DBMS Mini Project
Fitness Management System

Submitted By:
NISARGA BHASKAR
PES1UG20CS268
V Semester Section E

DBMS MINI PROJECT
FITNESS MANAGEMENT SYSTEM
NISARGA BHASKAR
PES1UG20CS268

Short Description and Scope of the Project

This project aims to automate gym and fitness admission process as the admission process in gyms and selecting a trainer is difficult.
Being healthy is the first thing to be kept in mind because most of the time our attitude depends on how we feel. Being healthy gives us the energy to work and do things.
This fitness management system will help to overcome such problems by booking the deciding the exercise routine online and pay fees by electronic money transfer.
In this fitness management system, there is 3 entity namely, Admin, Member, and Trainer. Admin can login using credentials. Admin can manage packages by adding cost, discount and deleting old membership types. Admin can manage member details by adding, updating and deleting. Admin can view the membership details of an individual member. Admin can also view the member's attendance taken by the trainer. Members can login using credentials. They can view their profile and list of trainers. They can also view the package and payment details. Members can give feedback on their trainers. They can make payments via card details. A trainer can log in using credentials. A trainer can set their profile. A trainer can take member's attendance daily, suggest the exercise routine for the given day.

DBMS MINI PROJECT
FITNESS MANAGEMENT SYSTEM
NISARGA BHASKAR
PES1UG20CS268

ER Diagram

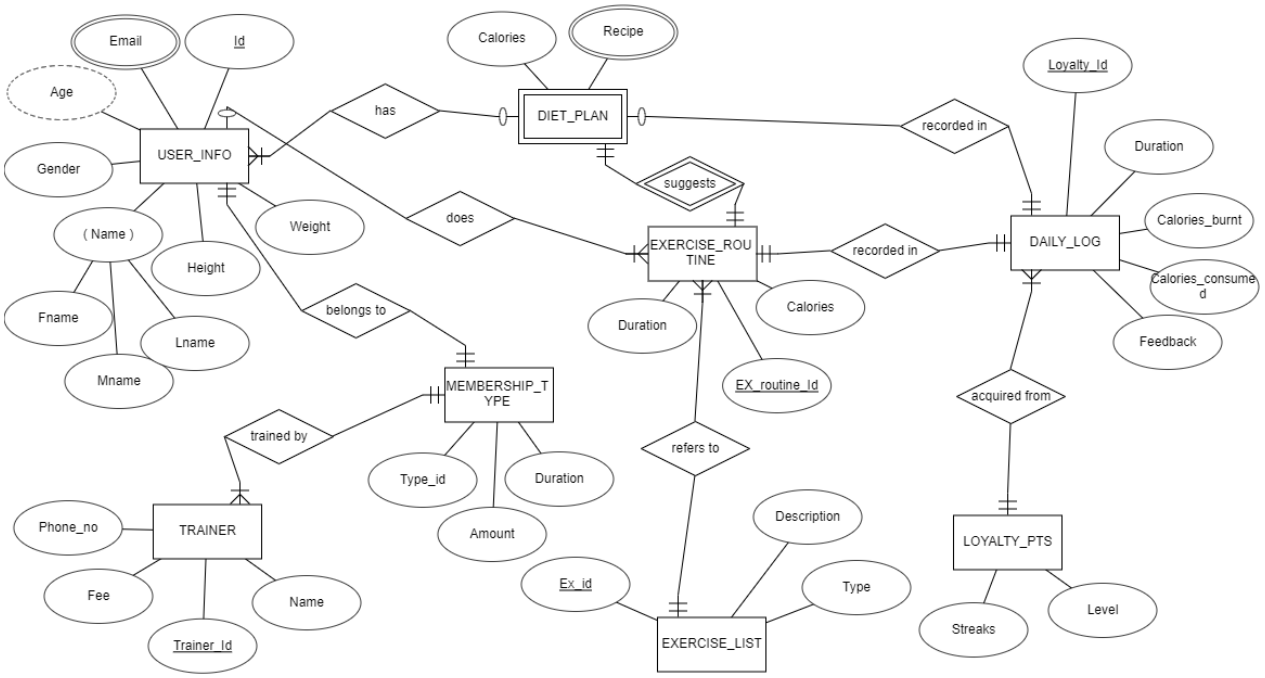


Figure 1: ER Diagram for the database System

Schema Diagram

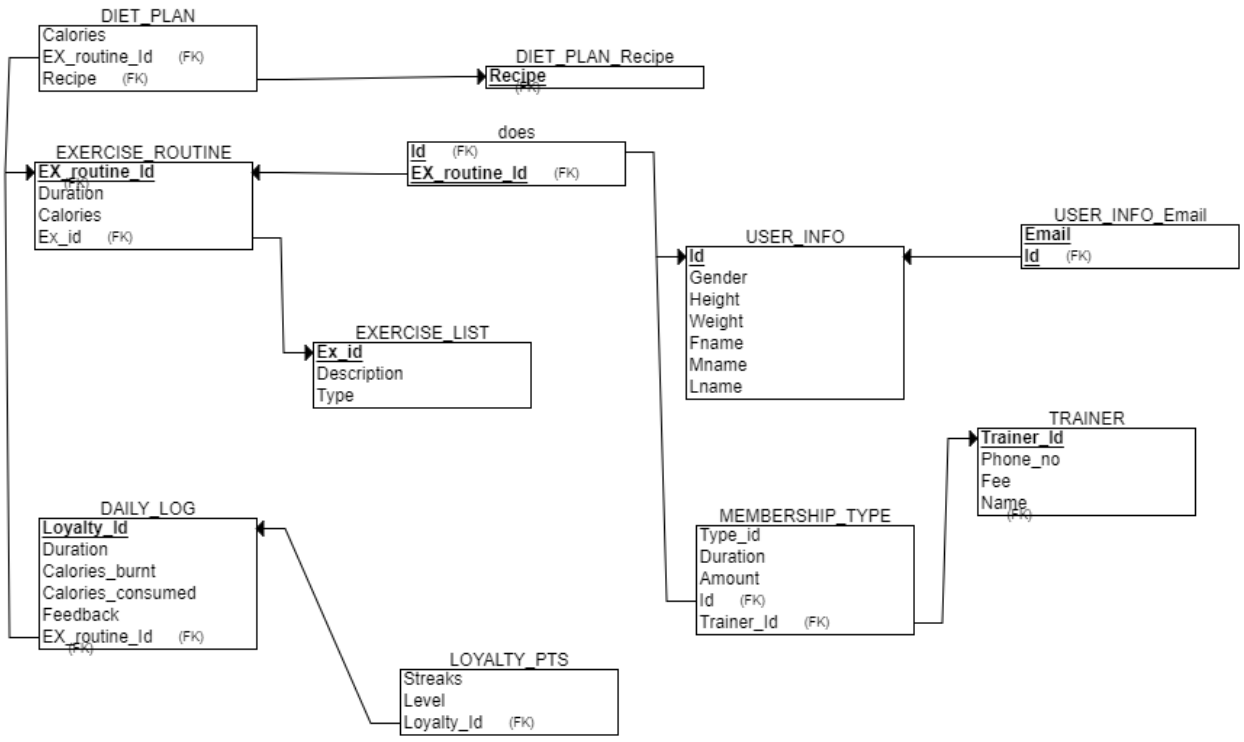


Figure 2: Relational Schema for the database

DDL statements – Building the database

```
CREATE DATABASE fitnessmanagement;
USE fitnessmanagement;

CREATE TABLE diet_plan (
 ex_id varchar(20) NOT NULL ,
 recipe varchar(30) DEFAULT NULL,
 calories int(10) DEFAULT NULL
);

CREATE TABLE exercise_routine (
 ex_routine_id varchar(20) NOT NULL,
 duration int NOT NULL,
 calories int(10) DEFAULT NULL,
 ex_id varchar(20) NOT NULL
);

CREATE TABLE daily_log (
 loyalty_id varchar(20) NOT NULL,
 duration int NOT NULL,
 calories_burnt int(10) DEFAULT NULL,
 calories_consumed int(10) DEFAULT NULL,
 feedback varchar(20) DEFAULT NULL,
 ex_routine_id varchar(20) NOT NULL
);

CREATE TABLE diet_plan_recipe (
 recipe varchar(30) DEFAULT NULL
);

CREATE TABLE does (
 id varchar(20) NOT NULL,
 ex_routine_id varchar(20) NOT NULL
);

CREATE TABLE exercise_list (
 ex_id varchar(20) NOT NULL ,
 description varchar(30) DEFAULT NULL,
```

```
 type varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE loyalty_pts (
 loyalty_id varchar(20) NOT NULL ,
 level int(3) NOT NULL,
 streaks int(10) NOT NULL DEFAULT 0
);

CREATE TABLE user_info(
 id varchar(20) NOT NULL,
 PRIMARY KEY(id),
 gender int(2) NOT NULL,
 height int NOT NULL,
 weight int NOT NULL,
 fname varchar(20) NOT NULL,
 mname varchar(20) DEFAULT NULL,
 lname varchar(20) DEFAULT NULL
);

CREATE TABLE membership_type (
 type_id varchar(20) NOT NULL,
 duration int NOT NULL,
 amount decimal(10,2) NOT NULL,
 id varchar(20) NOT NULL ,
 trainer_id varchar(20) NOT NULL
);

CREATE TABLE user_info_email (
 id varchar(20) NOT NULL ,
 email varchar(30) NOT NULL
);

CREATE TABLE trainer (
 trainer_id varchar(20) NOT NULL,
 PRIMARY KEY (trainer_id),
 phoneno int(10) NOT NULL,
 amount decimal(10,2) NOT NULL,
 name varchar(20) NOT NULL
);
```

```
ALTER TABLE diet_plan
 ADD KEY ex_id (ex_id),
 ADD KEY recipe (recipe);

ALTER TABLE exercise_routine
 ADD PRIMARY KEY (ex_routine_id),
 ADD KEY ex_id (ex_id);

ALTER TABLE daily_log
 ADD PRIMARY KEY (loyalty_id),
 ADD KEY ex_routine_id (ex_routine_id);

ALTER TABLE diet_plan_recipe
 ADD KEY recipe (recipe);

ALTER TABLE does
 ADD PRIMARY KEY (id,ex_routine_id),
 ADD KEY id (id),
 ADD KEY ex_routine_id (ex_routine_id);

ALTER TABLE exercise_list
 ADD PRIMARY KEY (ex_id);

ALTER TABLE loyalty_pts
 ADD KEY (loyalty_id);

ALTER TABLE membership_type
 ADD PRIMARY KEY (type_id),
 ADD KEY id (id),
 ADD KEY trainer_id (trainer_id);

ALTER TABLE user_info_email
 ADD PRIMARY KEY (id,email),
 ADD KEY id (id);

ALTER TABLE diet_plan
 ADD CONSTRAINT diet_plan_ibfk_1 FOREIGN KEY (ex_id) REFERENCES exercise_routine
(ex_routine_id);
```

```
ALTER TABLE diet_plan
 ADD CONSTRAINT diet_plan_ibfk_2 FOREIGN KEY (recipe) REFERENCES diet_plan_recipe (recipe);

ALTER TABLE exercise_routine
 ADD CONSTRAINT exercise_routine_ibfk_1 FOREIGN KEY (ex_id) REFERENCES exercise_list (ex_id);

ALTER TABLE does
 ADD CONSTRAINT does_ibfk_1 FOREIGN KEY (ex_routine_id) REFERENCES exercise_routine (ex_routine_id);

ALTER TABLE daily_log
  ADD CONSTRAINT daily_log_ibfk_1 FOREIGN KEY (ex_routine_id) REFERENCES exercise_routine (ex_routine_id);

ALTER TABLE loyalty_pts
 ADD CONSTRAINT loyalty_pts_ibfk_1 FOREIGN KEY (loyalty_id) REFERENCES daily_log (loyalty_id);

ALTER TABLE does
 ADD CONSTRAINT does_ibfk_2 FOREIGN KEY (id) REFERENCES user_info (id);

ALTER TABLE user_info_email
 ADD CONSTRAINT user_info_email_ibfk_1 FOREIGN KEY (id) REFERENCES user_info (id);

ALTER TABLE membership_type
 ADD CONSTRAINT membership_type_ibfk_1 FOREIGN KEY (id) REFERENCES user_info (id);

ALTER TABLE membership_type
 ADD CONSTRAINT membership_type_ibfk_2 FOREIGN KEY (trainer_id) REFERENCES trainer (trainer_id);
```

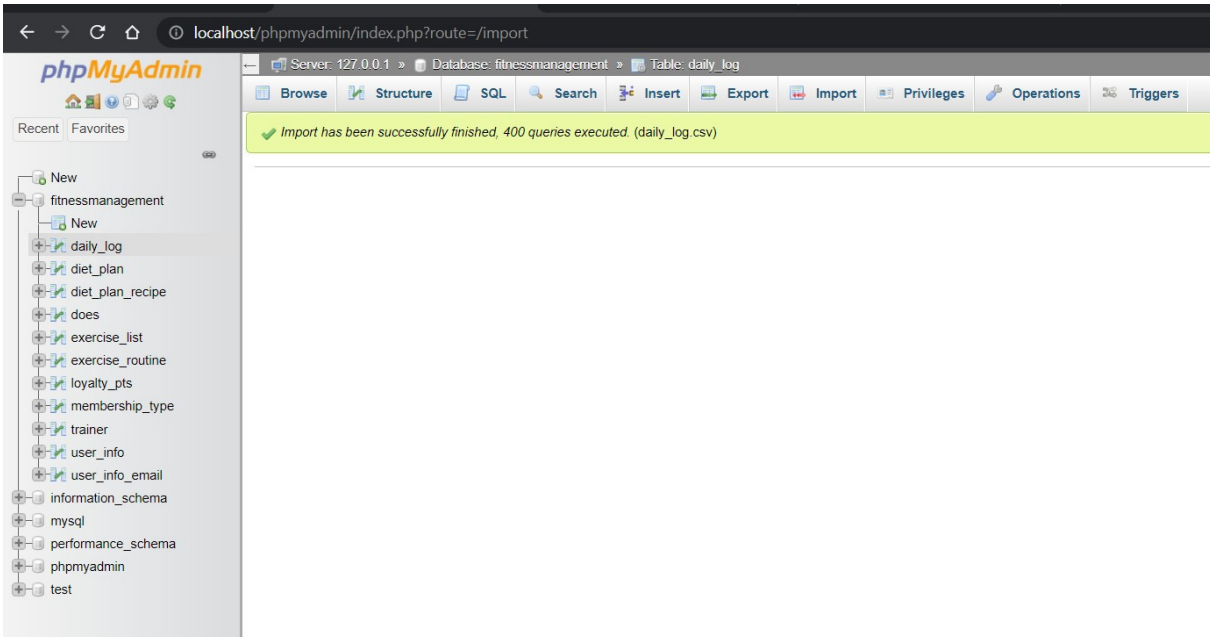
Populating the Database

The database is populated by two methods

- Using the insert command

```
MariaDB [fitnessmanagement]> insert into diet_plan_recipe values("Recipe0");
Query OK, 1 row affected (0.010 sec)
```

- Importing the CSV file in the PHP admin



Join Queries

1. Display the complete information of the user
   create view complete user info as select * from user_info ui natural join user_info_email;

```
MariaDB [fitnessmanagement]> select * from completeuserinfo;
+-------+--------+--------+--------+----------+----------+----------+-----------------------+
| id    | gender | height | weight | fname    | mname    | lname    | email                 |
+-------+--------+--------+--------+----------+----------+----------+-----------------------+
| USR12 |      0 |    178 |     94 | Fname41  | Mname200 | Lname223 | email.id.3@email.com  |
| USR12 |      0 |    178 |     94 | Fname41  | Mname200 | Lname223 | email.id.9@email.com  |
| USR13 |      0 |    173 |     56 | Fname150 | Mname190 | Lname248 | email.id.16@email.com |
| USR18 |      1 |    175 |     68 | Fname43  | Mname158 | Lname221 | email.id.32@email.com |
| USR19 |      2 |    193 |     76 | Fname175 | Mname98  | Lname23  | email.id.14@email.com |
| USR19 |      2 |    193 |     76 | Fname175 | Mname98  | Lname23  | email.id.31@email.com |
| USR19 |      2 |    193 |     76 | Fname175 | Mname98  | Lname23  | email.id.43@email.com |
```

2. Display the feedback for the session for each loyalty_id
   create view loyalty_pts_feedback as select dl.loyalty_id,feedback from daily_log dl join loyalty_pts lp on dl.loyalty_id = lp.loyalty_id;

```
MariaDB [fitnessmanagement]> select * from loyalty_pts_feedback;
+------------+------------+
| loyalty_id | feedback   |
+------------+------------+
| LYL113     | Feedback22 |
| LYL116     | Feedback50 |
| LYL119     | Feedback77 |
| LYL122     | Feedback2  |
| LYL126     | Feedback15 |
| LYL129     | Feedback63 |
| LYL134     | Feedback1  |
| LYL135     | Feedback95 |
```

3. Display the exercise routine followed for a given day
   create view exercise_routine_dailylog as select * from exercise_routine natural join daily_log;

```
MariaDB [fitnessmanagement]> select * from exercise_routine_dailylog;
+---------------+----------+----------+---------+------------+---------------+------------------+------------+
| ex_routine_id | duration | calories | ex_id   | loyalty_id | calories_burnt | calories_consumed | feedback   |
+---------------+----------+----------+---------+------------+---------------+------------------+------------+
| EX_routine28  |       47 |      181 | EX_ID34 | LYL190     |            100 |             1455 | Feedback54 |
| EX_routine30  |       47 |      161 | EX_ID44 | LYL293     |            152 |             1241 | Feedback6  |
| EX_routine44  |       83 |       64 | EX_ID68 | LYL7       |            275 |             2169 | Feedback47 |
| EX_routine47  |       45 |      320 | EX_ID91 | LYL391     |            209 |             1763 | Feedback32 |
| EX_routine57  |       75 |      189 | EX_ID44 | LYL252     |            273 |             1389 | Feedback71 |
| EX_routine61  |       89 |      387 | EX_ID55 | LYL67      |            276 |             1030 | Feedback42 |
| EX_routine79  |       65 |       58 | EX_ID71 | LYL233     |            148 |             1828 | Feedback52 |
+---------------+----------+----------+---------+------------+---------------+------------------+------------+
7 rows in set (0.012 sec)
```

4. Display the exercise routine followed by a each user
   create view exercise_routine_fname as select d.ex_routine_id,ui.fname from user_info ui join does d on d.id = ui.id;

```
MariaDB [fitnessmanagement]> select * from exercise_routine_fname;
+---------------+----------+
| ex_routine_id | fname    |
+---------------+----------+
| EX_routine52  | Fname168 |
| EX_routine60  | Fname168 |
| EX_routine78  | Fname168 |
| EX_routine43  | Fname97  |
| EX_routine72  | Fname43  |
| EX_routine36  | Fname175 |
| EX_routine59  | Fname175 |
| EX_routine29  | Fname125 |
| EX_routine82  | Fname5   |
| EX_routine23  | Fname72  |
| EX_routine56  | Fname72  |
| EX_routine93  | Fname72  |
```

Aggregate Functions

1. Display the no of users a trainer is currently training
   create view no_of_user_per_trainer as select trainer_id,count(*) as no_of_user_per_trainer from membership_type group by trainer_id;

```
MariaDB [fitnessmanagement]> select * from no_of_user_per_trainer;
+------------+------------------------+
| trainer_id | no_of_user_per_trainer |
+------------+------------------------+
| TRAINER10  |                      1 |
| TRAINER11  |                      1 |
| TRAINER20  |                      1 |
| TRAINER3   |                      1 |
| TRAINER7   |                      1 |
+------------+------------------------+
5 rows in set (0.005 sec)
```

2. Display the maximum calories burnt
   select max(calories_burnt) from exercise_routine_dailylog;

```
MariaDB [fitnessmanagement]> select max(calories_burnt) from exercise_routine_dailylog;
+--------------------+
| max(calories_burnt) |
+--------------------+
|                276 |
+--------------------+
1 row in set (0.003 sec)
```

3. Display the minimum calories consumed
   select min(calories_consumed) from exercise_routine_dailylog;

```
MariaDB [fitnessmanagement]> select min(calories_consumed) from exercise_routine_dailylog;
+-------------------------+
| min(calories_consumed) |
+-------------------------+
|                    1030 |
+-------------------------+
1 row in set (0.003 sec)
```

4. Display the average calories burnt for a given duration
   select avg(calories_burnt) as calories_burnt,duration from exercise_routine_dailylog group by duration;

```
MariaDB [fitnessmanagement]> select avg(calories_burnt) as calories_burnt,duration from exercise_routine_dailylog group by duration;
+----------------+----------+
| calories_burnt | duration |
+----------------+----------+
|       209.0000 |       45 |
|       126.0000 |       47 |
|       148.0000 |       65 |
|       273.0000 |       75 |
|       275.0000 |       83 |
|       276.0000 |       89 |
+----------------+----------+
6 rows in set (0.011 sec)
```

Set Operations

1. Display userid of the users under trainer 3 or trainer 7
   create view userid_of_trainer_3and7 as select id from membership_type where trainer_id = "TRAINER3" union select id from membership_type where trainer_id = "TRAINER7";

```
MariaDB [fitnessmanagement]> select * from userid_of_trainer_3and7;
+-------+
| id    |
+-------+
| USR27 |
| USR3  |
+-------+
2 rows in set (0.014 sec)
```

2. Display the userid who have done exercise routine 52 and 93
   select fname from exercise_routine_fname where ex_routine_id="EX_routine52"and exists (select fname from exercise_routine_fname where ex_routine_id="EX_routine93");

```
MariaDB [fitnessmanagement]> select fname from exercise_routine_fname where ex_routine_id="EX_routine52"and exists (select fname from exercise_routine_fname where ex_routine_id="EX_routine93");
+----------+
| fname    |
+----------+
| Fname168 |
| Fname110 |
+----------+
2 rows in set (0.002 sec)
```

3. Display the userid who have done exercise routine 29 and exercise routine 39
   select fname from exercise_routine_fname where ex_routine_id="EX_routine39"and exists (select fname from

exercise_routine_fname where ex_routine_id="EX_routine29");

```
MariaDB [fitnessmanagement]> select fname from exercise_routine_fname where ex_routine_id="EX_routine39"and exists (select fname from exercise_routine_fname where ex_routine_id="EX_routine29");
+----------+
| fname    |
+----------+
| Fname167 |
+----------+
1 row in set (0.002 sec)
```

4. Display exercise id which belong to exercise routine 52 or exercise routine 93

   select ex_id from exercise_routine where ex_routine_id="EX_routine52" union (select ex_id from exercise_routine where ex_routine_id="EX_routine93");

```
MariaDB [fitnessmanagement]> select ex_id from exercise_routine where ex_routine_id="EX_routine52" union (select ex_id from exercise_routine where ex_routine_id="EX_routine93");
+----------+
| ex_id    |
+----------+
| EX_ID120 |
| EX_ID53  |
+----------+
2 rows in set (0.004 sec)
```

Functions and Procedures

A function to correct the gender as some of the gender values where not acceptable

A procedure to call the above function

```
MariaDB [fitnessmanagement]> DELIMITER $$ ;
MariaDB [fitnessmanagement]> CREATE FUNCTION correcting_gender(GENDER INT)
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->
    ->    DECLARE gender INT;
    ->    SET gender = GENDER ;
    ->    IF gender > 1 THEN
    ->    SET gender = 1;
    ->    ELSE
    ->    SET gender = 0;
    ->    END IF;
    -> RETURN gender;
    ->
    -> END; $$
Query OK, 0 rows affected (0.007 sec)

MariaDB [fitnessmanagement]> DELIMITER ;
MariaDB [fitnessmanagement]>
MariaDB [fitnessmanagement]>
MariaDB [fitnessmanagement]> DELIMITER $$
MariaDB [fitnessmanagement]> CREATE procedure gender_updation()
    -> BEGIN
    -> UPDATE user_info
    -> SET gender = correcting_gender(gender);
    -> END;$$
Query OK, 0 rows affected (0.003 sec)

MariaDB [fitnessmanagement]> DELIMITER ;
MariaDB [fitnessmanagement]> call gender_updation();
Query OK, 59 rows affected (0.010 sec)
```

```
MariaDB [fitnessmanagement]> select * from user_info;
+-------+--------+--------+--------+----------+----------+----------+
| id    | gender | height | weight | fname    | mname    | lname    |
+-------+--------+--------+--------+----------+----------+----------+
| USR0  |      0 |    185 |     62 | Fname21  | Mname72  | Lname70  |
| USR1  |      0 |    177 |     49 | Fname93  | Mname37  | Lname92  |
| USR10 |      1 |    194 |     78 | Fname80  | Mname223 | Lname29  |
| USR11 |      0 |    163 |     45 | Fname166 | Mname51  | Lname116 |
| USR12 |      1 |    178 |     94 | Fname41  | Mname200 | Lname223 |
| USR13 |      0 |    173 |     56 | Fname150 | Mname190 | Lname248 |
| USR14 |      0 |    155 |     95 | Fname39  | Mname152 | Lname28  |
| USR15 |      0 |    163 |     48 | Fname168 | Mname64  | Lname197 |
| USR16 |      0 |    181 |     66 | Fname97  | Mname104 | Lname178 |
| USR17 |      0 |    167 |     83 | Fname198 | Mname173 | Lname36  |
| USR18 |      0 |    175 |     68 | Fname43  | Mname158 | Lname221 |
| USR19 |      0 |    193 |     76 | Fname175 | Mname98  | Lname23  |
| USR2  |      1 |    166 |     65 | Fname144 | Mname10  | Lname156 |
| USR20 |      1 |    176 |     77 | Fname20  | Mname20  | Lname64  |
| USR21 |      1 |    172 |     86 | Fname44  | Mname121 | Lname111 |
| USR22 |      1 |    182 |     82 | Fname125 | Mname248 | Lname131 |
| USR23 |      0 |    193 |     58 | Fname5   | Mname29  | Lname104 |
```

Triggers and Cursors

A trigger is created if a trainer has more than 4 users under him

```
MariaDB [fitnessmanagement]> DELIMITER $$
MariaDB [fitnessmanagement]> CREATE TRIGGER total_no_of_user_per_trainer_exceed_check
    -> BEFORE INSERT
    -> ON membership_type FOR EACH ROW
    -> BEGIN
    ->     DECLARE error_msg VARCHAR(255);
    ->     SET error_msg = (' total number of users per trainer exceeds more than 4');
    ->     IF (select count(*) from membership_type where trainer_id=new.trainer_id group by trainer_id) > 4 THEN
    ->     SIGNAL SQLSTATE '45000'
    ->     SET MESSAGE_TEXT = error_msg;
    ->     END IF;
    -> END $$
Query OK, 0 rows affected (0.008 sec)

MariaDB [fitnessmanagement]> DELIMITER ;
MariaDB [fitnessmanagement]> insert into membership_type values ("MEM09",4,6745,"USR75","TRAINER3");
ERROR 1644 (45000):  total number of users per trainer exceeds more than 4
MariaDB [fitnessmanagement]>
```

A cursor for displaying the emailing list

```
MariaDB [fitnessmanagement]> CREATE PROCEDURE createEmailList (
    -> INOUT emailList varchar(4000)
    -> )
    -> BEGIN
    -> DECLARE finished INTEGER DEFAULT 0;
    -> DECLARE emailAddress varchar(100) DEFAULT "";
    ->
    ->
    -> DEClARE curEmail
    -> CURSOR FOR
    -> SELECT email FROM user_info_email;
    ->
    ->
    -> DECLARE CONTINUE HANDLER
    ->         FOR NOT FOUND SET finished = 1;
    ->
    -> OPEN curEmail;
    ->
    -> getEmail: LOOP
    -> FETCH curEmail INTO emailAddress;
    -> IF finished = 1 THEN
    -> LEAVE getEmail;
    -> END IF;
    ->
    -> SET emailList = CONCAT(emailAddress,";",emailList);
    -> END LOOP getEmail;
    -> CLOSE curEmail;
    ->
    -> END$$
Query OK, 0 rows affected (0.015 sec)

MariaDB [fitnessmanagement]> DELIMITER ;
MariaDB [fitnessmanagement]> SET @emailList = "";
Query OK, 0 rows affected (0.004 sec)

MariaDB [fitnessmanagement]> CALL createEmailList(@emailList);
Query OK, 0 rows affected (0.010 sec)

MariaDB [fitnessmanagement]> SELECT @emailList;
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------+
| @emailList
                                                                                                                                                                                                     |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------+
| email.id.7@email.com;email.id.27@email.com;email.id.22@email.com;email.id.1@email.com;email.id.45@email.com;email.id.24@email.com;email.id.18@email.com;email.id.7@email.com;email.id.46@email.com;email.id.40@em
ail.com;email.id.27@email.com;email.id.43@email.com;email.id.38@email.com;email.id.19@email.com;email.id.46@email.com;email.id.28@email.com;email.id.45@email.com;email.id.10@email.com;email.id.4@email.com;email.
id.42@email.com;email.id.25@email.com;email.id.16@email.com;email.id.17@email.com;email.id.45@email.com;email.id.49@email.com;email.id.44@email.com;email.id.37@email.com;email.id.18@email.com;email.id.4@email.co
m;email.id.42@email.com;email.id.33@email.com;email.id.23@email.com;email.id.28@email.com;email.id.21@email.com;email.id.34@email.com;email.id.5@email.com;email.id.35@email.com;email.id.34@email.com;email.id.31@
email.com;email.id.37@email.com;email.id.19@email.com;email.id.33@email.com;email.id.18@email.com;email.id.27@email.com;email.id.9@email.com;email.id.25@email.com;email.id.31@email.com;email.id.25@email.com;emai
l.id.39@email.com;email.id.23@email.com;email.id.42@email.com;email.id.38@email.com;email.id.50@email.com;email.id.43@email.com;email.id.31@email.com;email.id.14@email.com;email.id.32@email.com;email.id.16@email
.com;email.id.9@email.com;email.id.3@email.com; |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Developing a Frontend

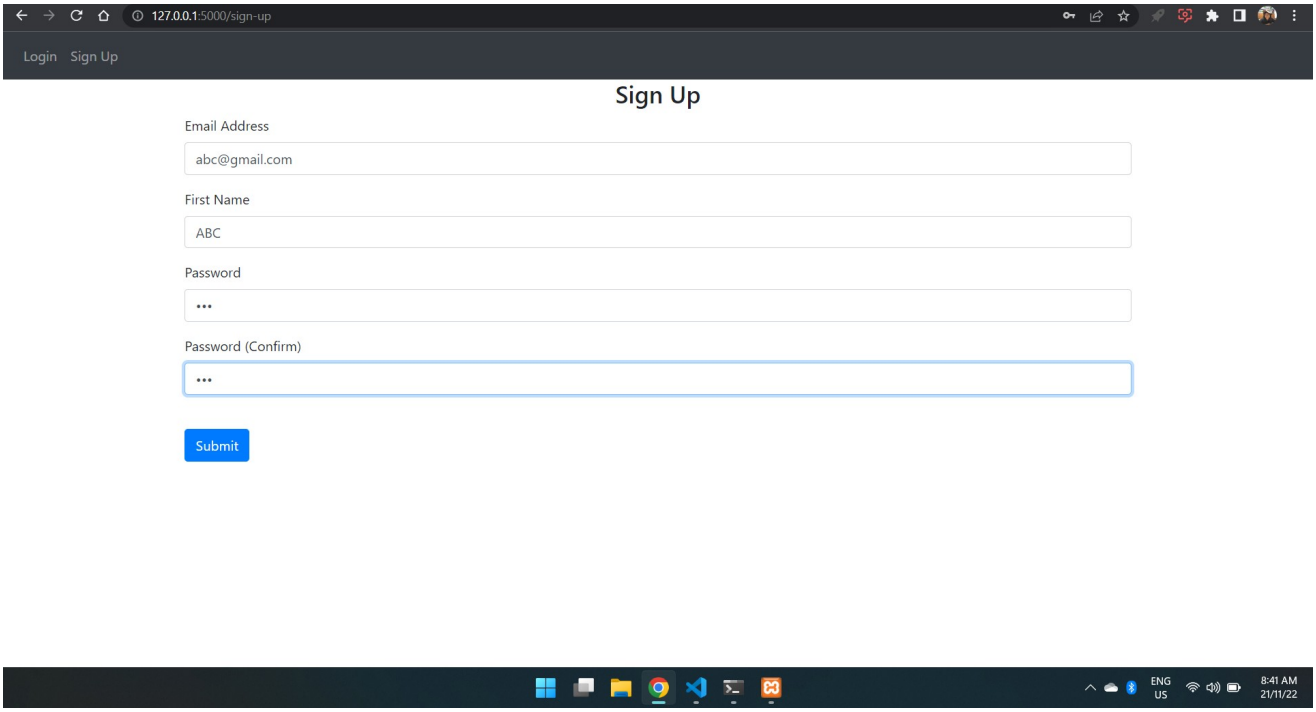


Figure 3: Sign up page

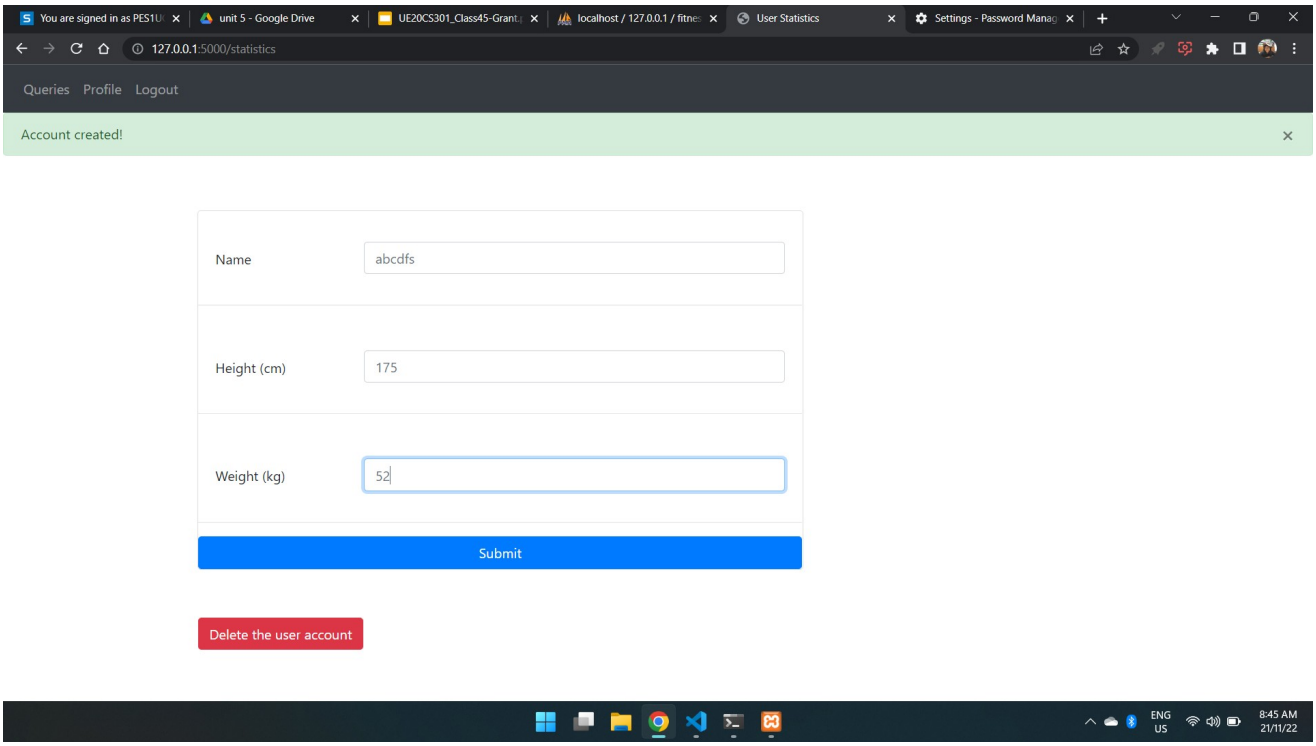


Figure 4: Creation of an account

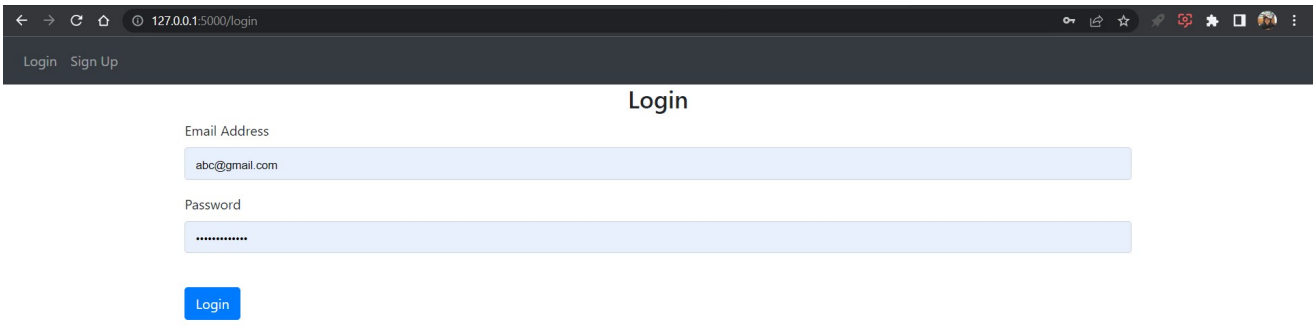


Figure 5: Login page for existing account

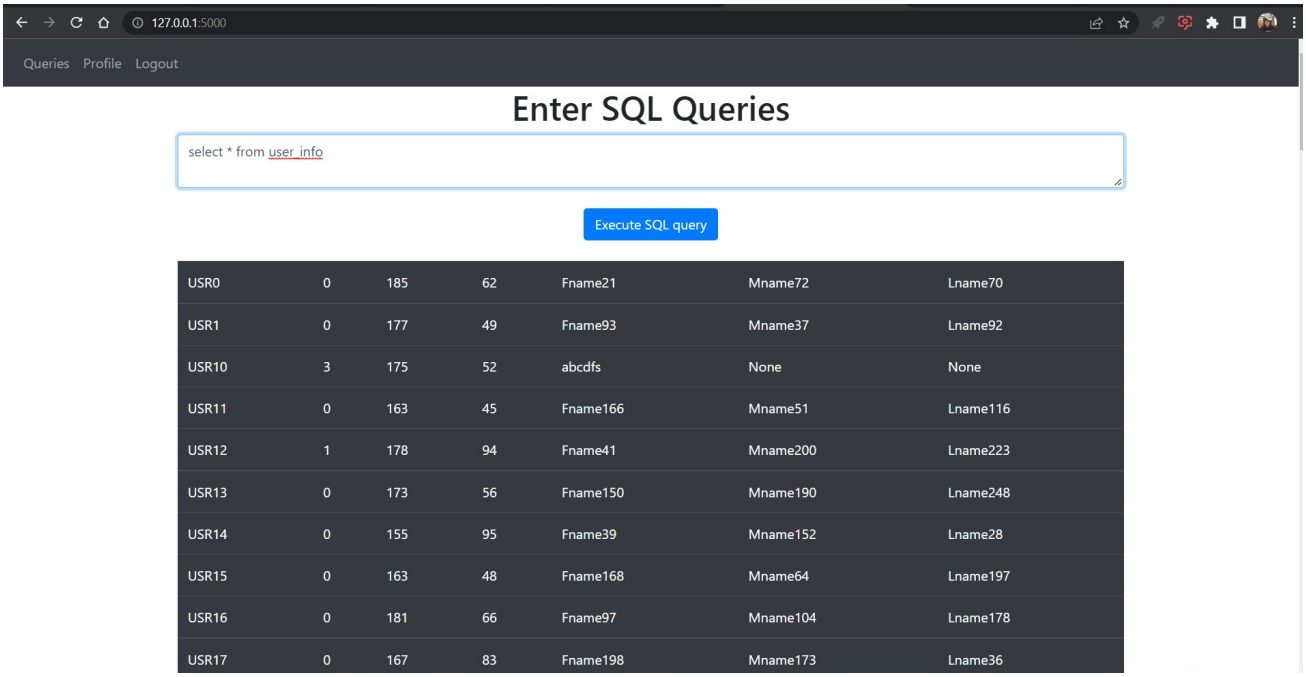


Figure 6: A window to accept and run any SQL statement