# INTRODUCTION TO PROJECT

Gyms are the essential part of our lives, providing best exercise and body building facilities to our society. Therefore, at the management end there are some necessary steps to maintain the records of every individual including trainer, gym members, and admin but maintaining the records on paper is very difficult. So it is necessary to have a computerized system that manages all these functionality. Thus working on the management system for Gym Industry is the basis of our project.

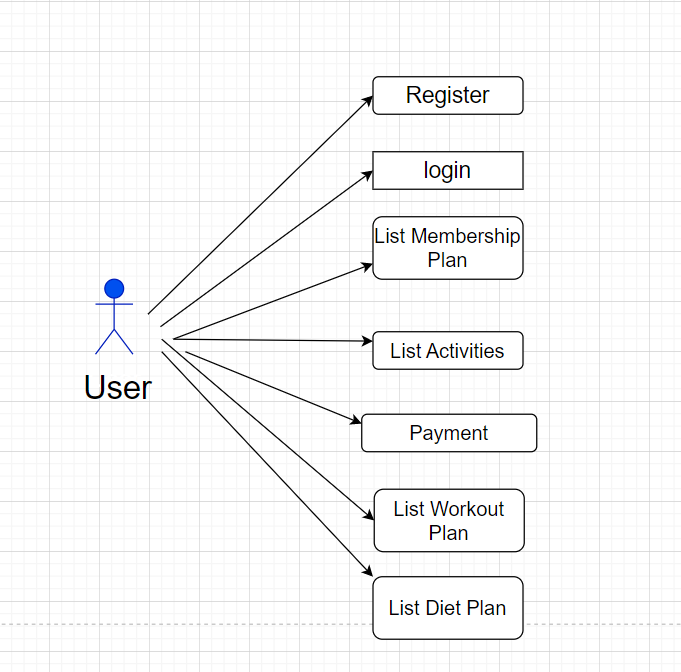
The web based “Gym Management System” project is an attempt to simulate basic concepts of gym management system. The system enables the member to do the things such as search for workout-plans, diet plans and choose them based on the details of the plans.

The system provides a facility that provides details about gym contact details with locations, Gallery, Programs and Blogs. But if user wants to book plan then he/she must login into his/her account.

The system allows the gym-members to search for subscription-plans, workout-plans and diet-plans which are provided by the management. The system displays all the plan’s details such as plan-Id, plan- Name, duration and amount. Services provided by Gym are also handled by this system.

**2.REQUIREMENTS**

**2.1 FUNCTIONAL REQUIREMENTS**

****

**2.1 User Account**

First user has to fill the details using ‘sign up’ function. In this form user has been asked to fill their name, address, email id, contact and password, joining date will be automatically selected, and also there will be option to choose membership where various memberships are present called regular, gold, silver and diamond etc. according to that user has to select plan.

The Gym Management System then activates the login form. Here the user enters the User name and password and our system starts the authentication process in which the username and password are matched with the existing username and password in the database. If the password matches then it is allowed to the ‘sign in’ page else it will give

Alert to user as Invalid User name or password. After successful authentication the system activates menus.

**2.2 Registration and creation of user profile**

The system shall require a user to register, in order to access further functionality. It will ask the user for the information such as name, password, address, contact number and email address.

Existing user can directly login from sign in page by providing email id and password. After successful login user can see plans provided by management and select one plan from list while filling the ‘membership form’, activated plan details are visible in My orders section.

A new user, on the other hand, would either have to register himself with the tem by providing personal information without providing personal details user cannot access any of facility which is provided by the gym.

.

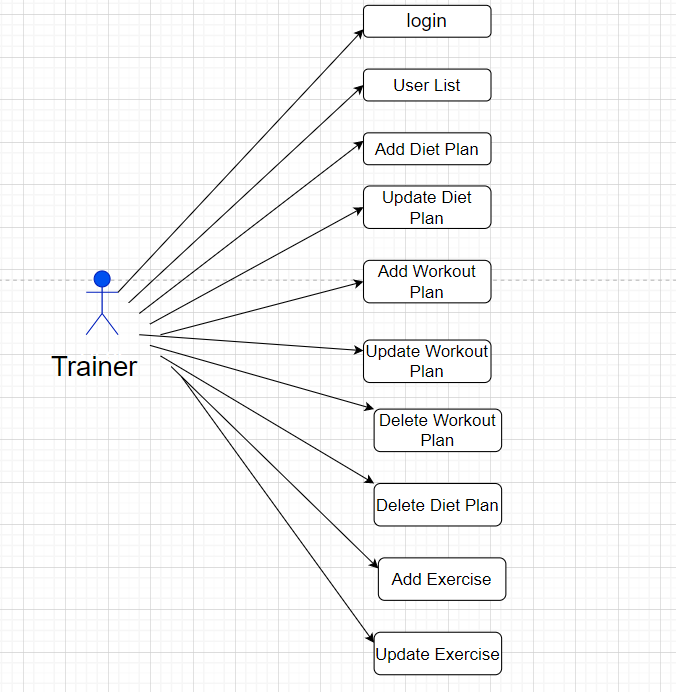
****

* 1. **Trainer Account**

To create trainer first we need to login from admin account. Admin has to fill the details using ‘add user’ function. In this form admin needs to fill their name, address, email id, contact ,password and ‘role as trainer’.

Existing trainer can directly login from sign in page by providing email id and password. Trainer can view his/her own profile.

After successful login trainer can perform add, delete and update on diet-plans table and workout-plans table.



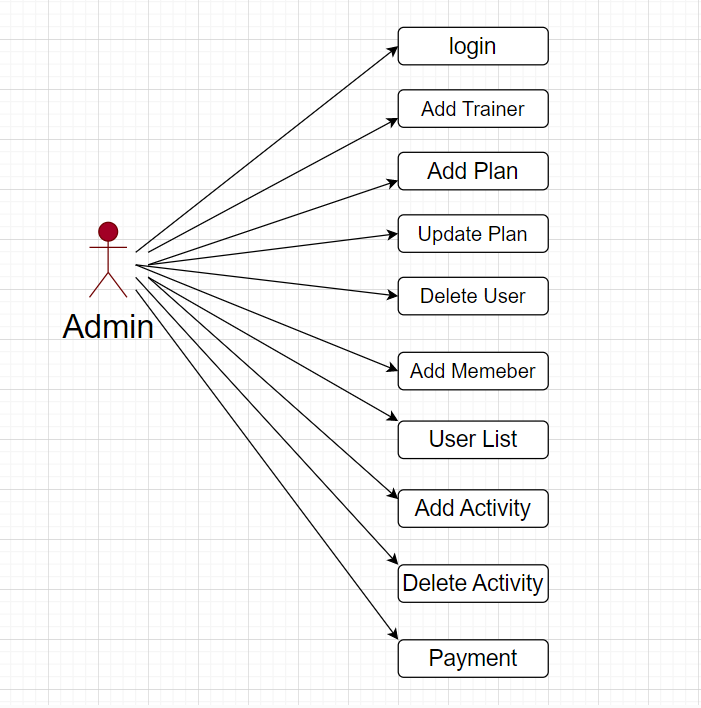
* 1. **Admin Account**

With the help of the login page admin can login with email-id and password. Admin is the one who runs the gym and has all the permissions to see the member details as well as trainer details and also the promotional materials which are very important part of the gym.

To create trainer first we need to login from admin account because only an existing admin can create another admin. Existing admin can directly login from sign in page by providing email id and password. Admin can view his/her own profile.

An admin can view the user-list which contains all the users who have not subscribed to any plan, trainers and admins as well.

After successful login admin can perform add, delete and update on membership-plans. Admin can also view ‘gym member list’ which includes all the users who have subscribed to a plan.



* 1. **Quick Search**

Here we provided Quick Search facility for any user to search gym facilities without login into account .This will provide user an option for home, gallery, programs, blogs and contacts.

Home page contains basic information about our website and it is the home page for the Gym Management System application.

Gallery contains collection of images from our different gym locations. Program contains different types of training provide by our gym and events conducted by management such as blood test etc. It also includes various wellness programs for corporate people as well as individuals.

Blogs contains discussion and information regarding nutrition, health awareness and our gym member’s inspirational stories and experiences.

Contact contains different branch details with concern persons information.

**3. DESIGN**

**3.1 Database Design**

The following table structures depict the database design.

# Table1: User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| PK | User\_id | int | 4 | 0 |
|  | Address | Varchar | 255 | 1 |
|  | Contact | Varchar | 255 | 1 |
|  | Email | Varchar | 255 | 1 |
|  | Name | Varchar | 255 | 1 |
|  | Password | Varchar | 255 | 1 |
|  | Role | Varchar | 255 | 1 |

## **Table2: Gym member**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| PK | MemberId | int | 4 | 0 |
|  | Gender | Varchar | 255 | 1 |
|  | Height | Double | 4 | 0 |
|  | JoinDate | Date | 10 | 1 |
|  | Weight | Double | 4 | 0 |
| FK | MembershipPlanId | int | 4 | 1 |
| FK | User\_id | int | 4 | 1 |
|  | Payment\_status | Bit(1) | 1 | 0 |
| FK | Dietplan\_Id | int | 4 | 1 |
| FK | Workout\_Id | int | 4 | 1 |
| FK | Activity\_Id | int | 4 | 1 |

## **Table3: Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| PK | SubplanId | int | 4 | 0 |
|  | Amount | int | 4 | 0 |
|  | Duration | int | 4 | 0 |
|  | PlanName | Varchar | 255 | 1 |

## **Table4: DietPlan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| PK | Planid | int | 4 | 0 |
|  | Breakfast | Varchar | 255 | 1 |
|  | Dinner | Varchar | 255 | 1 |
|  | Lunch | Varchar | 255 | 1 |
|  | Planname | Varchar | 255 | 1 |
|  | Preworkout | Varchar | 255 | 1 |

## **Table5: WorkoutPlan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null (1=Yes;0=No)** |
| PK | Workout\_Id | int | 4 | 0 |
|  | Set1 | Varchar | 255 | 1 |
|  | Set2 | Varchar | 255 | 1 |
|  | Set3 | Varchar | 255 | 1 |
|  | Workout\_Name | Varchar | 255 | 1 |

**E-R Diagram, Use Case diagram and Class Diagram:**

Go to Appendix A

Go to Appendix B

Go to Appendix C

**4. CODING STANDARDS IMPLEMENTED**

### Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | Person, BankVault, SMSMessage, Dept | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type prefixes like ‘C’ for class. |
| Method | Camel | getDetails, updateStore | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | personName, bankCode | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios. |
| Interface | Pascal with "I" prefix | Disposable | Do not use the ‘\_’ sign |
| Property | Pascal | ForeColor, BackColor | Use a noun or noun phrase to name properties. |
| Associated private member variable | \_camelCase | \_foreColor, \_backColor | Use underscore camel casing for the private member variables |
| Exception Class | Pascal with "Exception" suffix | WebException, |  |

### Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

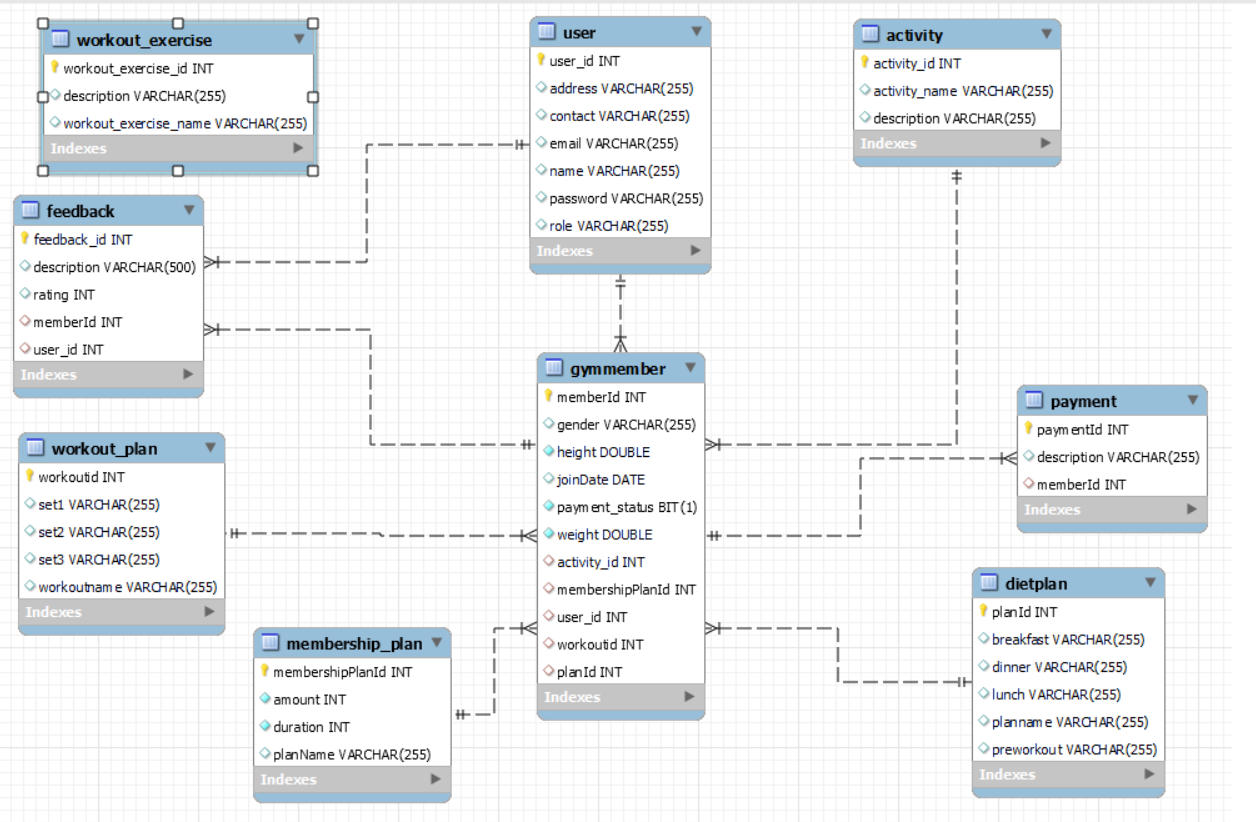
**5. TEST REPORT**

**GENERAL TESTING:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
|  | Register Page | Redirected to Sign in Page | OK | Enter Required Fields |
|  | Sign in Page | Redirected to User UI | Ok | Please enter username and password again. |
|  | Checking login or not | User is logged in or not | Ok | Nothing |
|  | User Created | New user added to list | ok | Nothing |
|  | Find User | User Details Displayed | ok | Nothing |
|  | Find All | User Details Displayed | ok | Nothing |
|  | Membership Plan | All the fields should be filled for submission | Ok | Enter Required Fields |
|  | Health Status | BMI displayed | Ok | Enter Required Fields |
|  | Delete Plan | Plan removed from list | Ok | Nothing |
|  | Add Diet-Plan | All the fields should be filled for submission | Ok | Enter Required Fields |
|  | Update Diet-Plan | Diet-Plan Updated | Ok | Enter Required Fields |
|  | Delete Diet-Plan | Diet-Plan Removed from list | Ok | Nothing |
|  | Add Workout-Plan | All the fields should be filled for submission | Ok | Enter Required Fields |
|  | Update Workout Plan | Workout-Plan Updated | Ok | Enter Required Fields |
|  | Delete Workout-Plan | Workout -Plan Removed from list | Ok | Nothing |

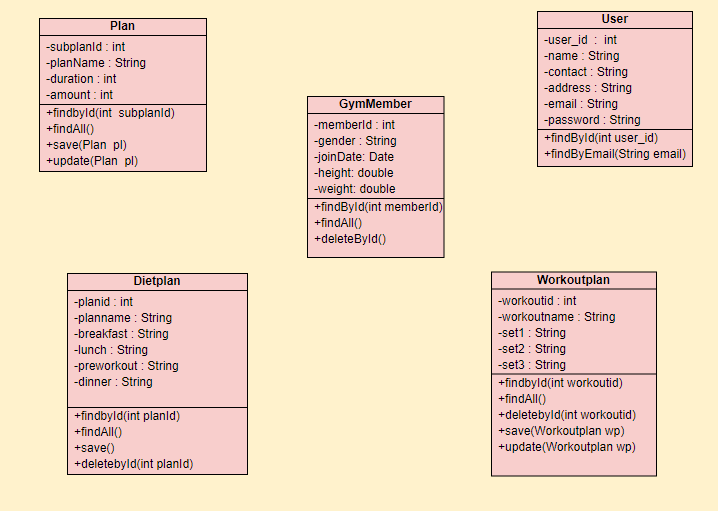
Appendix A

Entity Relationship Diagram



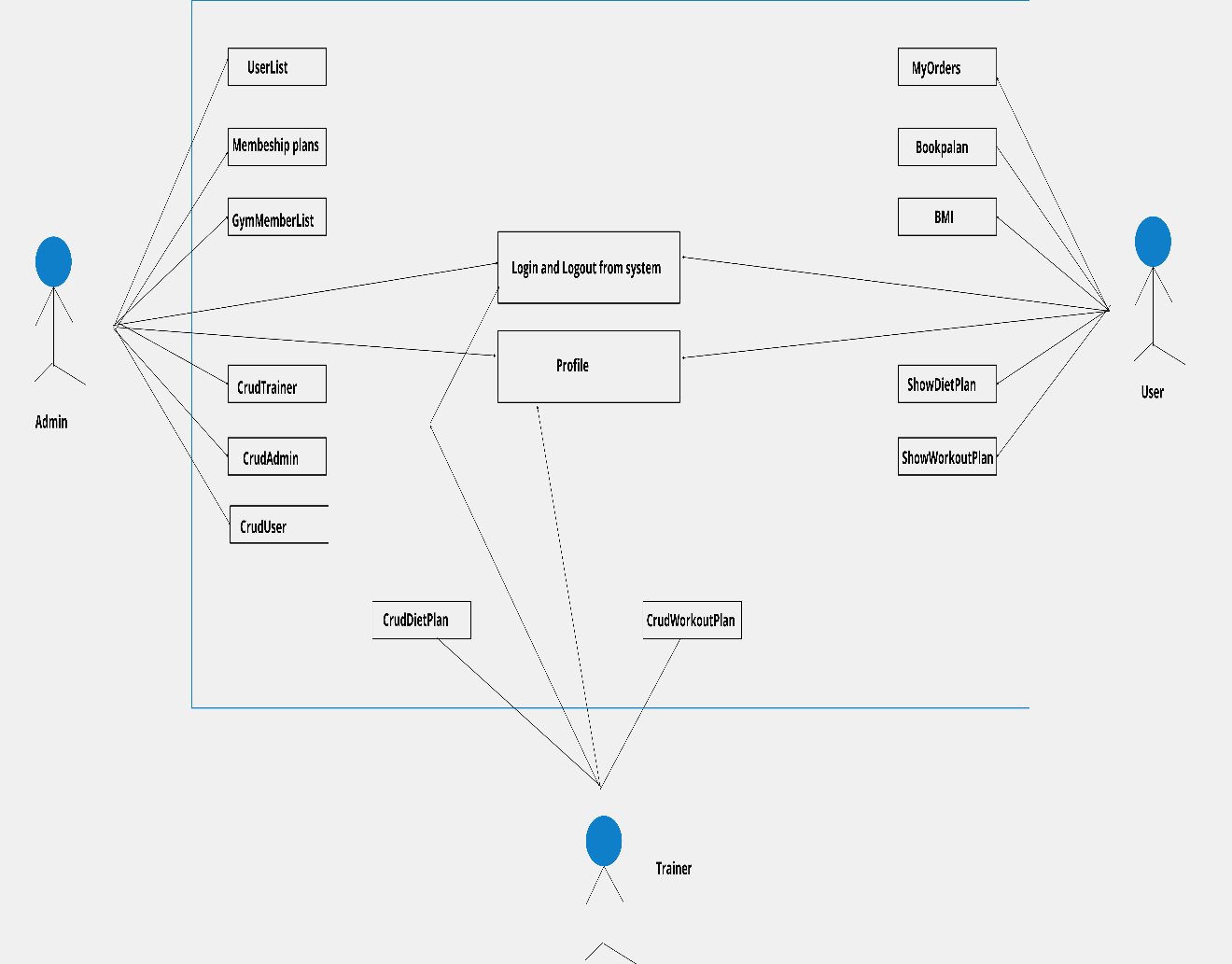
AppendixB

**Class Diagram**



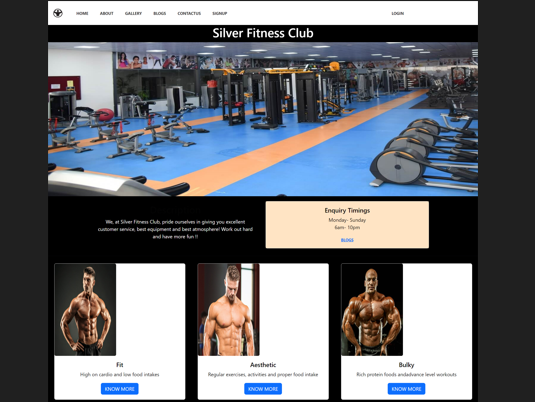
AppendixC

**Use Case Diagram**

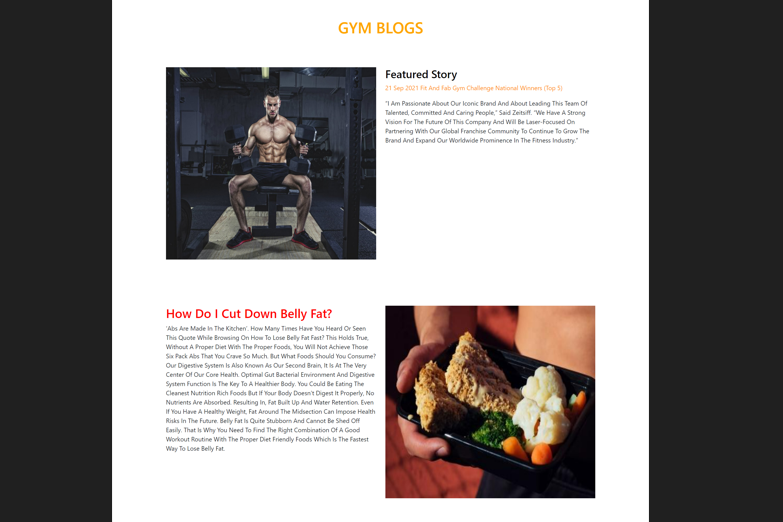


AppendixD

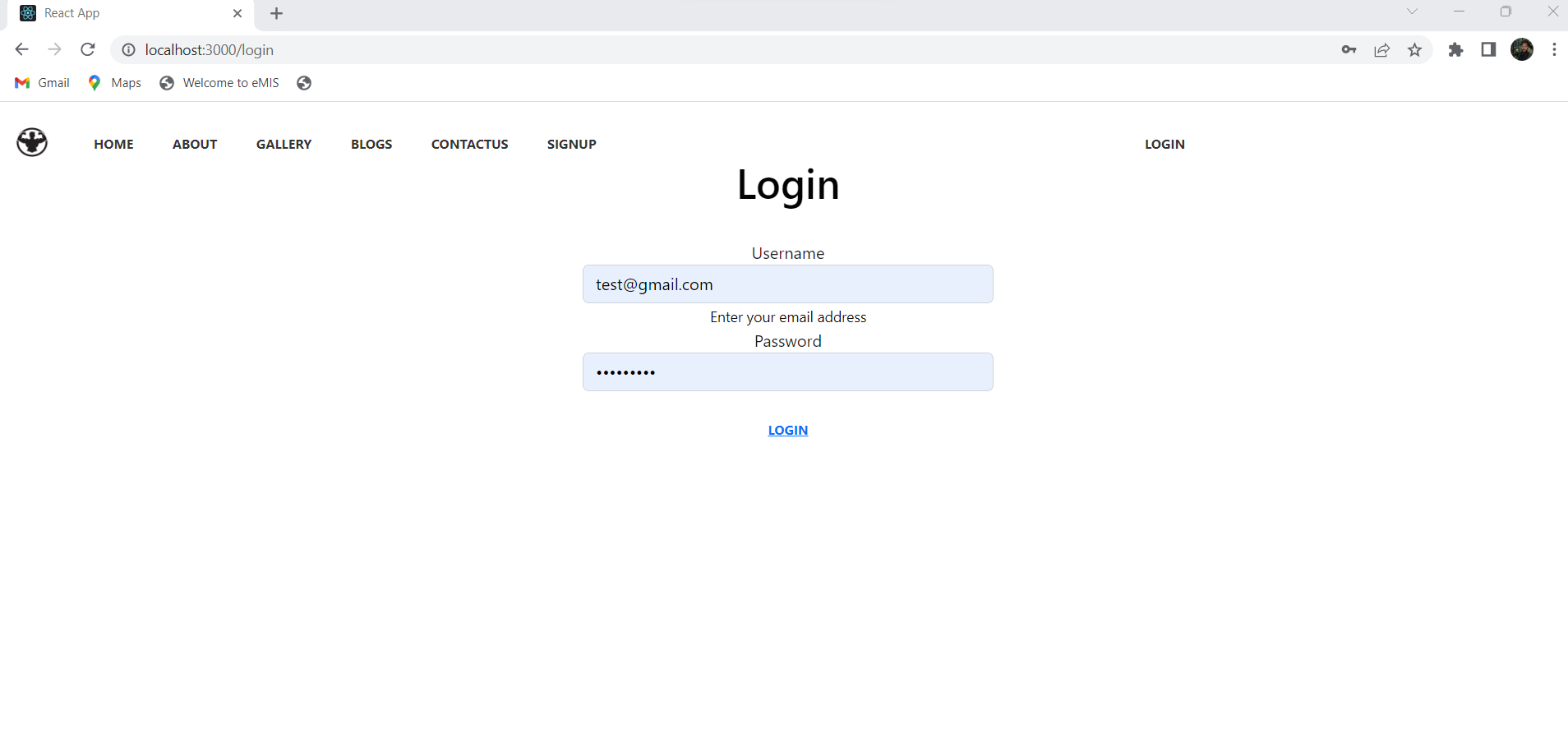
Homepage:



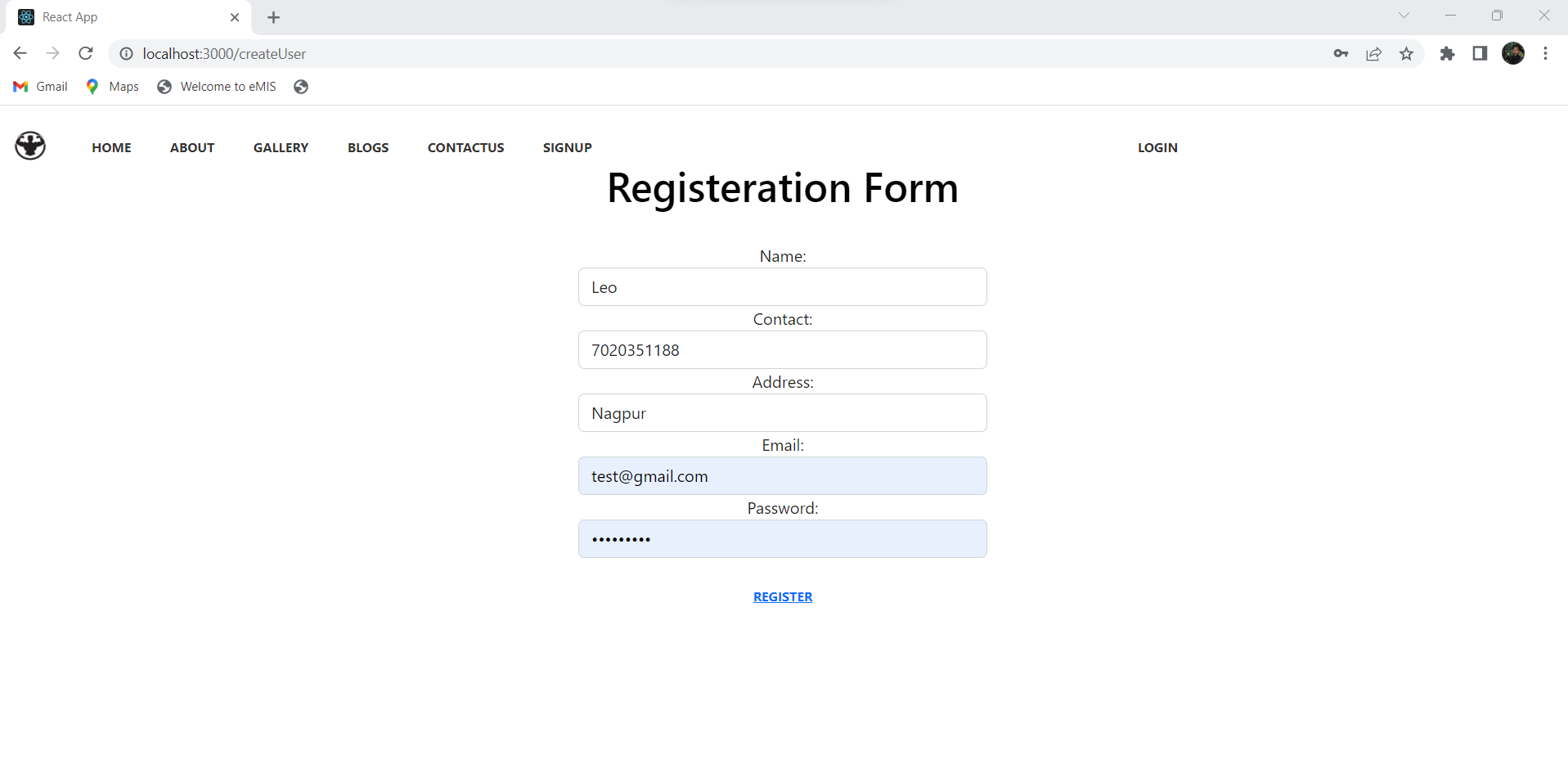
**Blogs:**

****

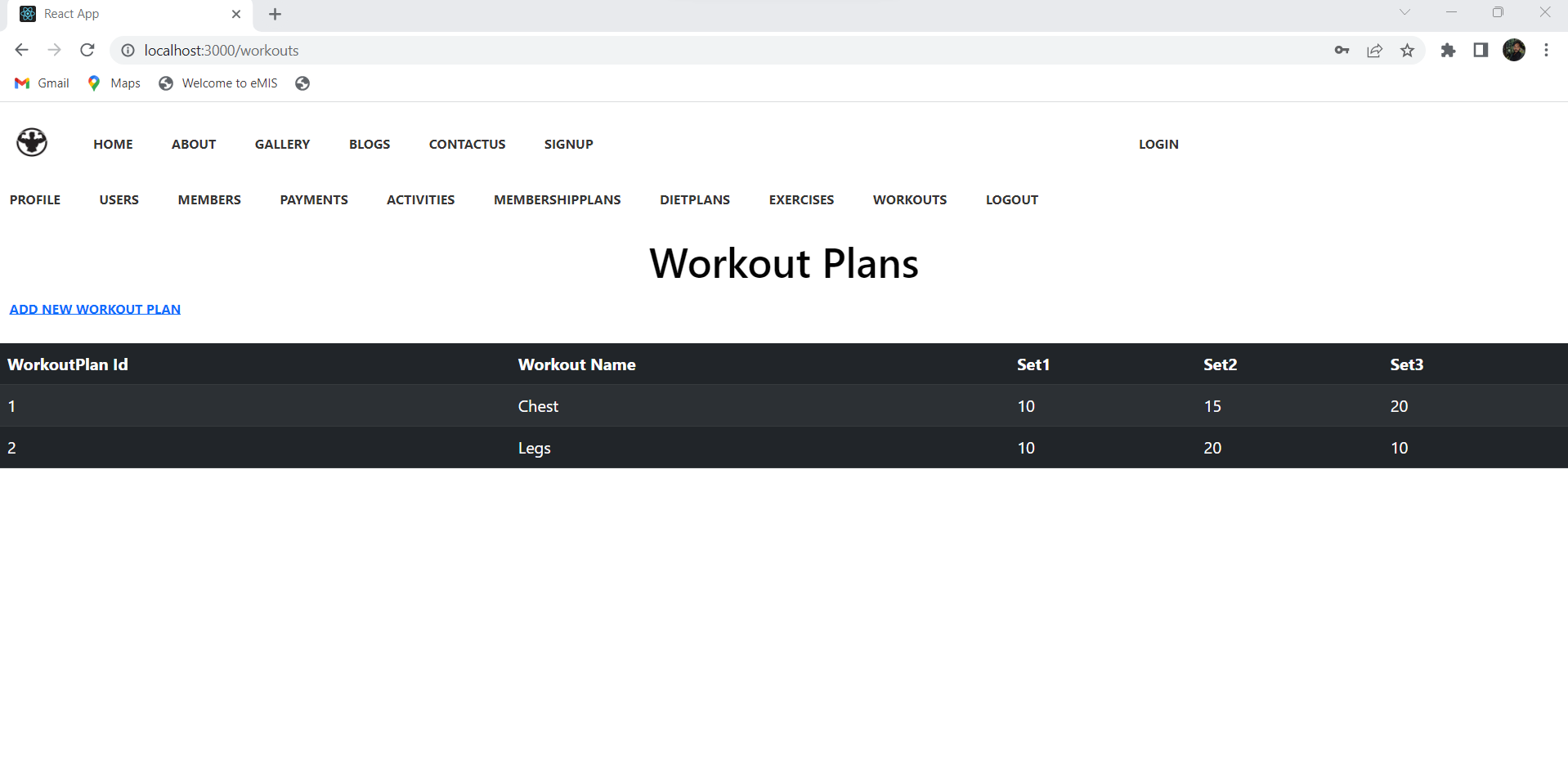
LogIn:



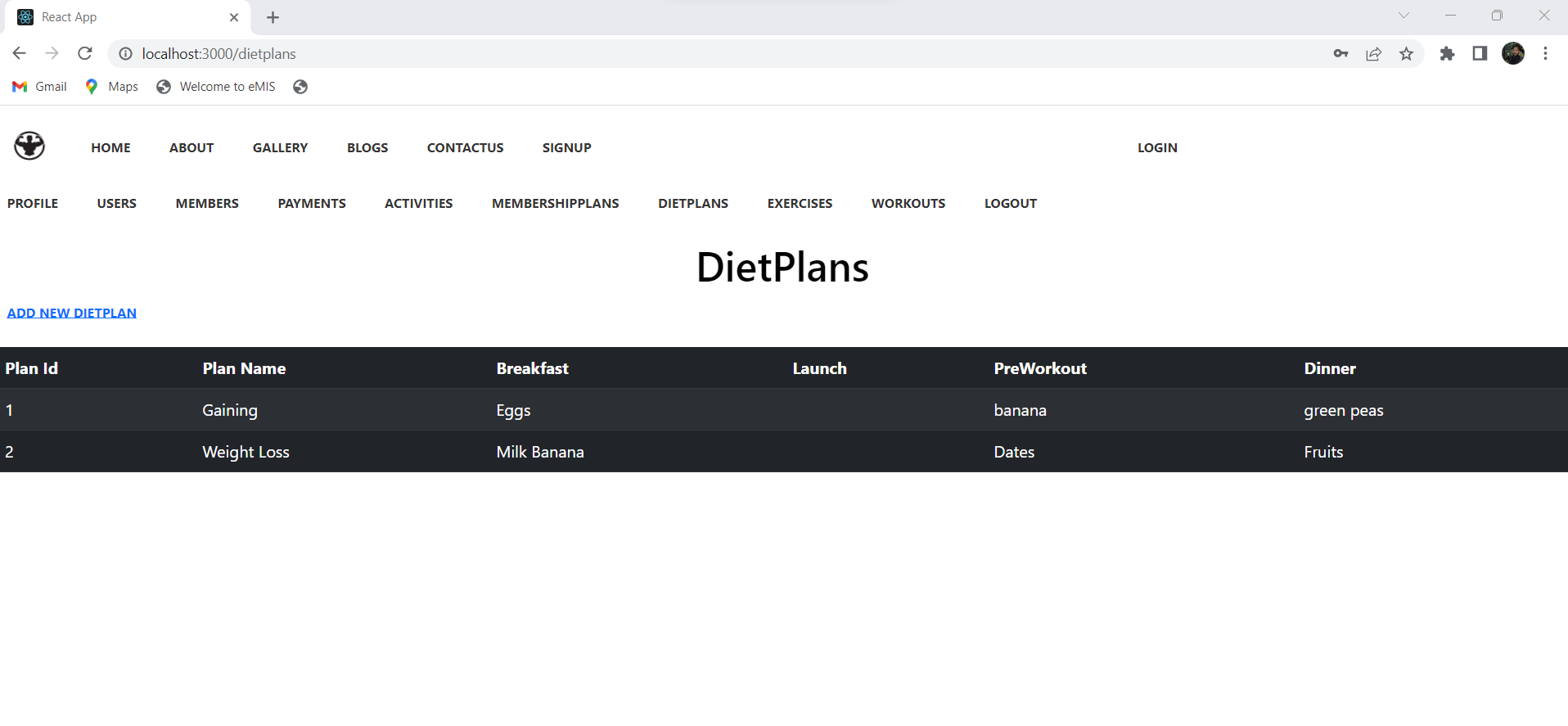
SignUp:



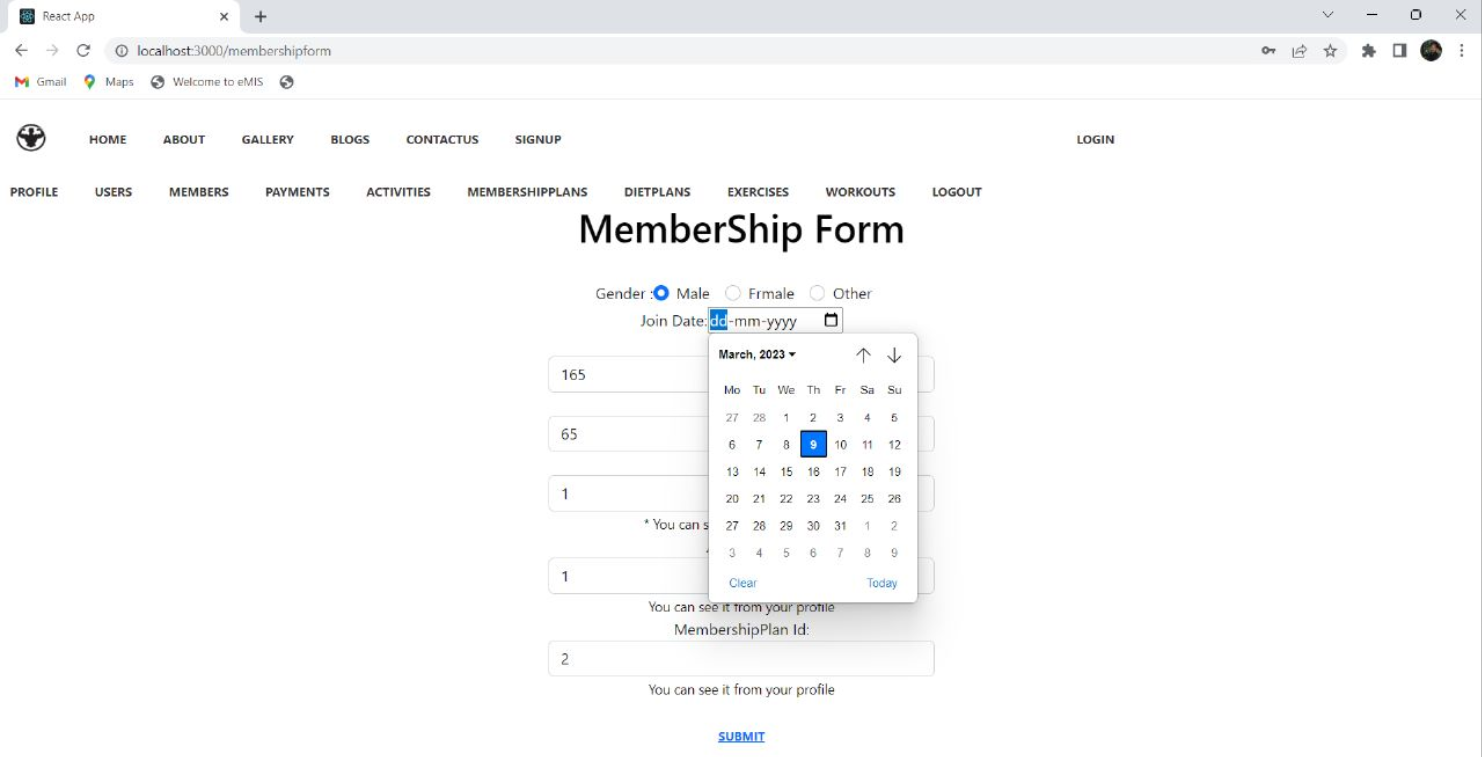
**Update/Add Workout Plan:**

****

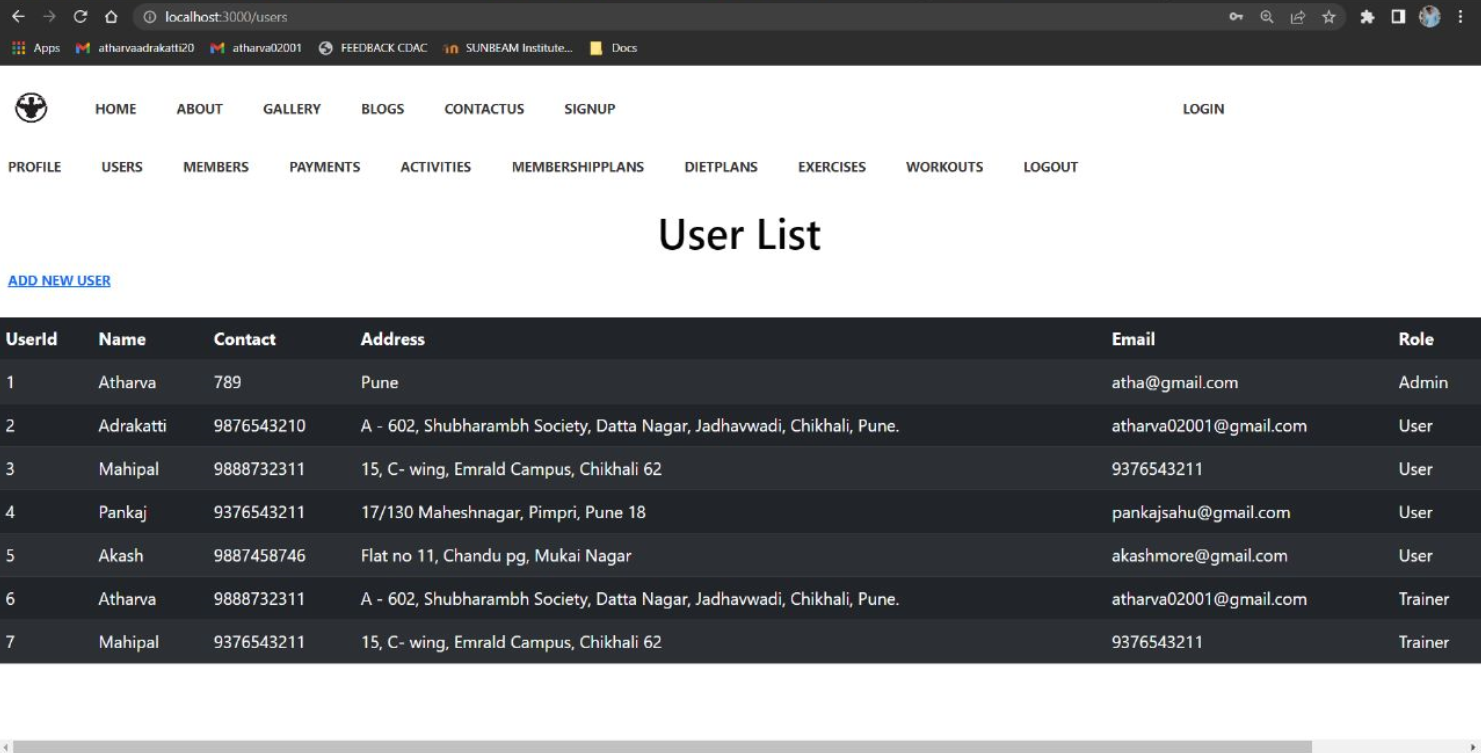
**Update/Add Diet Plan:**

****

**Membership Form:**

****

User List:



7.REFERENCES:

<http://www.google.com>

<https://www.getbootstrap.com>

<https://www.npmjs.com/>

<https://www.goldsgym.in/>

https://www.silverfitnessclub.in/

http://www.w3.org

http://www.wikipedia.org

<http://www.physcgym.com/>