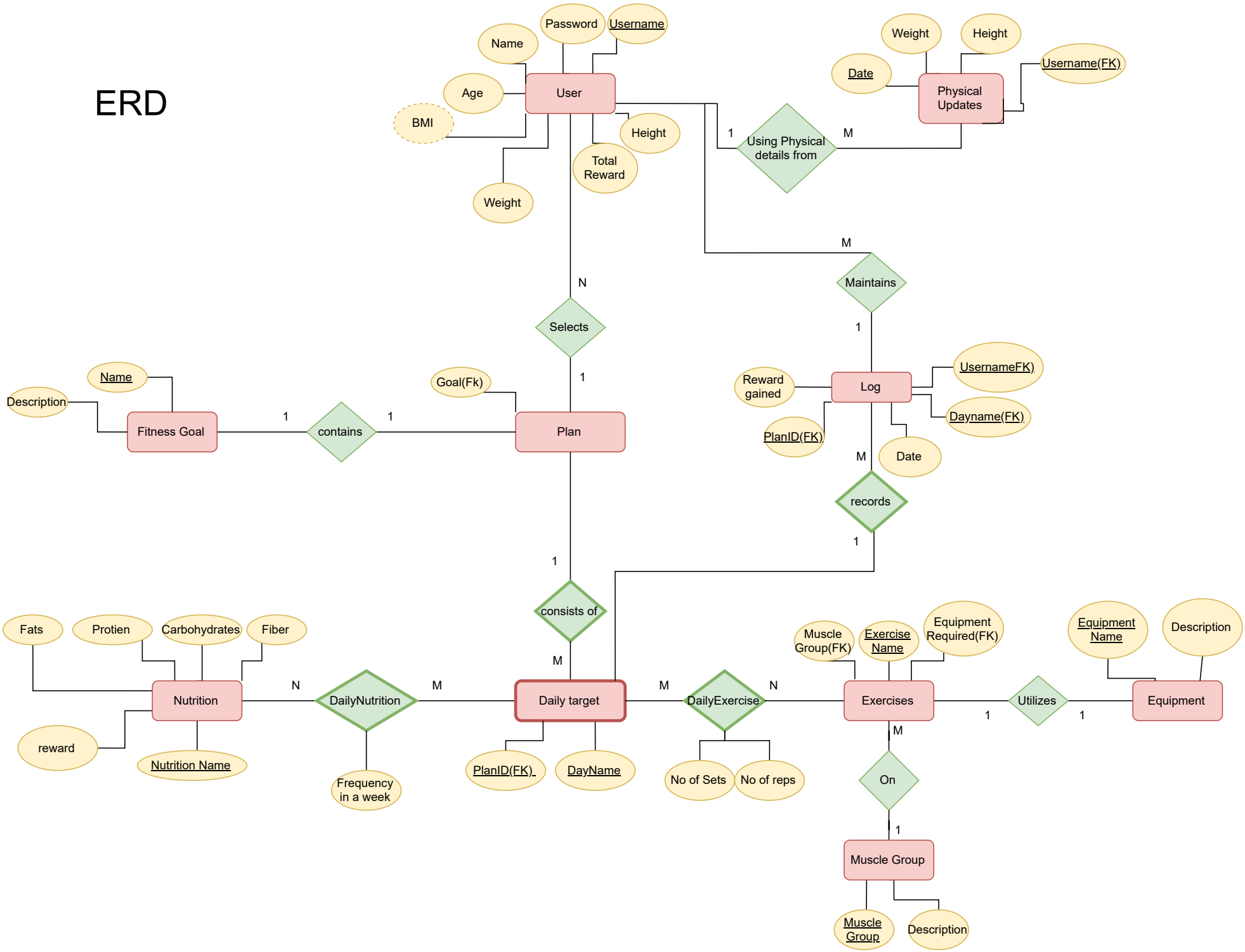

ERD
Password
Username
Name
Age
User
BMI
Height
Total Reward
Weight
1
Using Physical details from
M
Weight
Height
Date
Physical Updates
Username(FK)
M
Maintains
1
N
Selects
1
Name
Description
Fitness Goal
1
contains
1
Goal(Fk)
Plan
Reward gained
Log
UsernameFK)
Dayname(FK)
PlanID(FK)
Date
M
records
1
1
consists of
M
Fats
Protien
Carbohydrates
Fiber
Nutrition
reward
Nutrition Name
N
DailyNutrition
M
Frequency in a week
Daily target
PlanID(FK)
DayName
M
DailyExercise
N
No of Sets
No of reps
Muscle Group(FK)
Exercise Name
Equipment Required(FK)
Exercises
M
On
1
Muscle Group
Muscle Group
Description
1
Utilizes
1
Equipment Name
Description
Equipment

Ahmed Bilal i190580, Shazil Mehmood i190542, Sher Ali i192007 CSF

Relational Schema

(Note: Reward basically is the value we get from completing a daily target. Depending on the reward value gained by a user by completing daily targets, we can show the developments in weight, muscle, BMI.
The higher the reward, the higher the development.)

(Additional Note: PlanID and DayName are together a Primary key of the relation 'Daily Target'. Hence, they are both imported together as foriegn keys from Daily targets to Log).

Equipment

Equipment Name(PK)	Equipment Description

Muscle Group

Muscle group name(PK)	Description

Exercises

Exercise(PK)	Muscle Group(FK) FK(Muscle Group name) from Table 'Muscle Group'	Equipment (FK) FK(Equipment Name) from Table 'Equipment'	Reward

Nutrition

Nutrition Name(PK)	Reward Value	Fats	Proteins	Carbohydrates	Fiber

Goal

Goal Name(PK)	Goal Description

Plan

PlanID (PK)	Period	Goal (FK) FK (Goal Name) from Table 'Goal'

Daily Targets

PlanID (PK) FK (PlanID) from Table ‘Plan’	DayID (PK)

DailyNutrition (Nutrition)

PlanID(PK) FK (PlanID) from Table ‘DailyTargets’	DayName(PK) FK (DayName) from Table ‘Daily Targets’	Nutrition Name (PK) FK (Nutrition Name) from Table ‘Nutrition’	Quantity

DailyExercise (Exercise)

PlanID(PK) FK (PlanID) from Table ‘Daily Targets’	DayName(PK) FK (DayName) from Table ‘Daily Targets’	Exercise (PK) FK (Exercise) from Table ‘Exercise’	Number of sets	Number of Reps

User

Username (PK)	Password	Name	Age	Weight	Height	BMI	Total Reward Gained

Log

Username(PK) FK (Username) from Table ‘User’	PlanID (PK) FK (PlanID) from table ‘Daily Targets’	DayName (PK) FK (DayName) From Table ‘Daily Targets’	Date	Reward Update status

Table Description

Equipment
EquipmentName(PK) contains the name of the equipment e.g Bench, Barbell, Dumbbell and Description attribute contains a short description for the Equipment.

Muscle Group
MuscleGroup(PK) contains the Muscle Group Name e.g Shoulders, Legs, Abs and the Description attribute contains a short description of the anatomy of the body part e.g Left delt, right delt, rotator cuff for the shoulder.

Exercises
Exercise(PK) contains the Exercise Name e.g Squat, Lat Pulldown. Against each Exercise Name, we have the Muscle Group it works on, the equipment it uses (If any) and its reward value. Reward Value is basically the level of fitness gained from completing that Exercise once. It is later used to generate fitness progress reports.

Nutrition
Nutrition(PK) contains the name of the food e.g fish, whey protein, eggs. Against each food, we have its nutritional composition e.g amount of fats, protein, fiber and carbohydrates. Like exercise, each food has a reward value.

Goal
Goal(PK) contains the different goals that a plan can have e.g Weight Loss, Muscle Gain e.t.c. Along with it their is short description of the goal e.g for weight loss: cut your body fat and retain your muscle mass for a fitter body.

Plan
PlanID(PK) contains the ID of the Plan (surrogate key). Each Plan has a period (in weeks) so there is a period attribute. Each Plan has a goal e.g weight loss, muscle gain so there is a Goal attribute which is a FK from the table Goal.

Daily Targets
Each Plan has a Daily Target each day of the week. PlanID and DayName together make up the PK for this table e.g (P1, Tuesday).

Daily Nutrition (M:N Relationship between Daily Targets and Nutrition)
Imports DayName and PlanID as FK's from table 'Daily Targets' and imports Nutrition Name as FK from table 'Nutrition'. Together, these 3 make the PK and uniquely identify each tuple. The purpose of this table is to keep a record of daily nutrition Target. Quantity attribute keeps record of the quantity of the food. For Example: (P1, Tuesday, Eggs, 4).

Daily Exercise(M:N Relationship between Daily Targets and Exercise)
Same as 'Daily Nutrition', imports PlanID and DayName as FKs from Exercise. Along with Exercise Name (FK from table 'Exercise'), PlanID and DayName make the PK. There are also Attributes of 'Number of Sets' and 'Number of Reps'. An example tuple would be:
(P1,Tuesday, Pullups, 3, 4)

Log
PlanID, Dayname imported as FK's from Daily Targets. Username imported as FK from 'User' table.Together these 3 make the PK. There is also a timestamp (Date+Time) attribute. Lastly, there is a Reward attribute which tells us the reward gained by completing however much of the target the user has completed. And there is a reward update attribute for the purpose of confirming if the reward has been accounted into the predicted progress(Weight loss/Muscle Gain). Example tuple: (fitnessfreak, P1, Tuesday, 1.2, TRUE)

User
Username is PK.There is also a password column(candidate key). Rest of the attributes are weight, height, BMI(weak attribute derived from height and weight) and age. The 'Total Reward Gained' attribute tells us the total reward the user has gained up till now by completing daily targets. This total reward can be used to calculate a progress report.

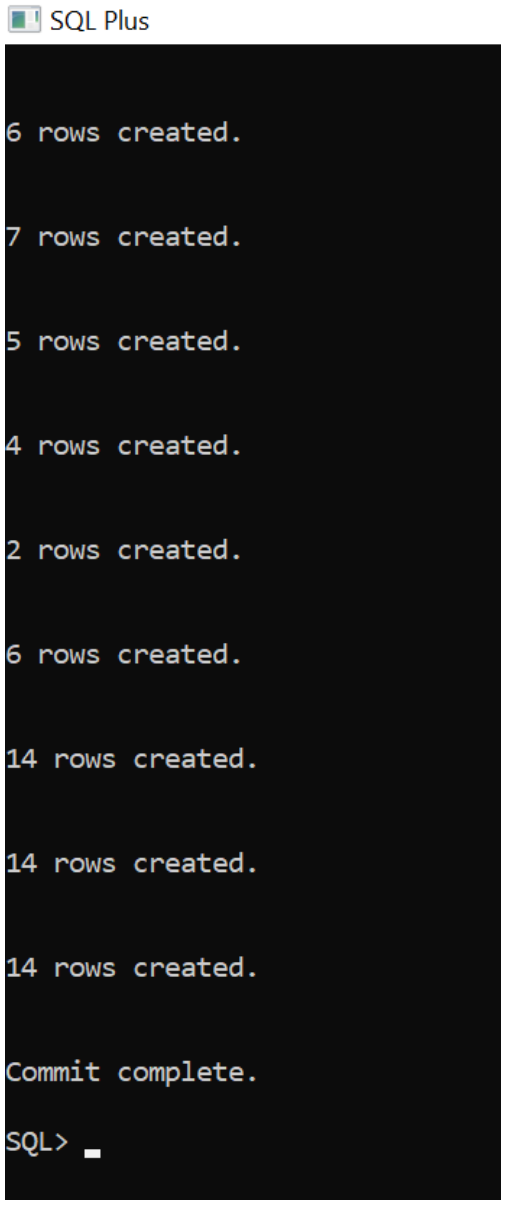
Database in SQL

SQL File: tables.sql (Submitted as one of the deliverables)
It contains the sql code to create tables and to insert tuples into each created table.

tables - Notepad
File Edit Format View Help

```
CREATE TABLE Nutrition
(Name varchar(30),
RewardValue number(3,2),
Fats number(3),
Protein number(3),
Carbohydrates number(3),
Fiber number(3),
PRIMARY KEY (Name));

CREATE TABLE Equipment
(EquipmentName varchar(15),
Description varchar(200),
PRIMARY KEY (EquipmentName));

CREATE TABLE MuscleGroup
(MuscleGroupName varchar(25),
Description varchar(200),
PRIMARY KEY (MuscleGroupName));


CREATE TABLE Exercise
(ExerciseName varchar(25),
RewardValue number(3,2),
MuscleGroupName varchar(25),
EquipmentRequired varchar(15),
PRIMARY KEY (ExerciseName),
FOREIGN KEY (MuscleGroupName) REFERENCES MuscleGroup (MuscleGroupName),
FOREIGN KEY (EquipmentRequired) REFERENCES Equipment (EquipmentName));


CREATE TABLE FitnessGoal
(Name varchar(25),
```

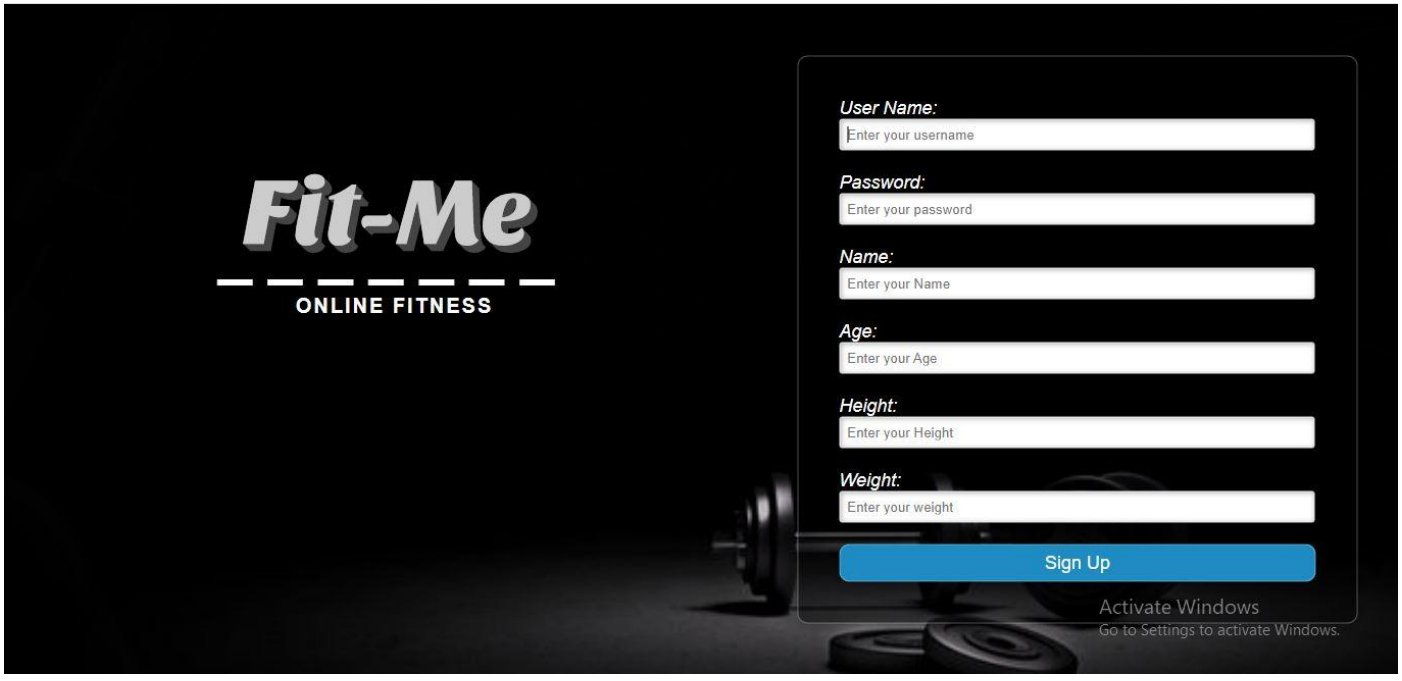
After Loading tables.sql into SQLplus:



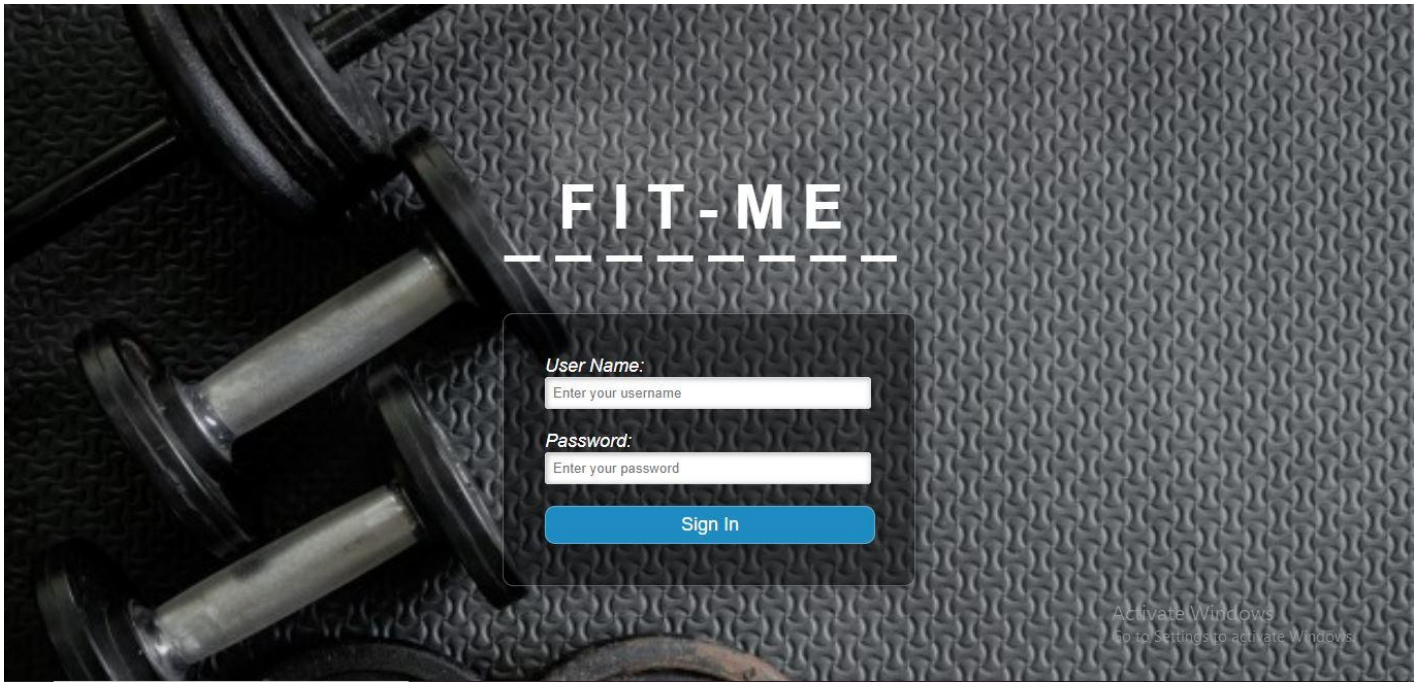
Website (Interface, forms, and reports)

1)The first and foremost interface. Allows user to create a new account or sign in using an already existing account.

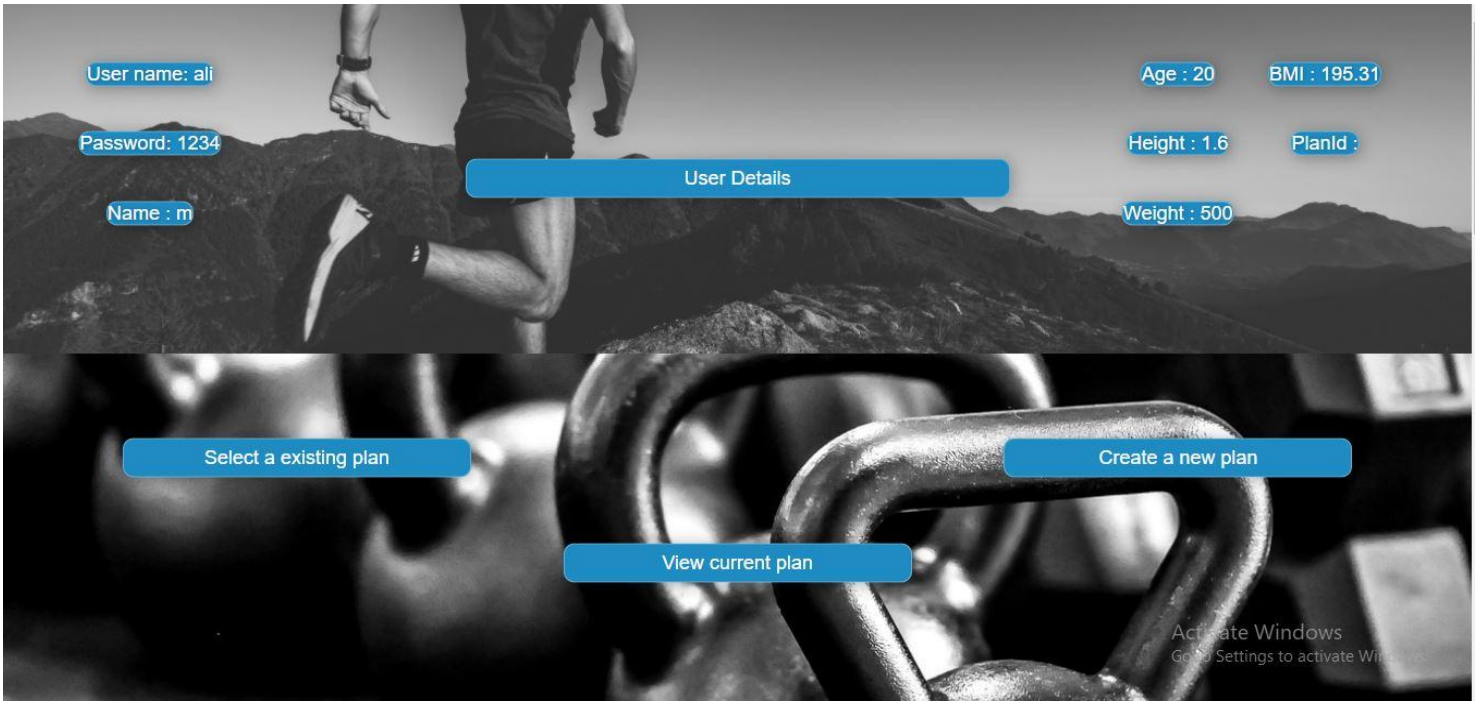
2.a) If '*Create New Account' is clicked*, php redirects to this interface.



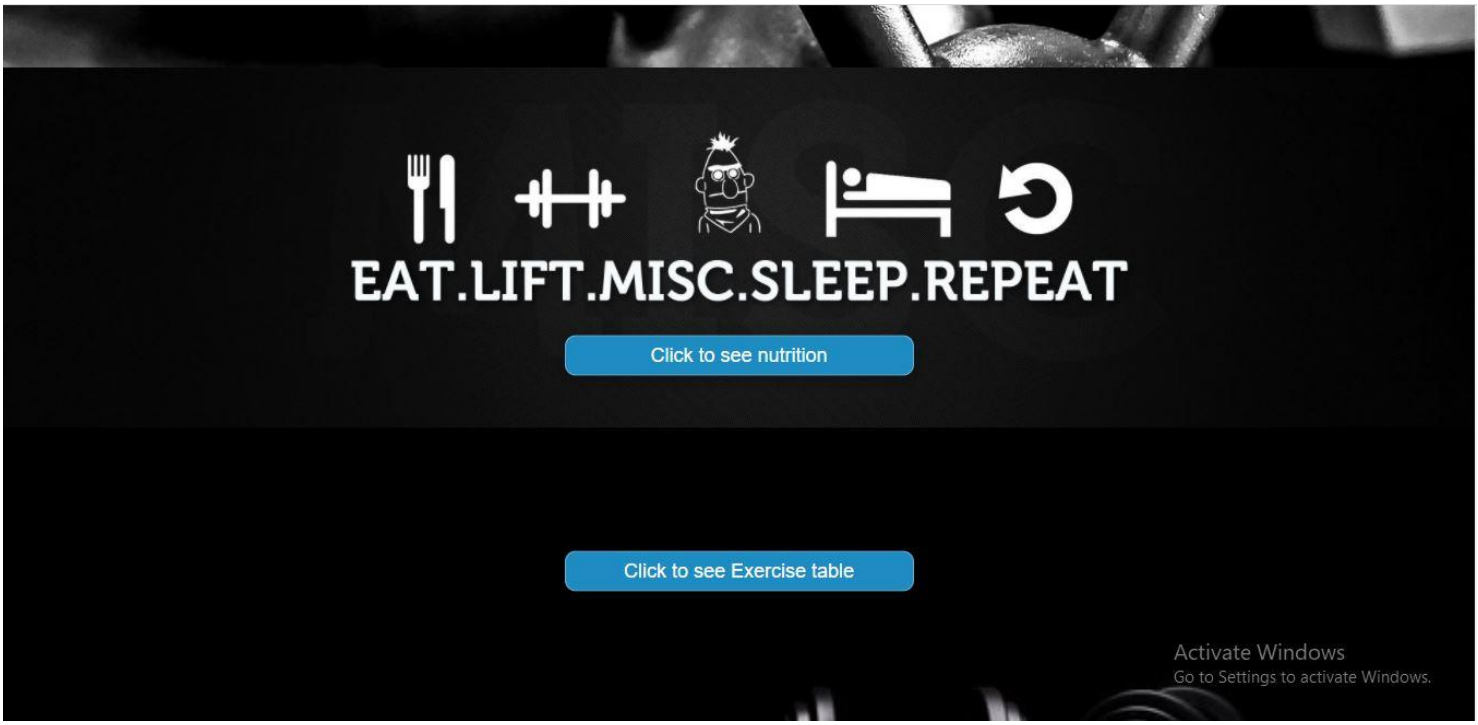
2.b) If '*Sign in*' is clicked, php redirects to the sign in interface.



3) Once a user signs in, he/she can see his/her main portfolio containing personal and physical details such as Name, Age, weight, height, BMI. There are also options to create a fitness plan, view existing (already made) fitness plans and see the current plan in which user is enrolled.

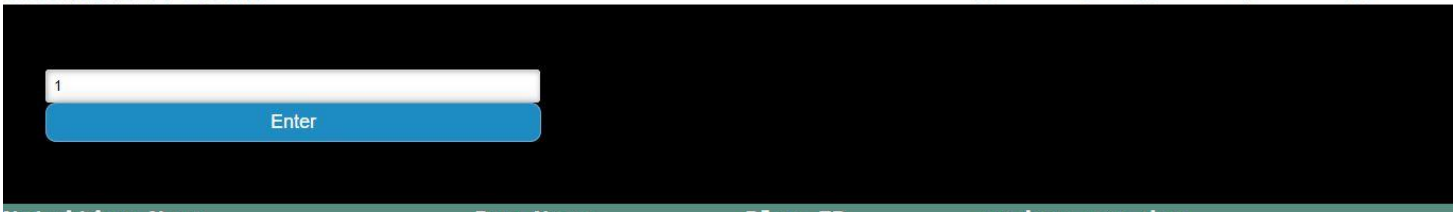


4) Scrolling down this same page, there are also other options such as the option to view and select different kinds of nutrition foods available for selection and the different exercises available for selection.



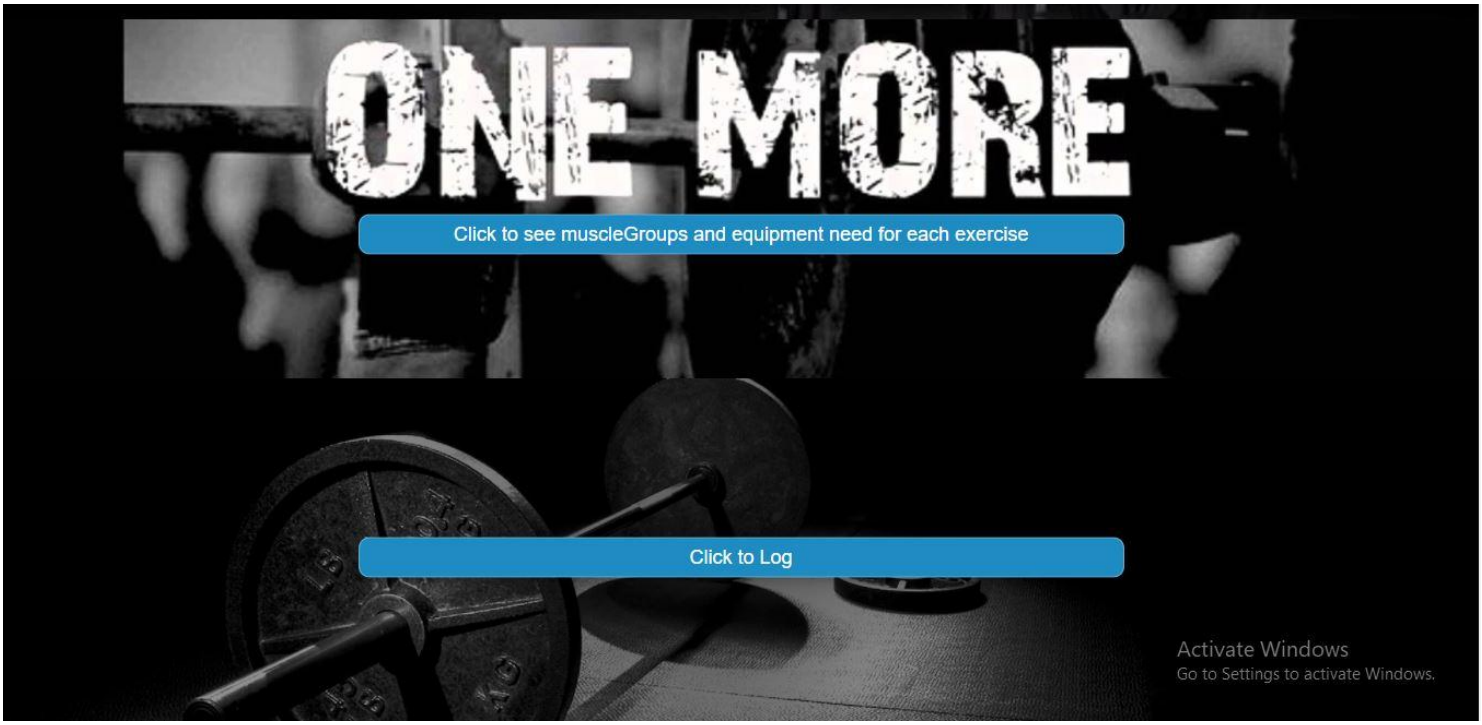
4.b) Clicking on ‘Click to see Nutrition’, we can see some nutrition foods (These can be select by user to add to his/her plan) as well along their nutritional composition.

Name	Reward	Fats	PROTEIN	CARBOHYDRATES			FIBER	
Eggs				.1	5	7	12	10
Meat				.4	4	27	9	15
Milk				.2	3	8	5	3
Cheese				.3	6	8	5	3
Essential Proteins				.7	0	10	4	13

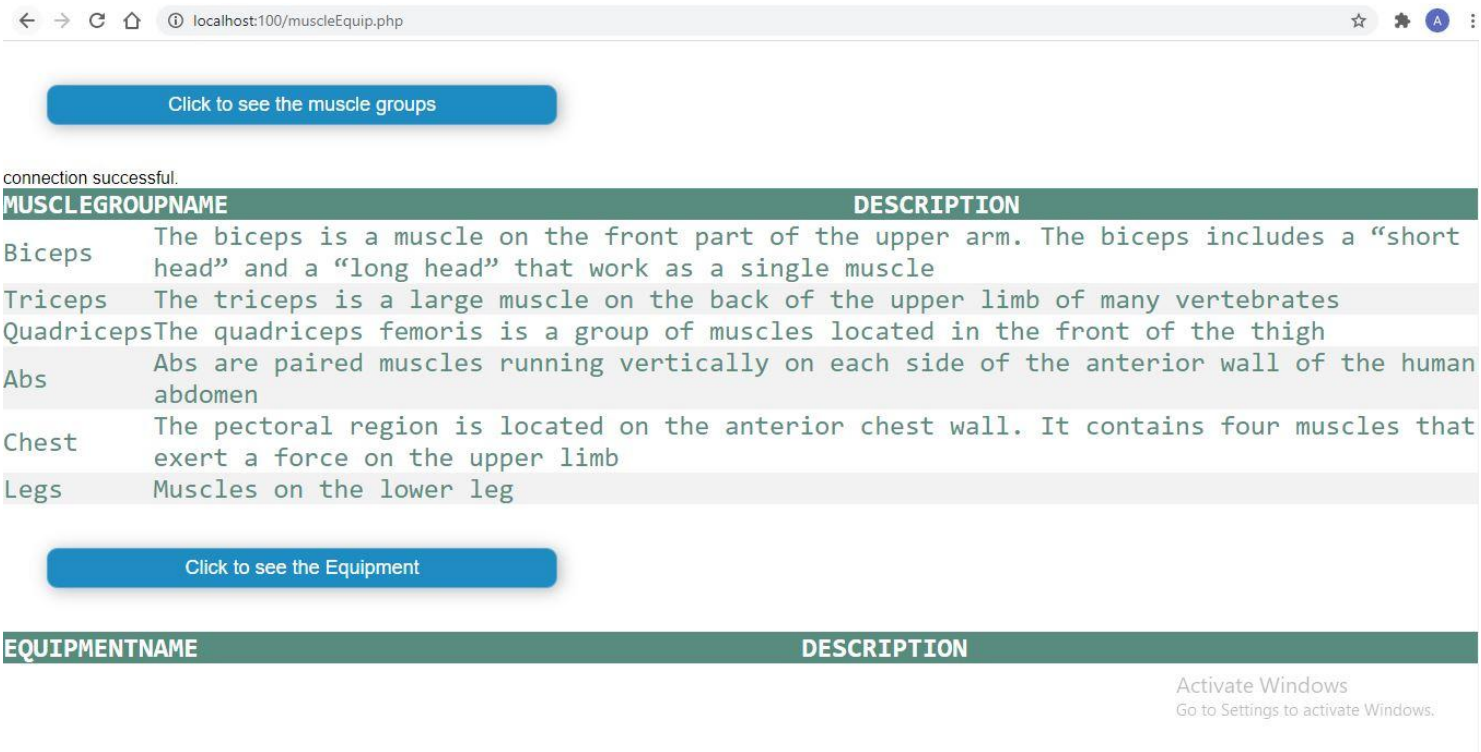


Nutrition Name	Day Name	Plan ID	number per day	
Eggs	Monday	1	3	
Meat	Tuesday	1	1	
Eggs	Wednesday	1	5	
Milk	Thrusday	1	2	
Cheese	Friday	1	1	
Eggs	Saturday	1	4	
Essential Proteins	Sunday	1	3	

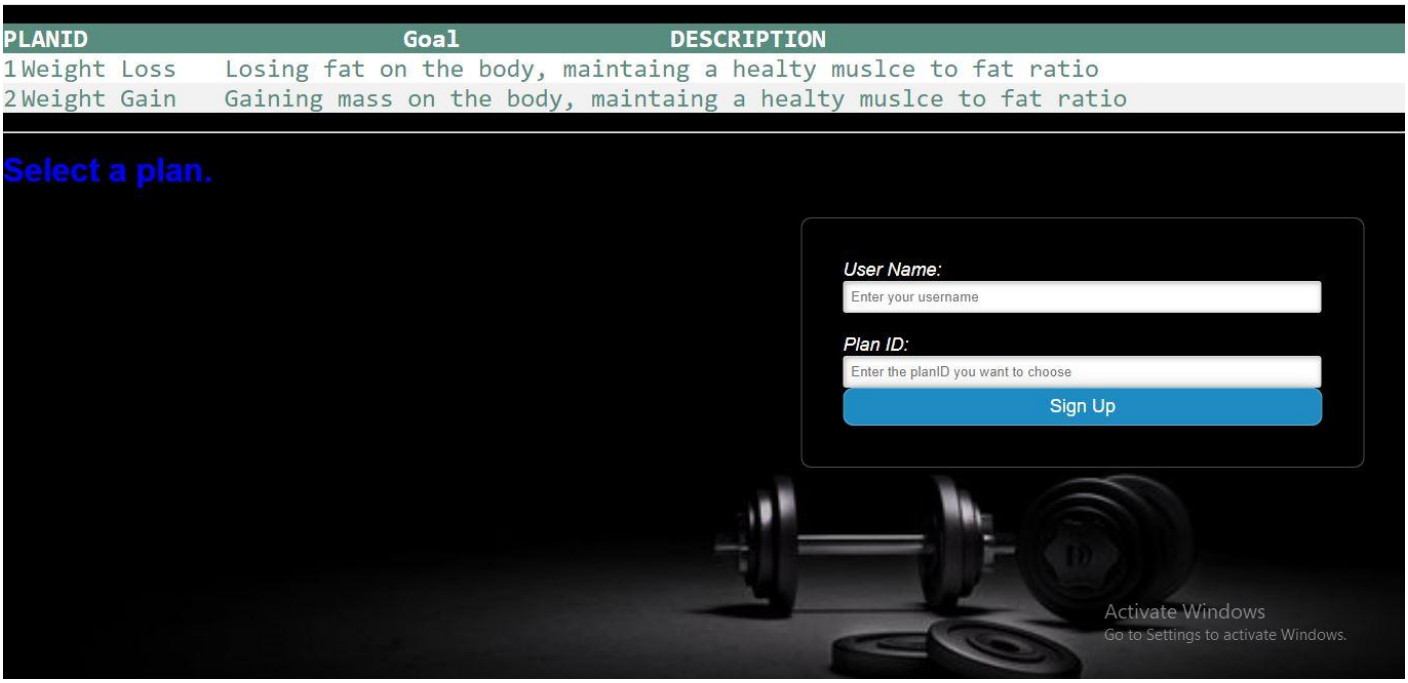
4.b) Further scrolling down, user has has the option (Button) to view information about muscle groups and types of equipment used for each exercise. Lastly, the user has the option to log his daily target completion (after being redirected to the log page) when he clicks the log button.



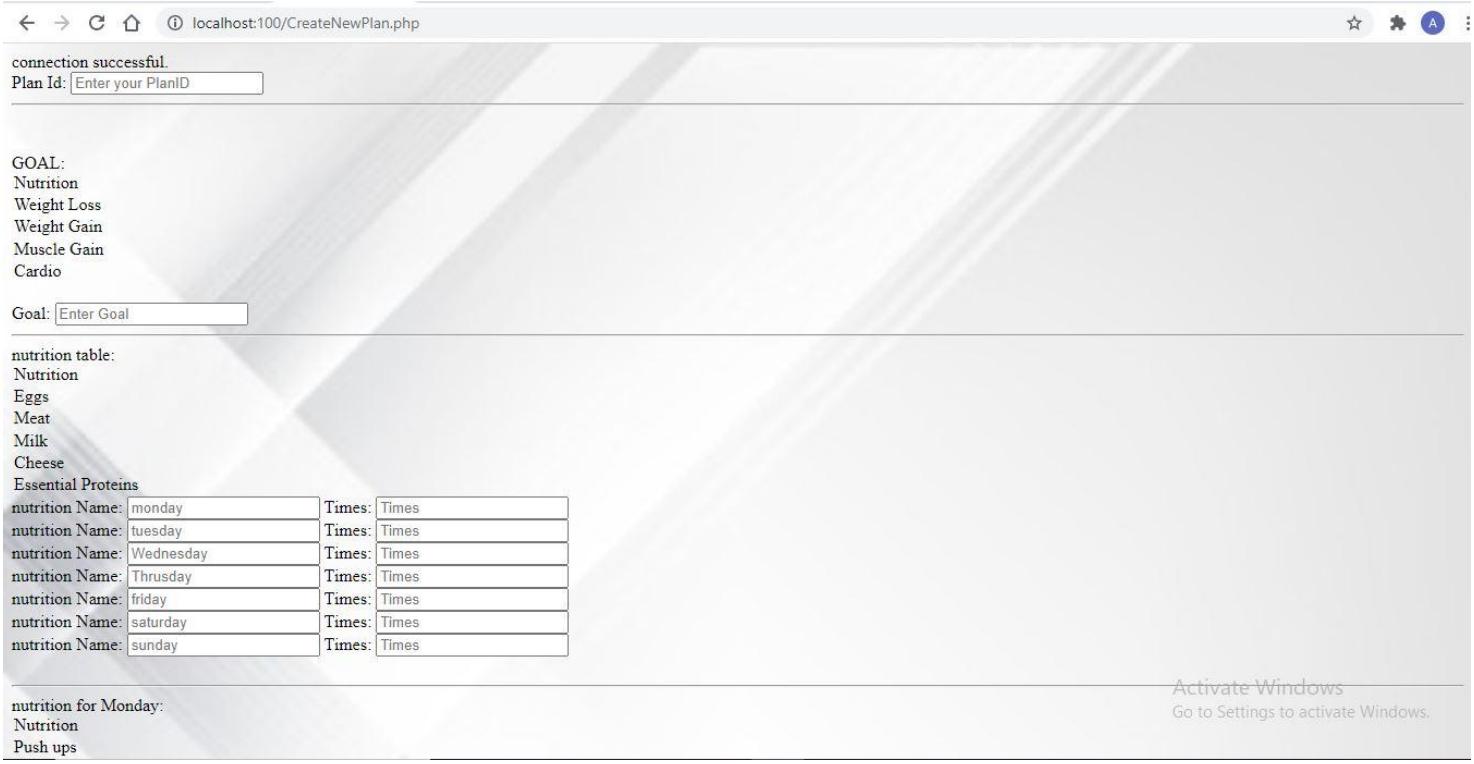
5)Here is how the muscle group description would show up by clicking first button. You can also choose to see equipment.



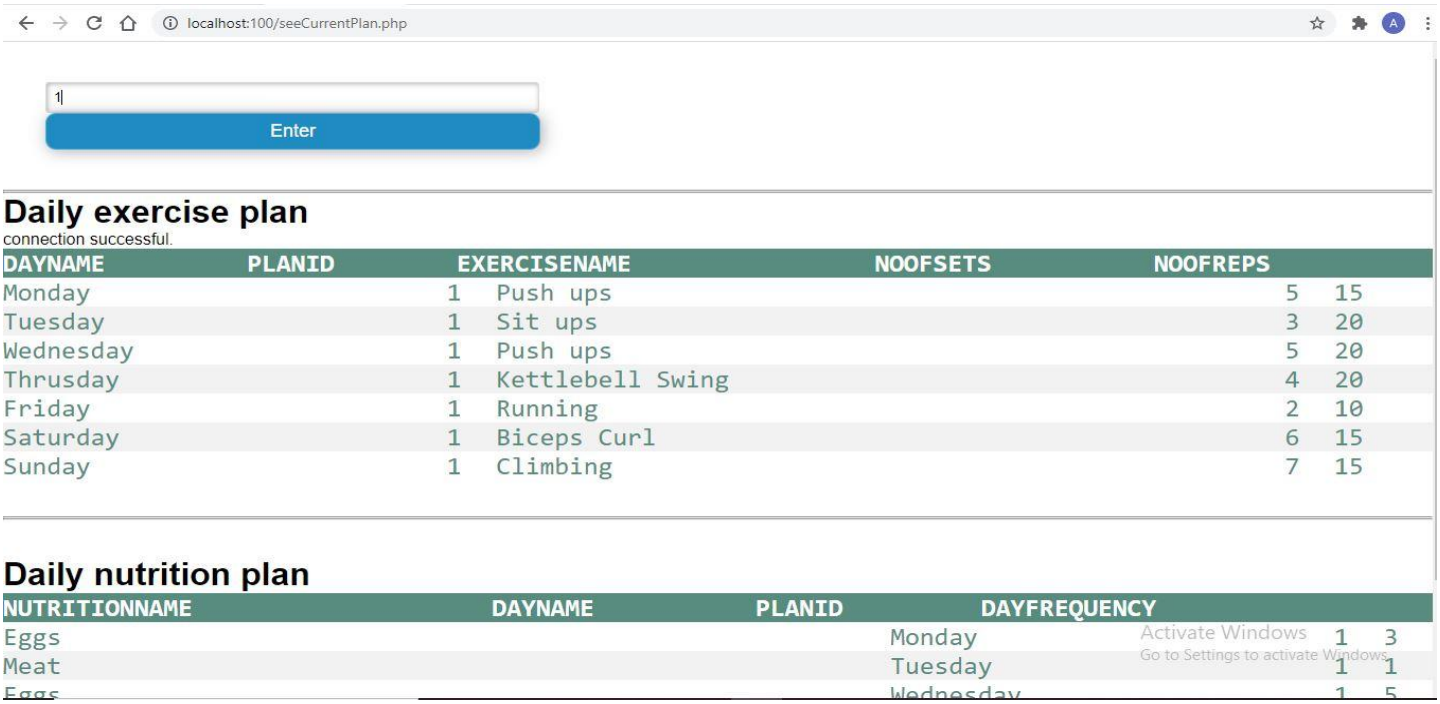
6)Here is the form for user to **select** an existing plan



7)Here's the form for user to **create** a plan. User has to choose a goal for his plan. Then he needs to select some nutritions from the list of nutritions available and also the quantity of each nutrition to be consumed (Daily).



8)After selecting exercises and nutrition, this report allows user to see his daily nutrition and exercise plans.



9)This is the daily target log form. It asks the user about the degree of completion of the daily target by asking about each of the mentioned tasks in the daily target he assigned to himself and based off of the percentage of completion of the whole target, generates a predicted fitness report.

PL/SQL usage

PL/SQL was used to update total reward for each user after a new entry corresponding to his name in the log table. A procedure was written that basically takes the sum of all the reward values corresponding to

a username, adds them and assigns them to the total reward attribute of the corresponding username in the user account table

The Procedure:

```
SQL>  create or replace procedure rewardsum(user IN varchar, totalreward OUT number)
  2  is
  3
  4  begin
  5  select sum(rewardgained) into totalreward from log where username = user;
  6
  7  end;
  8  /

Procedure created.

SQL>
SQL> create or replace procedure updateReward(user IN varchar)
  2  is
  3
  4  rew number;
  5  begin
  6  rewardsum(user, rew);
  7  update useraccountn set reward = rew where username = user;
  8
  9  end;
 10  /

Procedure created.
```

Successful Example:

```
SQL> select username, reward from useraccountn where username like '%shahid%';

USERNAME                REWARD
-------------------- ----------
shahid                       0

SQL> exec updateReward('shahid');

PL/SQL procedure successfully completed.

SQL> select username, reward from useraccountn where username like '%shahid%';

USERNAME                REWARD
-------------------- ----------
shahid                       3

SQL>
```

Final Note(IMPORTANT). We could find a double lined box to represent a weak entity and a double lined diamond for a weak relationship in the erd designing software we used (draw.io). Hence the weak entity in the erd (Daily targets) contains dark borders and so do its corresponding relationship diamonds.