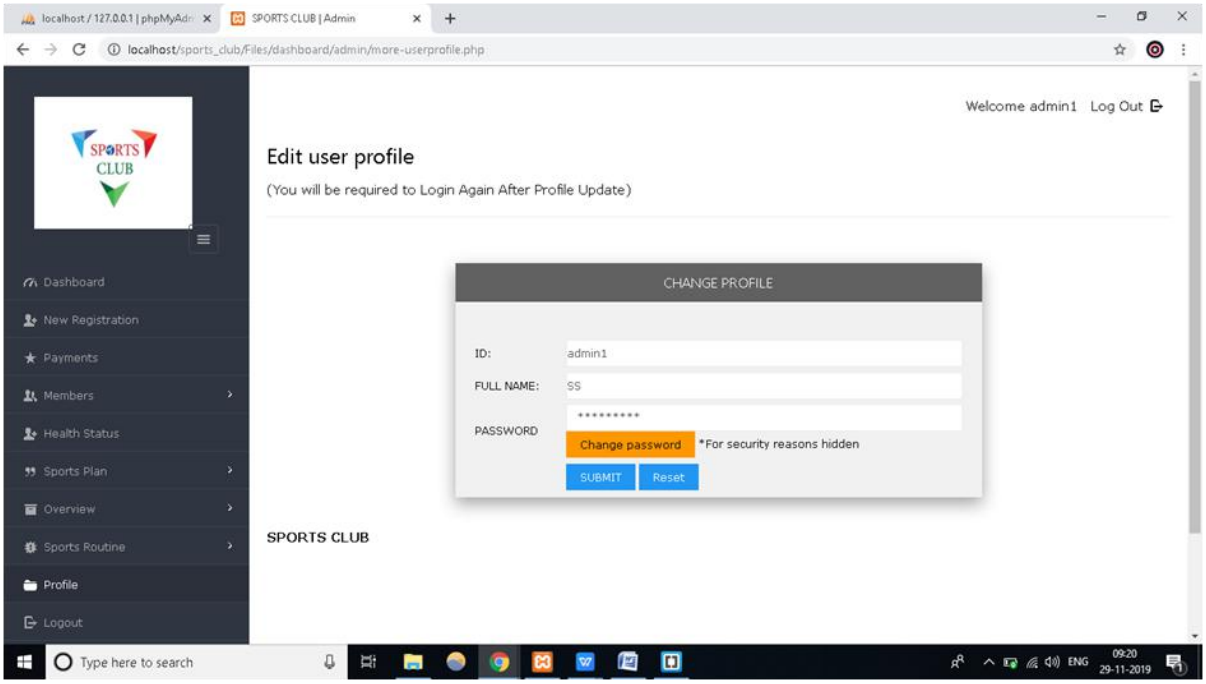


Figure 4.10 Profile

Chapter 5

Conclusion and Future Enhancement

5.1 Conclusion

Planned approach toward working: The maintenance of Sports Club will be well planned and organized. The data will be stored efficiency with optimal disk space consumption in data stores which will help in retrieval of information as well as its storage under resource constraints.

Accuracy: The level of accuracy in the proposed system will be higher. All operations would conform to integrity constraints and correctness and it will be ensured that whatever information is received at or sent from the centre is accurate.

Reliability: The reliability of the proposed system will be high due to the above mentioned reasons. This comes from the fact that only the data which conforms accuracy clause would be allowed to commit back to the disk. Other properties like transaction management and rollback during system or power failure etc get automatically taken care of by the SQL systems, which is undoubtedly an excellent choice of the DBMS system. Properties of atomicity, consistency, isolation and data security are intrinsically maintained.

5.2 Future Enhancement

No redundancy: In the proposed system it will be ensured that no repetition of information occurs; neither on a physical storage nor on a logical implementation level. This economizes on resource utilization in terms of storage space. Also even in case of concurrent access no anomalies occur and consistency is maintained. In addition to all this, principles of normalization have been endeavour to be followed.

Immediate retrieval of information: The main objective of the proposed system is to provide a quick and efficient platform for retrieval of information.queries allowed by the database.

References

[1] Oracle Database MySQL PL/SQL 101- Christopher Allen(Oracle press)

[2] Fundamentals of Database Systems, seventh edition, Elamasri Navathe.

[3] https://stackoverflow.com/questions/26236028/xampp-connect-to-SQL-server

[4] https://www.tutorialspoint.com

[5] www.youtube.com

[6] www.geeksforgeeks.com