Chapter 4

IMPLEMENTATION

Hardware Requirements

Processor	pentium i3,i5 or more
RAM	2GB or Higher
Disk Space	10GB

Software Requirements

Operating System	Windows 7,8,10
Language	PHP,HTML,CSS,
Database	MySQL
Tools	XAMPP

4.1 Front end and Back end used

MySQL

MySQL can be used for a variety of application, but it is most commonly found on Web servers. A Website that uses MySQL may includes Web pages that access information from a database. These pages are often referred to as “dynamic,” meaning the content of each page is generated from a database as the page loads. Website that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a web scripting languages like Php to access information from the database. MySQL commands can be incorporated into the Php code, allowing part or all of a web page to be generated from database information. Because both MySQL and Php are open source, the Php/MySQL combination has become a popular choice for database-driven websites.

MySQL is used as back end. MySQL is a powerful Relational Database Management System (RDBMS) which we will use the learn the basic principles of database using Structures Query Language (SQL) statement. SQL is a database language that is used

to retrieve, insert, delete, update store data. This is achieved by constructing conditional statements that conform to a specific syntaxes.

How does MySQL works

MySQL is a database server program and as such is installed on the machine, but can be used as a 'server' for the database to a variety of location. Fig 4.1 shows the working of MySQL.

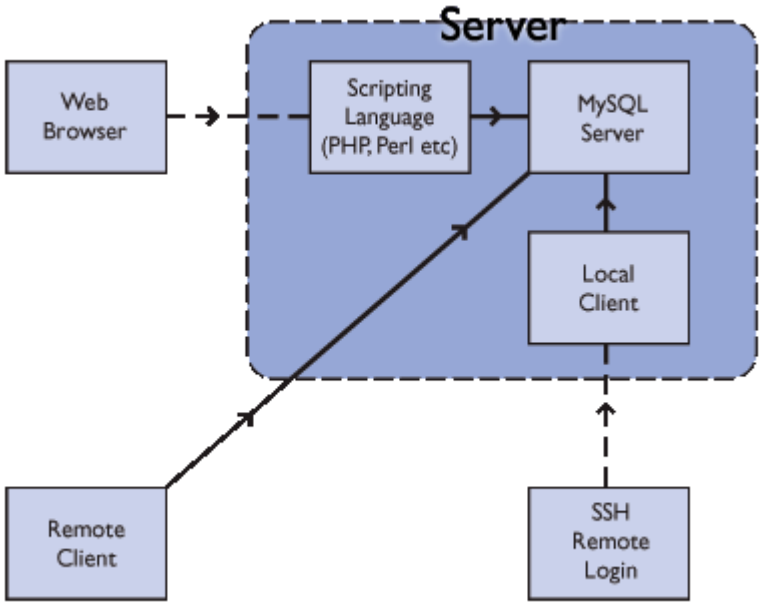

Fig 4.1 MySQL Server

The MySQL Server is installed on a Server and can be accessed directly via various client interfaces, which send SQL statements to the server and then display the results to a user. Some of these are:

A Local Client - a program on the same machine as the server. An example of this is the command line MySQL client software we will be using in the rest of the MySQL workshops (although there are other programs including graphical interfaces).

A Scripting Language - can pass SQL queries to the server and display the result.

A Remote Client - a programme on a different machine that can connect to the server and run SQL statements.

You can also use two more indirect methods.

Remote Login - You may be able to connect to the Server Machine to run one of its local clients.

Web Browser - you can use a web browser and scripts that someone has written (we're going to use this method for the rest of the workshop).

PHP

PHP stands for Hypertext Preprocessor. It is a scripting language designed to fill the gap between SSI(Server side includes) and perl, intended for the web environment. Its principle application is the implementation of web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the open source software movement. Its open popularity derives from its C-like syntax, and its simplicity. The newest version of PHP is 7.0 and it is heavily recommended to always use the newest version for better security performance and of course features.

HTML

HTML is the standard markup language for creating Web pages. HTML stands for Hyper Text Markup Language. Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages.

CSS

CSS stands for Cascading Style Sheets. It is a language used for describing the presentation of the document written in markup language like HTML. It is used for describing the presentation of Web pages, including colors, layouts and fonts. It allows one to adapt the presentaion to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based Markup Language

XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a free and open-source cross-platform web server solution stack package developed by Apache friends. It is simple, light weighted Apache server that makes it

extremely easy for developers to create a local http server with just few clicks. It provides solid and a reliable foundation for building a web applications. It is basically a localhost or local server. This localhost works on our own desktop or computer. The Cross-platform usually means that it can run on any computer with any operating system. MariaDB is the most famous database server and it is developed by MySQL team. PHP usually provides a space for web development. PHP is a server side scripting language. And the last perl is a programming language and is used to develop a web applications.

4.2 Coding

Code for Login page

```
$LoginRS__query=sprintf("SELECT username, password FROM admin_tbl WHERE username=%s AND password=%s",

GetSQLValueString($loginUsername, "text"), GetSQLValueString($password, "text"));

  $LoginRS = mysql_query($LoginRS__query, $localhost) or die(mysql_error());

 $loginFoundUser = mysql_num_rows($LoginRS);

 if ($loginFoundUser) {

   $loginStrGroup = "";

if (PHP_VERSION >= 5.1) {session_regenerate_id(true);} else
{session_regenerate_id();}

   //declare two session variables and assign them

   $_SESSION['MM_Username'] = $loginUsername;

   $_SESSION['MM_UserGroup'] = $loginStrGroup;



   if (isset($_SESSION['PrevUrl']) && false) {

    $MM_redirectLoginSuccess = $_SESSION['PrevUrl'];

   }
```

```
  header("Location: " . $MM_redirectLoginSuccess );
}
else {
  header("Location: ". $MM_redirectLoginFailed );
}
```

Code for Insertion

```
$insertSQL = sprintf("INSERT INTO tbl_userreg (firstname, lastname, age, sex,
phone, address, service, timestap, amount, plan) VALUES (%s, %s, %s, %s, %s, %s,
%s, %s, %s, %s)",
                    GetSQLValueString($_POST['fname'], "text"),
                    GetSQLValueString($_POST['lname'], "text"),
                    GetSQLValueString($_POST['agetxt'], "int"),
                    GetSQLValueString($_POST['radio'], "text"),
                    GetSQLValueString($_POST['phonetxt'], "int"),
                    GetSQLValueString($_POST['addresstxt'], "text"),
                    GetSQLValueString($_POST['service'], "text"),
                    GetSQLValueString($_POST['datetime-local'], "date"),
                    GetSQLValueString($_POST['amount'], "int"),
                    GetSQLValueString($_POST['select'], "int"));
```

Code for Search

```
$searchword=$_POST['txt_search'];
$query_Recordset1="select * from tbl_userreg where firstname LIKE
'%".$searchword."%'";
```

Code for Update

```
$updateSQL = sprintf("UPDATE tbl_userreg SET firstname=%s, lastname=%s,
age=%s, sex=%s, phone=%s, address=%s, service=%s, amount=%s WHERE
userid=%s",

        GetSQLValueString($_POST['firstname'], "text"),

        GetSQLValueString($_POST['lastname'], "text"),

        GetSQLValueString($_POST['age'], "int"),

        GetSQLValueString($_POST['sex'], "text"),

        GetSQLValueString($_POST['phone'], "int"),

        GetSQLValueString($_POST['address'], "text"),

        GetSQLValueString($_POST['service'], "text"),

        GetSQLValueString($_POST['amount'], "int"),

        GetSQLValueString($_POST['userid'], "int"));
```

Code for Delete

```
$deleteSQL = sprintf("DELETE FROM tbl_userreg WHERE userid=%s",

        GetSQLValueString($_GET['userid'], "text"));
```

Code for Connection

```
<?php
$hostname_localhost = "localhost";
$database_localhost = "gym";
$username_localhost = "root";
$password_localhost = "";
$localhost = @mysql_pconnect($hostname_localhost, $username_localhost,
$password_localhost) or trigger_error(mysql_error(),E_USER_ERROR); ?>
```

Code for logout

```
if (!isset($_SESSION)) {
  session_start();
}
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true";
if ((isset($_SERVER['QUERY_STRING'])) && ($_SERVER['QUERY_STRING'] != "")){
  $logoutAction .="&". htmlentities($_SERVER['QUERY_STRING']);
}
if ((isset($_GET['doLogout'])) &&($_GET['doLogout']=="true")){
  $_SESSION['MM_Username'] = NULL;
  $_SESSION['MM_UserGroup'] = NULL;
  $_SESSION['PrevUrl'] = NULL;
  unset($_SESSION['MM_Username']);
  unset($_SESSION['MM_UserGroup']);
  unset($_SESSION['PrevUrl']);
   $logoutGoTo = "../index.php";
  if ($logoutGoTo) {
    header("Location: $logoutGoTo");
    exit;
  }
}
```

Code for Mainframe

```
mysql_select_db($database_localhost, $localhost);

$query_totalmembers = "SELECT * FROM tbl_userreg";

$totalmembers = mysql_query($query_totalmembers, $localhost) or
die(mysql_error());

$row_totalmembers = mysql_fetch_assoc($totalmembers);

$totalRows_totalmembers = mysql_num_rows($totalmembers);

mysql_select_db($database_localhost, $localhost);

$query_amountsum = "SELECT SUM( amount) FROM tbl_userreg";

$amountsum = mysql_query($query_amountsum, $localhost) or die(mysql_error());

$row_amountsum = mysql_fetch_assoc($amountsum);

$totalRows_amountsum = mysql_num_rows($amountsum)
```

4.3 Applications of Project Work

i. User Registration Form is to retrieve and update user information.
ii. Print view of invoice can be done.
iii. Ability to show member plan expiration and accounts status.
iv. Income growth and member growth plan is supported.
v. Helps to delete member details.
vi. Payment status of members can be easily viewed.
vii. Equipment expenses can be seen.
viii. Vendor details are stored.
ix. Security of data of clients.
x. Minimizes manual data entry.
xi. Better service with greater efficiency.

4.4 Discussion of the Results

Login Page: In login page it allows the admin to enter his username and password as shown in Fig 4.2. If login is successful it goes to main frame. If the username and password is invalid then it displays an unsuccessful login and asks for valid username and password.

Fig 4.2 Login Page

Main Frame: It has a menu bar with Dashboard, Member List, Equipment List, Vendor, Payment, Status, Logout and also contains User registration, Equipment registration, Payment, Manage Members as shown in Fig 4.3.

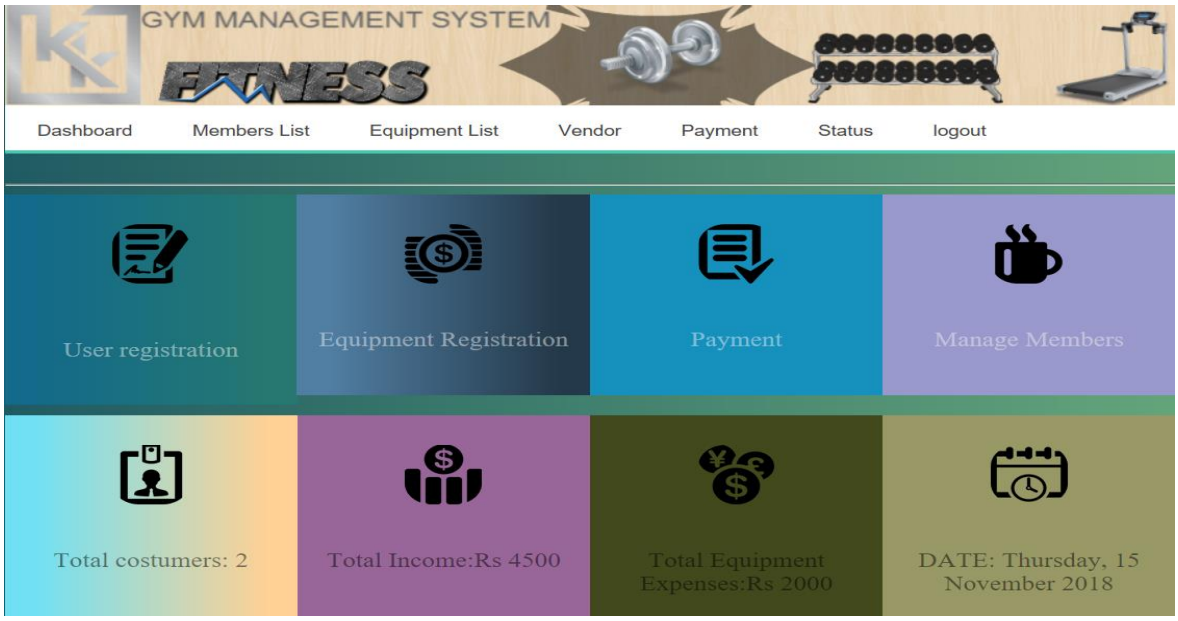

Fig 4.3 Main Frame

User Registration: It has a dropdown menu where we have to insert the details and when we click on submit it shows the member list as shown in Fig 4.4.

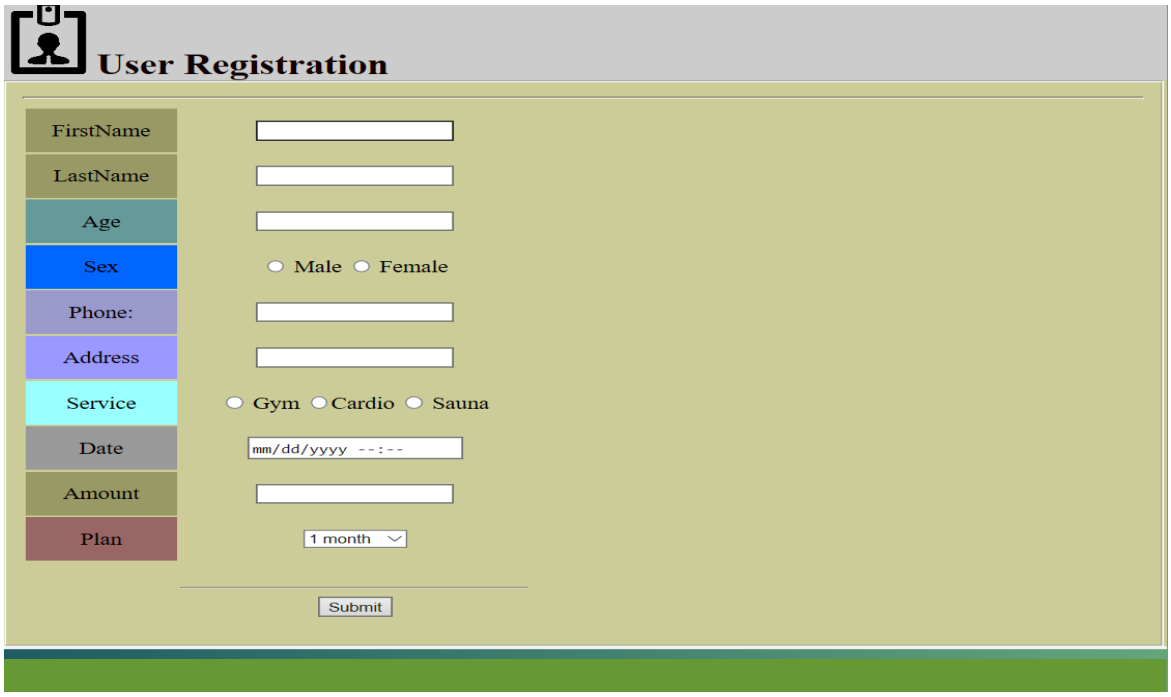

Fig 4.4 User Registration

Member List – It retrieves the details of the members. We can also edit, delete and print the details of the members as shown in Fig 4.5.

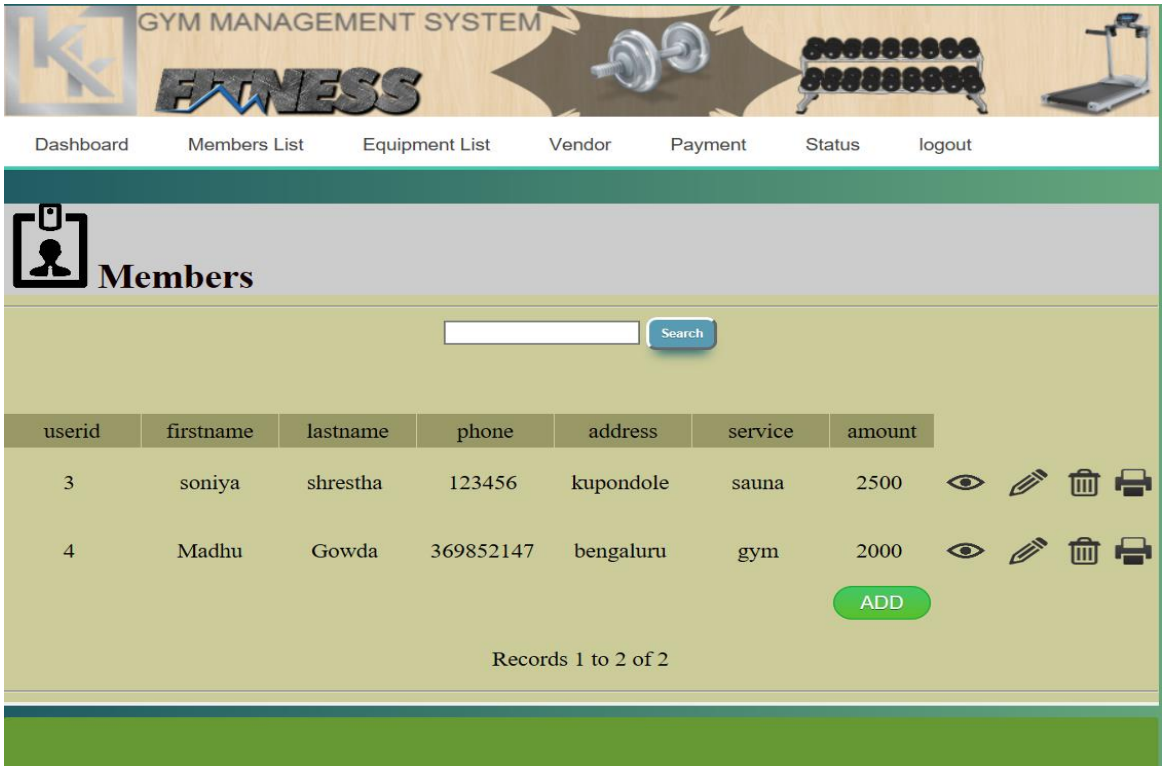

Fig 4.5 Member List

Equipment Registration: It has a dropdown menu where we have to insert the details of equipments as shown in Fig 4.6.

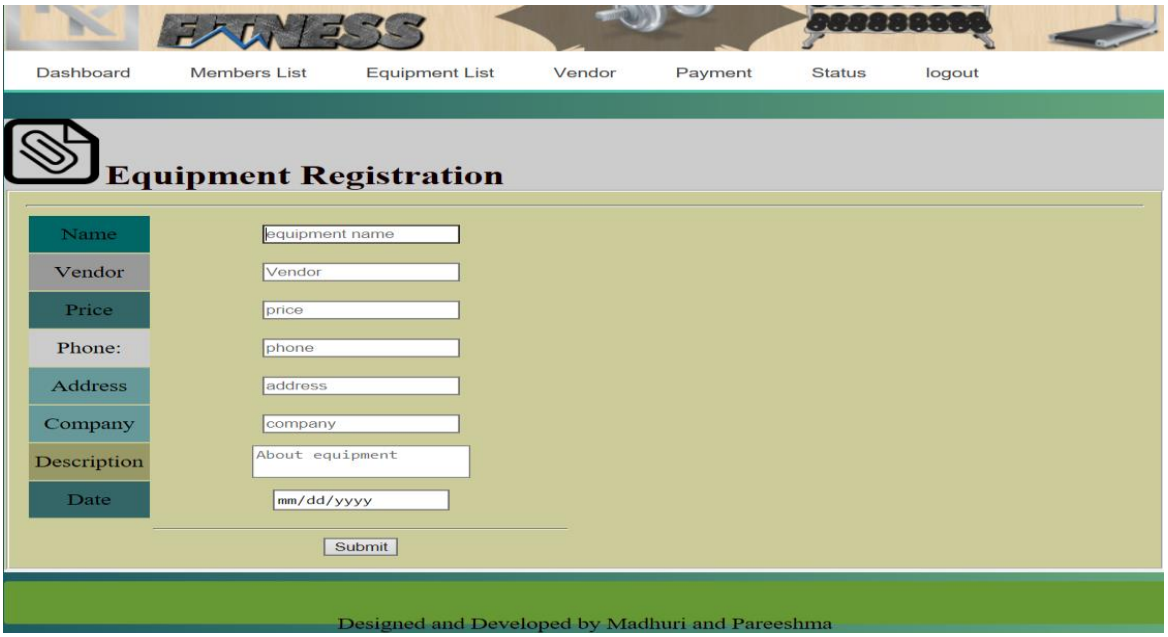

Fig 4.6 Equipment Registration

Equipment List: It retrieves the details of the equipments and we can also delete or add the records of the equipments as shown in Fig 4.7.

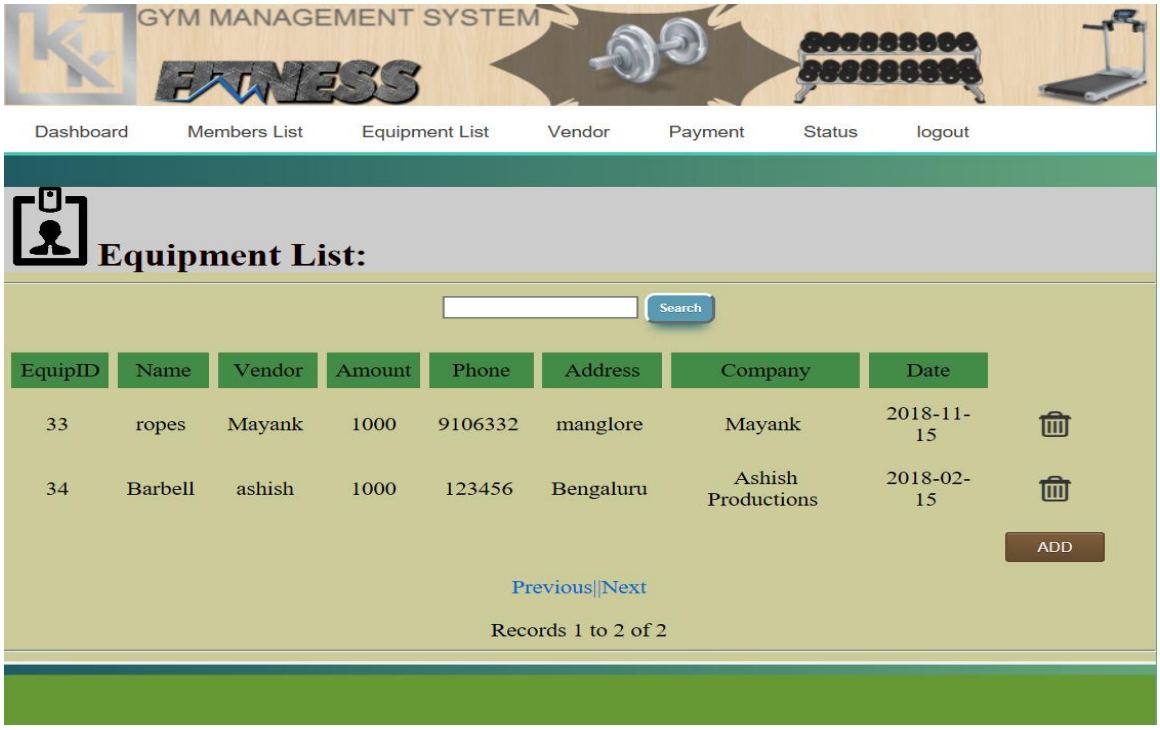

Fig 4.7 Equipment List

Vendor List: It retrieves the details of vendors and we can also delete or add the records of vendors as shown in Fig 4.8.

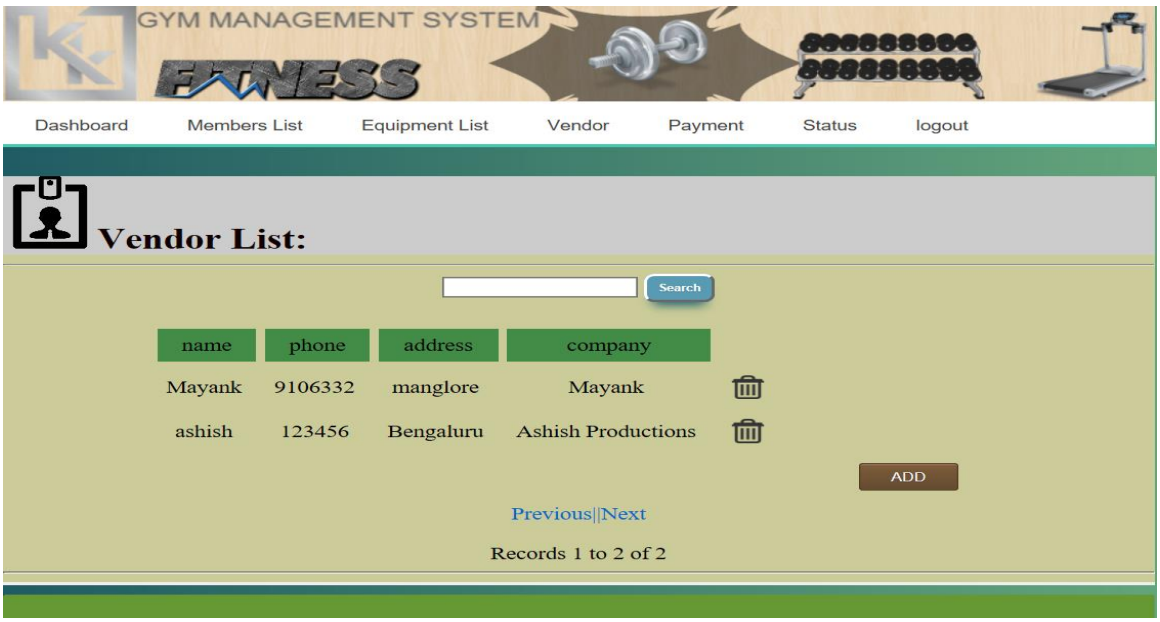

Fig 4.8 Vendor List

Payment: It displays the payment list of the members as shown in Fig 4.9.

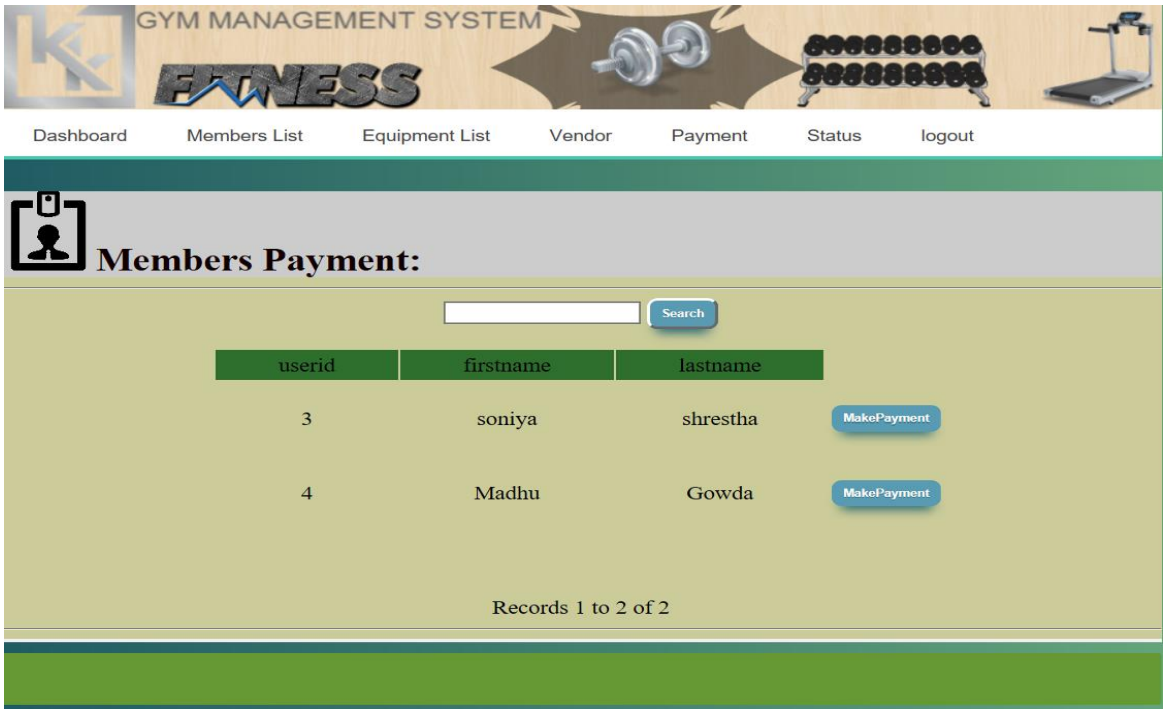

Fig 4.9 Payment details

Make Payment: Displays the detailed payment list of a client as shown in Fig 4.10.

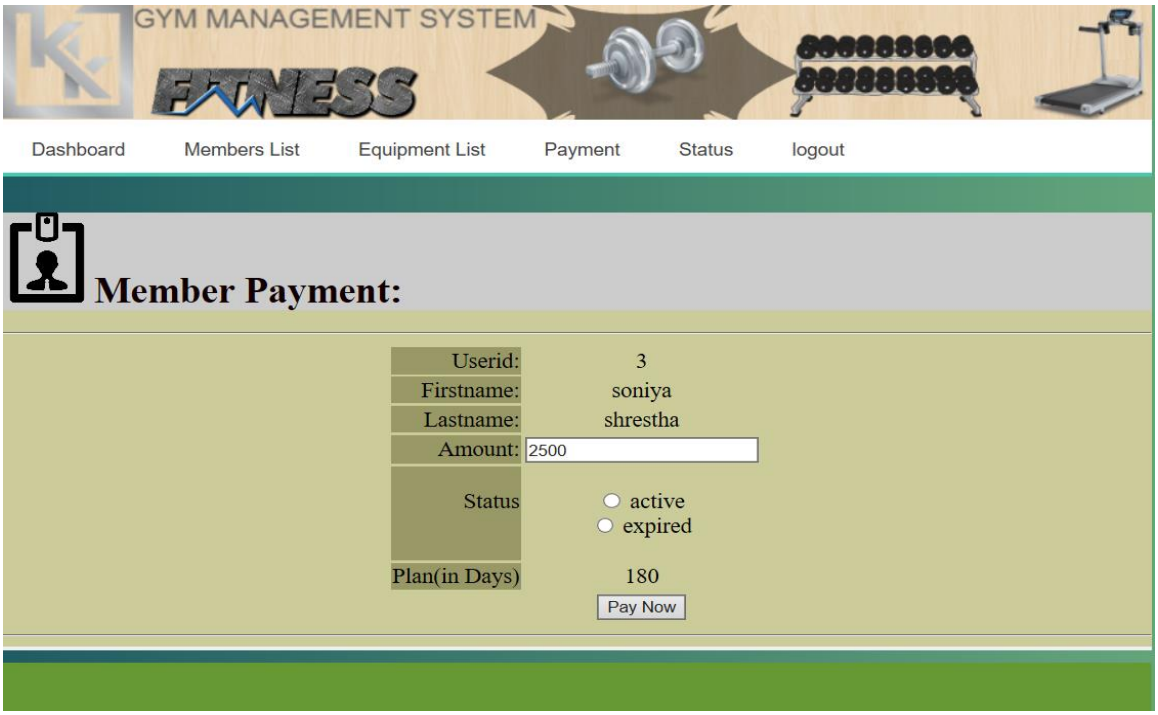

Fig 4.10 Make Payment

Status – It shows the status details of all the members by displaying their name, service they have registered for and their plans as shown in Fig 4.11.

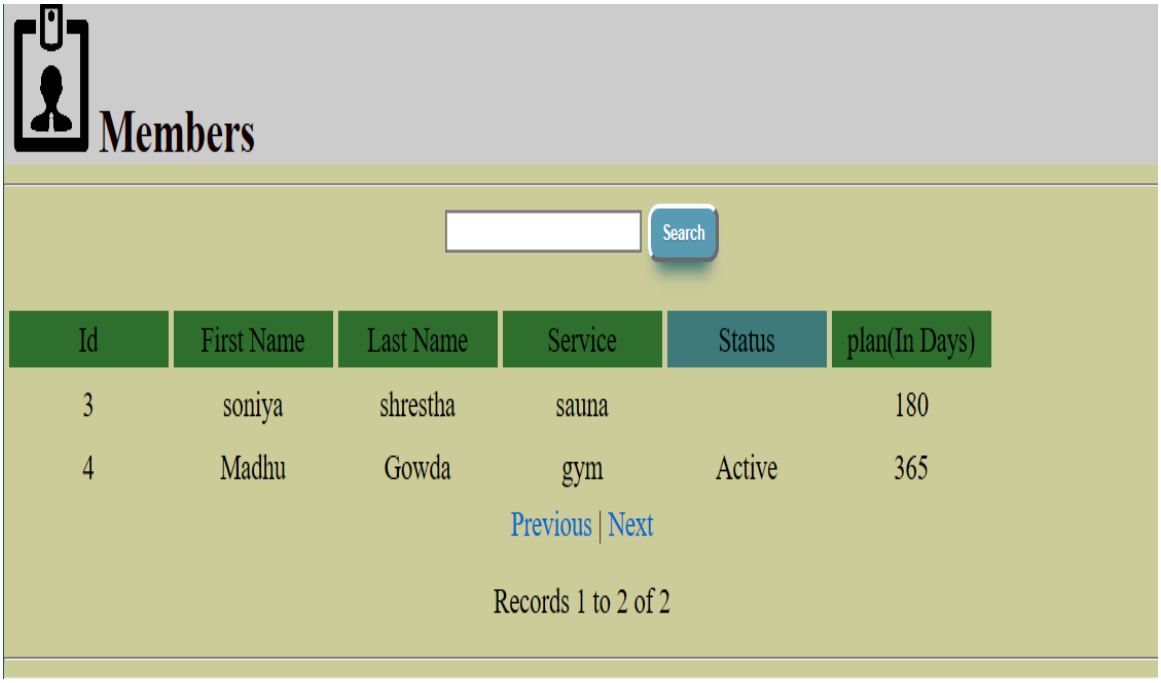

Fig 4.11 Status of all client