
FitZone - Gym Management System

FitZone is a comprehensive solution designed to streamline the operations of a fitness center. This system incorporates a robust database structure and a user-friendly web interface to manage various aspects of the gym, including member information, staff management, equipment tracking, class scheduling, attendance monitoring, payment processing, and more.

Key Features

Member Management:

- Capture and store member details, including personal information, contact details, and membership type.
- Track membership start dates, renewals, and member-specific health assessments.

Staff Management:

- Manage gym staff information, including roles, responsibilities, and hire dates.
- Assign and schedule tasks for staff members.

Equipment Tracking:

- Maintain an inventory of gym equipment with details on purchase dates, maintenance schedules, and current status.
- Receive alerts for equipment maintenance and updates.

Class Scheduling:

- Create and manage fitness classes with details on class names, instructors, schedules, and maximum capacities.
- Allow members to register for classes online.

Attendance Monitoring:

- Track member attendance for each class.
- Generate attendance reports for staff and members.

Payment Processing:

- Handle member payments securely with various payment methods.
- Provide a detailed payment history for members and staff.

Instructors' Schedule:

- Create schedules for fitness instructors with information on classes and timings.

Workout Plans:

- Offer predefined workout plans with descriptions and associated exercises.
- Allow members to personalize their workout plans based on fitness goals.

Health Assessment:

- Record and monitor health assessment details for each member, including BMI and other health metrics.

Feedback System:

- Collect feedback from members about classes, equipment, and overall gym experience.
- Provide a platform for staff to review and respond to member feedback.

Gym Locations:

- Maintain information on different gym locations, including addresses, contact details, and facilities.

Membership Types:

- Define and manage various membership types with details on benefits and pricing.

Web Interface:

- A responsive and user-friendly website accessible to both members and staff.
- Personalized dashboards for members and staff to view relevant information.
- Admin panel for administrative tasks, including database management and user permissions.

Security Measures:

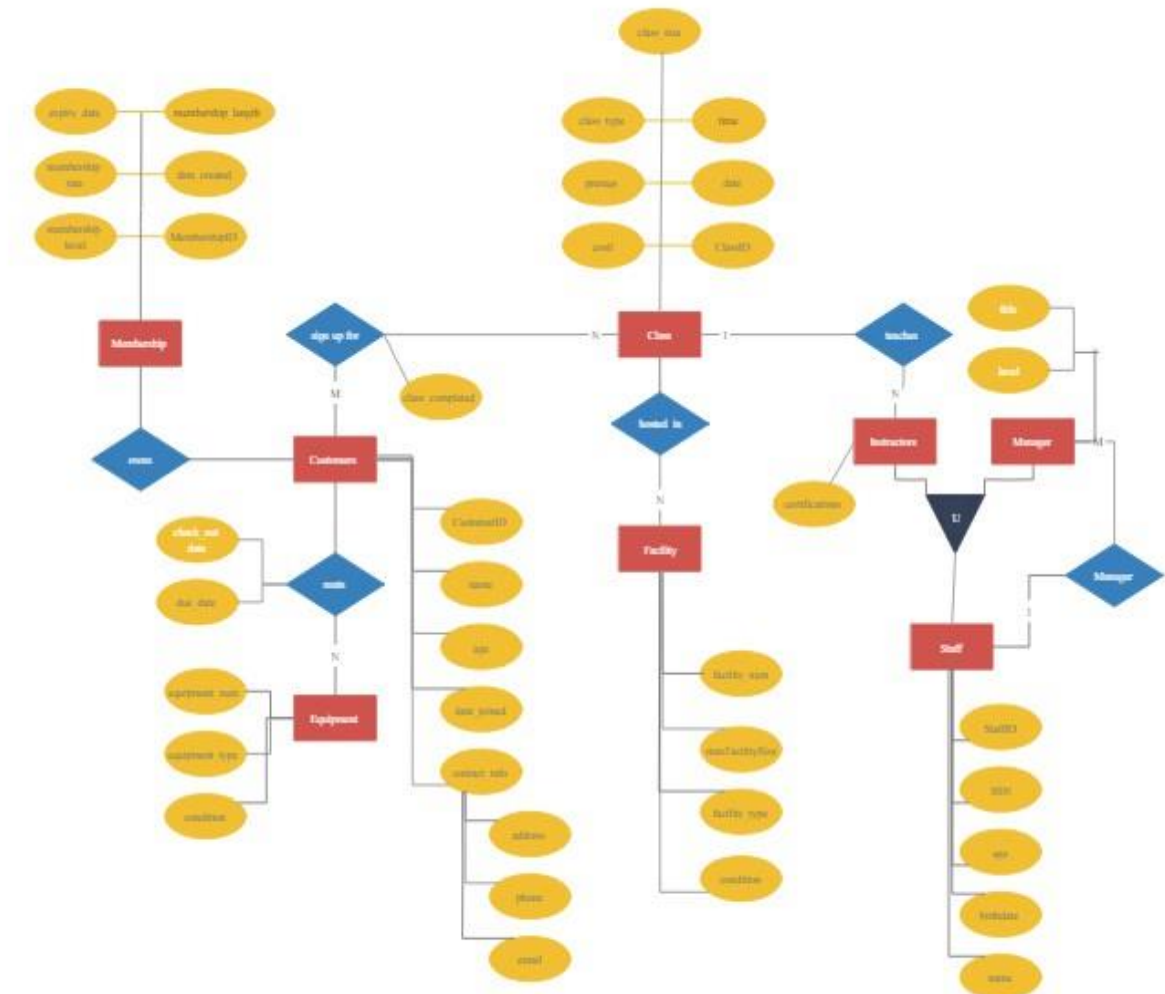
- User authentication and authorization to ensure data security.
- Encrypted communication for sensitive information.
- Regular backups to prevent data loss.

Technology Stack:

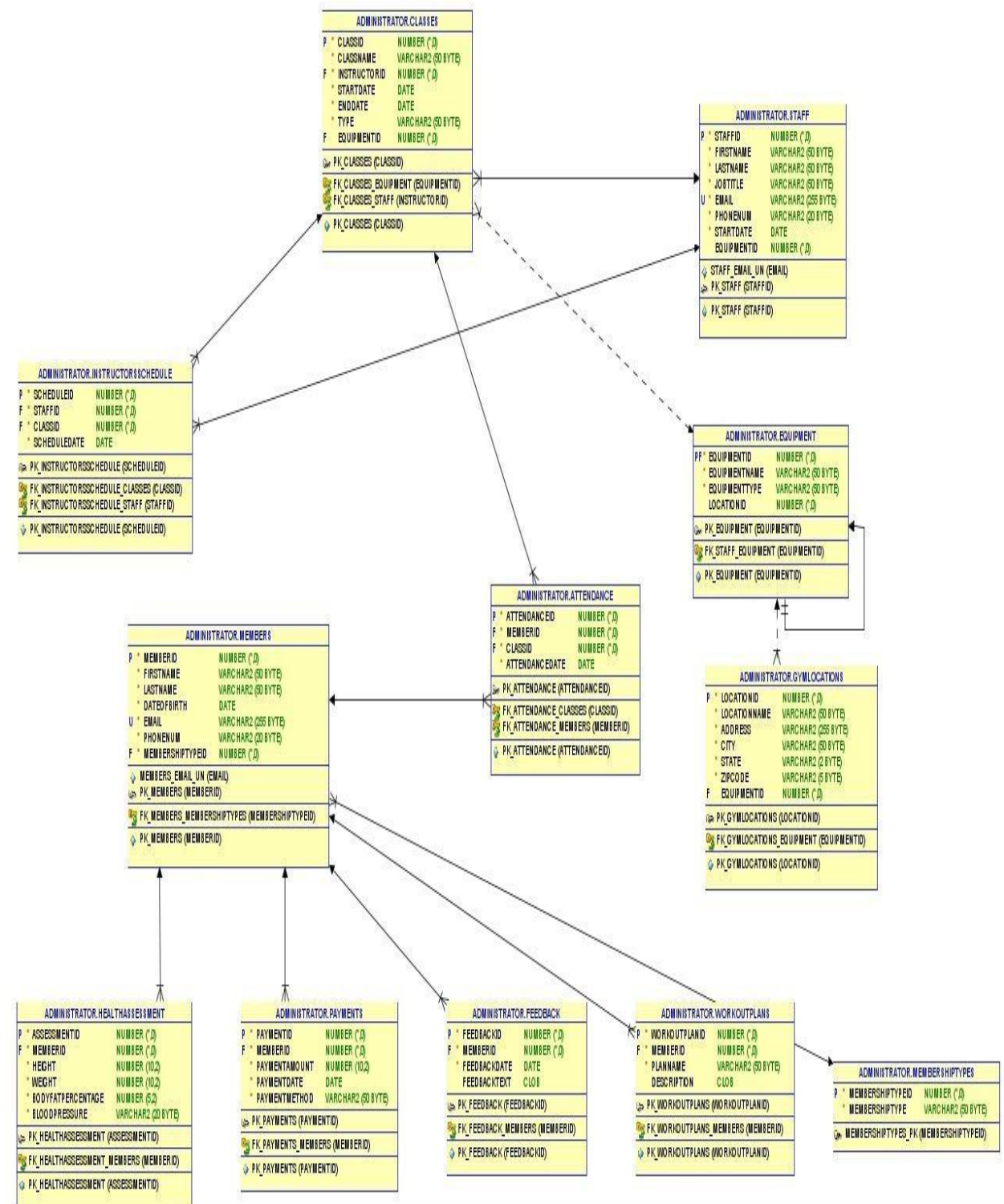
- Database: MySQL, PostgreSQL, or any preferred relational database management system.
- Backend: PHP, Python (Django/Flask), or Node.js.
- Frontend: HTML5, CSS3, JavaScript (React, Angular, or Vue.js).
- Additional tools: Payment gateway integration, charting libraries for data visualization.

The Gym Management System is designed to enhance the efficiency of gym operations, provide a seamless experience for members, and empower staff with tools to manage daily tasks effectively. It aims to create a well-organized and technologically advanced environment for both fitness enthusiasts and gym administrators.

ER Diagram

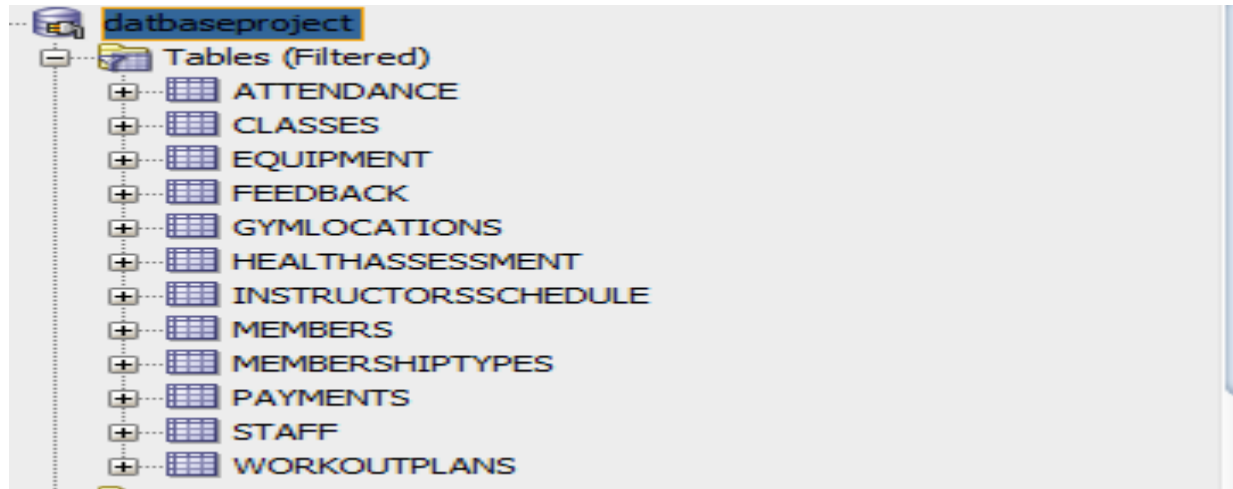


Data Dictionary



Database Details

Tables



Members

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 MEMBERID	NUMBER (38,0)	No	(null)	1	(null)	
2 FIRSTNAME	VARCHAR2 (50 BYTE)	No	(null)	2	(null)	
3 LASTNAME	VARCHAR2 (50 BYTE)	No	(null)	3	(null)	
4 DATEOFBIRTH	DATE	No	(null)	4	(null)	
5 EMAIL	VARCHAR2 (255 BYTE)	No	(null)	5	(null)	
6 PHONENUM	VARCHAR2 (20 BYTE)	No	(null)	6	(null)	
7 MEMBERSHIPID	NUMBER (38,0)	No	(null)	7	(null)	

Staff

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 STAFFID	NUMBER (38,0)	No	(null)	1	(null)	
2 FIRSTNAME	VARCHAR2 (50 BYTE)	No	(null)	2	(null)	
3 LASTNAME	VARCHAR2 (50 BYTE)	No	(null)	3	(null)	
4 JOBTITLE	VARCHAR2 (50 BYTE)	No	(null)	4	(null)	
5 EMAIL	VARCHAR2 (255 BYTE)	No	(null)	5	(null)	
6 PHONENUM	VARCHAR2 (20 BYTE)	No	(null)	6	(null)	
7 STARTDATE	DATE	No	(null)	7	(null)	
8 EQUIPMENTID	NUMBER (38,0)	Yes	(null)	8	(null)	




Equipment

Payments

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 PAYMENTID	NUMBER (38,0)	No	(null)	1 (null)	
2 MEMBERID	NUMBER (38,0)	No	(null)	2 (null)	
3 PAYMENTAMOUNT	NUMBER (10,2)	No	(null)	3 (null)	
4 PAYMENTDATE	DATE	No	(null)	4 (null)	
5 PAYMENTMETHOD	VARCHAR2 (50 BYTE)	No	(null)	5 (null)	

Instructors Schedule

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

 ▼ Actions...

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1	SCHEDULEID	NUMBER (38,0)	No	(null)	1	(null)
2	STAFFID	NUMBER (38,0)	No	(null)	2	(null)
3	CLASSID	NUMBER (38,0)	No	(null)	3	(null)
4	SCHEDULEDATE	DATE	No	(null)	4	(null)




Workout Plans

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	WORKOUTPLANID	NUMBER(38,0)	No	(null)	1 (null)	
2	MEMBERID	NUMBER(38,0)	No	(null)	2 (null)	
3	PLANNAME	VARCHAR2(50 BYTE)	No	(null)	3 (null)	
4	DESCRIPTION	CLOB	Yes	(null)	4 (null)	




Health Assessment

Column	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ASSESSMENTID	NUMBER(38,0)	No	(null)	1 (null)	
2 MEMBERID	NUMBER(38,0)	No	(null)	2 (null)	
3 HEIGHT	NUMBER(10,2)	No	(null)	3 (null)	
4 WEIGHT	NUMBER(10,2)	No	(null)	4 (null)	
5 BODYFATPERCENTAGE	NUMBER(5,2)	No	(null)	5 (null)	
6 BLOODPRESSURE	VARCHAR2(20 BYTE)	No	(null)	6 (null)	

Feedback




Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
  	▼ Actions...											
	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS						
1	FEEDBACKID	NUMBER (38,0)	No	(null)	1	(null)						
2	MEMBERID	NUMBER (38,0)	No	(null)	2	(null)						
3	FEEDBACKDATE	DATE	No	(null)	3	(null)						
4	FEEDBACKTEXT	CLOB	Yes	(null)	4	(null)						

Gym Locations

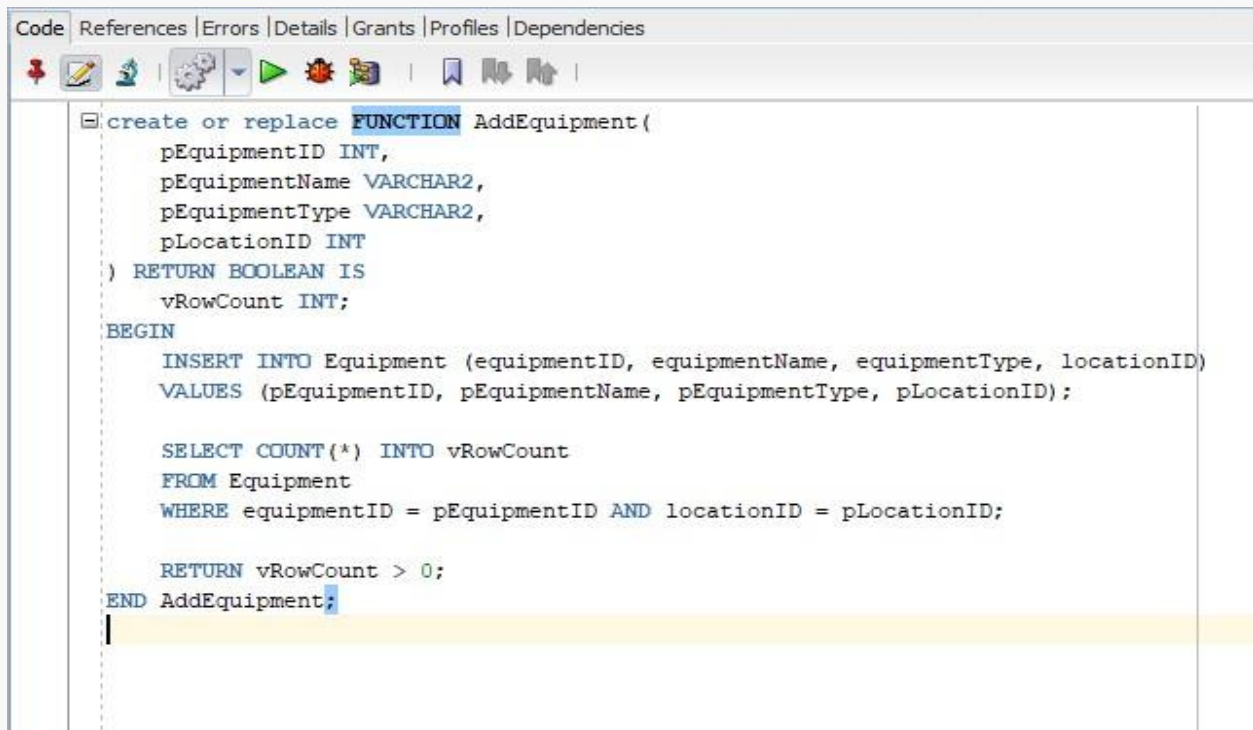
Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
  	▼ Actions...											
	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS						
1	LOCATIONID	NUMBER (38,0)	No	(null)	1	(null)						
2	LOCATIONNAME	VARCHAR2 (50 BYTE)	No	(null)	2	(null)						
3	ADDRESS	VARCHAR2 (255 BYTE)	No	(null)	3	(null)						
4	CITY	VARCHAR2 (50 BYTE)	No	(null)	4	(null)						
5	STATE	VARCHAR2 (2 BYTE)	No	(null)	5	(null)						
6	ZIPCODE	VARCHAR2 (5 BYTE)	No	(null)	6	(null)						
7	EQUIPMENTID	NUMBER (38,0)	Yes	(null)	7	(null)						

Membership Types

Function

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
  	▼ Actions...											
	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS						
1	MEMBERSHIPTYPEID	NUMBER (38,0)	No	(null)	1	(null)						
2	MEMBERSHIPTYPE	VARCHAR2 (50 BYTE)	No	(null)	2	(null)						

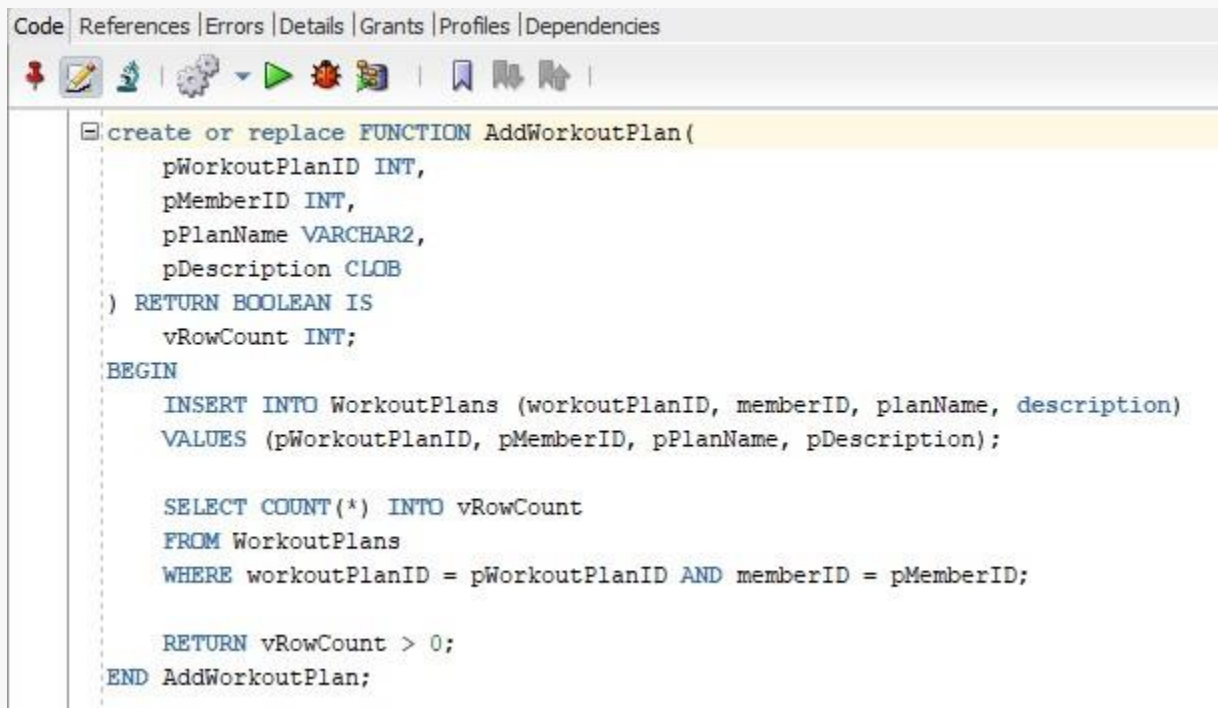
dd-Equipment



The screenshot shows a SQL IDE window with a toolbar at the top containing icons for saving, running, and other database operations. Below the toolbar, the 'Code' tab is active, displaying the following SQL code:

```
create or replace FUNCTION AddEquipment(  
    pEquipmentID INT,  
    pEquipmentName VARCHAR2,  
    pEquipmentType VARCHAR2,  
    pLocationID INT  
) RETURN BOOLEAN IS  
    vRowCount INT;  
BEGIN  
    INSERT INTO Equipment (equipmentID, equipmentName, equipmentType, locationID)  
    VALUES (pEquipmentID, pEquipmentName, pEquipmentType, pLocationID);  
  
    SELECT COUNT(*) INTO vRowCount  
    FROM Equipment  
    WHERE equipmentID = pEquipmentID AND locationID = pLocationID;  
  
    RETURN vRowCount > 0;  
END AddEquipment;
```

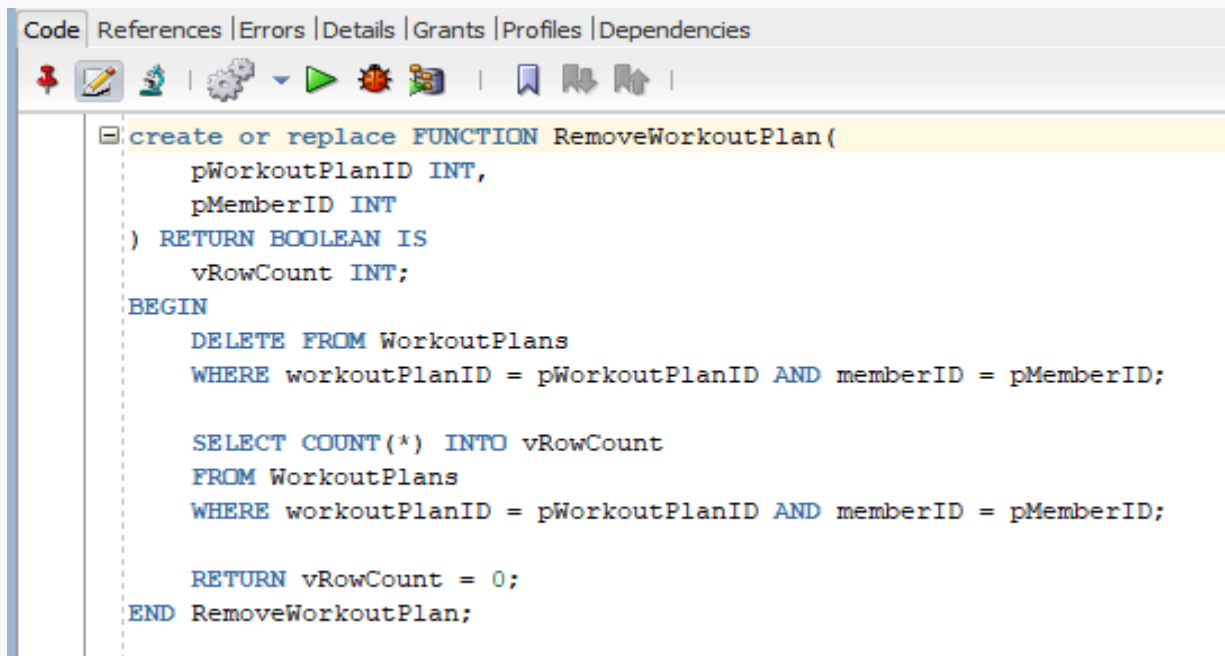
addWorkOutPlan



The screenshot shows a SQL IDE window with a toolbar at the top. Below the toolbar, the 'Code' tab is active, displaying the following SQL code:

```
create or replace FUNCTION AddWorkoutPlan(  
    pWorkoutPlanID INT,  
    pMemberID INT,  
    pPlanName VARCHAR2,  
    pDescription CLOB  
) RETURN BOOLEAN IS  
    vRowCount INT;  
BEGIN  
    INSERT INTO WorkoutPlans (workoutPlanID, memberID, planName, description)  
    VALUES (pWorkoutPlanID, pMemberID, pPlanName, pDescription);  
  
    SELECT COUNT(*) INTO vRowCount  
    FROM WorkoutPlans  
    WHERE workoutPlanID = pWorkoutPlanID AND memberID = pMemberID;  
  
    RETURN vRowCount > 0;  
END AddWorkoutPlan;
```

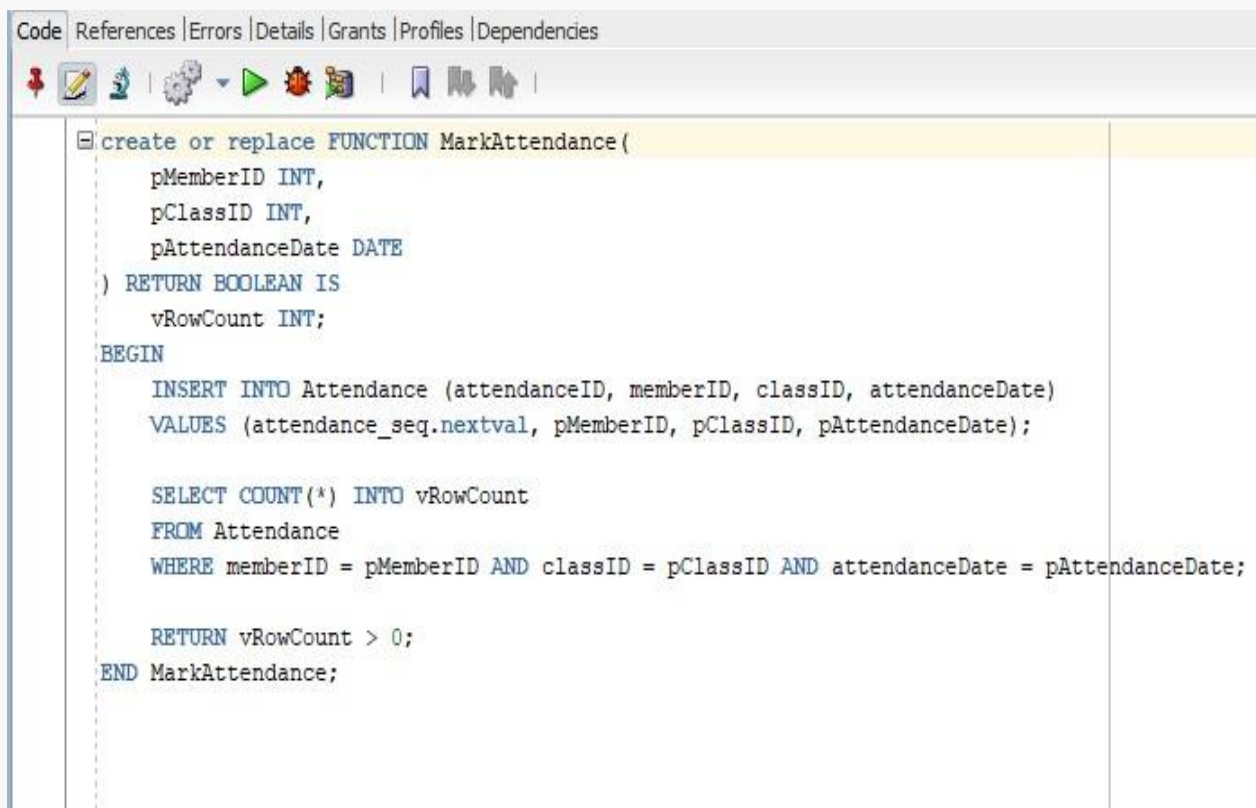
removeWorkOutPlan



The screenshot shows a database IDE window with a tab labeled 'Code'. The window contains the following SQL code for the 'RemoveWorkoutPlan' function:

```
create or replace FUNCTION RemoveWorkoutPlan(  
    pWorkoutPlanID INT,  
    pMemberID INT  
) RETURN BOOLEAN IS  
    vRowCount INT;  
BEGIN  
    DELETE FROM WorkoutPlans  
    WHERE workoutPlanID = pWorkoutPlanID AND memberID = pMemberID;  
  
    SELECT COUNT(*) INTO vRowCount  
    FROM WorkoutPlans  
    WHERE workoutPlanID = pWorkoutPlanID AND memberID = pMemberID;  
  
    RETURN vRowCount = 0;  
END RemoveWorkoutPlan;
```

markAttendance



The screenshot shows a database IDE window with a tab labeled 'Code'. The window contains the following SQL code for the 'MarkAttendance' function:

```
create or replace FUNCTION MarkAttendance(  
    pMemberID INT,  
    pClassID INT,  
    pAttendanceDate DATE  
) RETURN BOOLEAN IS  
    vRowCount INT;  
BEGIN  
    INSERT INTO Attendance (attendanceID, memberID, classID, attendanceDate)  
    VALUES (attendance_seq.nextval, pMemberID, pClassID, pAttendanceDate);  
  
    SELECT COUNT(*) INTO vRowCount  
    FROM Attendance  
    WHERE memberID = pMemberID AND classID = pClassID AND attendanceDate = pAttendanceDate;  
  
    RETURN vRowCount > 0;  
END MarkAttendance;
```

cancelMembership

```
Code | References | Errors | Details | Grants | Profiles | Dependencies
create or replace FUNCTION CancelMembership(
    pMemberID INT
) RETURN BOOLEAN IS
    vRowCount INT;
BEGIN
    DELETE FROM Members
    WHERE memberID = pMemberID;

    SELECT COUNT(*) INTO vRowCount
    FROM Members
    WHERE memberID = pMemberID;

    RETURN vRowCount = 0;
END CancelMembership;
```

Views

Attendance View

	ATTENDANCEID	MEMBERID	CLASSID	ATTENDANCEDATE
1	1	1	1	15-JAN-24
2	2	2	2	05-MAR-24
3	4	4	4	15-MAR-24
4	3	3	3	20-FEB-24
5	5	5	5	25-FEB-24
6	6	6	6	31-MAR-24

Members View

	MEMBERID	FIRSTNAME	LASTNAME	DATEOFBIRTH	EMAIL	PHONENUM	MEMBERSHIPTYPEID
1	1	John	Doe	01-JAN-90	john@example.com	1234567890	1
2	2	Sophia	Anderson	12-MAY-85	sophia@example.com	4445556677	2
3	3	James	Wilson	30-SEP-95	james@example.com	9991112233	3
4	4	Emma	Taylor	18-DEC-88	emma@example.com	7778889900	4
5	5	Oliver	Clark	08-MAR-73	oliver@example.com	2223334455	5
6	6	Ava	Martinez	22-JUL-98	ava@example.com	6667778899	1

Staff View

Columns Data Grants Dependencies Details Triggers SQL Errors								
Sort.. Filter:								
	STAFFID	FIRSTNAME	LASTNAME	JOBTITLE	EMAIL	PHONENUM	STARTDATE	EQUIPMENTID
1	1	Jane	Smith	Instructor	jane@example.com	9876543210	01-JAN-20	1
2	2	Alice	Johnson	Trainer	alice@example.com	1234567890	15-MAR-21	1
3	3	Bob	Williams	Instructor	bob@example.com	9876543210	20-MAY-22	2
4	4	Charlie	Miller	Front Desk	charlie@example.com	5551237890	10-JUL-23	3
5	5	Eva	Brown	Trainer	eva@example.com	1112223344	25-AUG-21	4
6	6	David	Davis	Front Desk	david@example.com	9998887766	01-DEC-22	5

Equipment View

Columns Data Grants Dependencies Details Triggers SQL Errors				
Sort.. Filter:				
	EQUIPMENTID	EQUIPMENTNAME	EQUIPMENTTYPE	LOCATIONID
1	1	Treadmill	Cardio	1
2	2	Elliptical Machine	Cardio	1
3	3	Dumbbells Set	Strength	2
4	4	Dance Floor	Dance	3
5	5	Yoga Mats	Yoga	4
6	6	Stationary Bike	Cardio	5

Classes View




Columns Data Grants Dependencies Details Triggers SQL Errors							
Sort.. Filter:							
	CLASSID	CLASSNAME	INSTRUCTORID	STARTDATE	ENDDATE	TYPE	EQUIPMENTID
1	1	Yoga Class	1	01-JAN-24	01-FEB-24	Yoga	1
2	2	Cardio Blast	2	01-MAR-24	30-MAR-24	Cardio	1
3	4	Zumba Dance	3	20-FEB-24	20-MAR-24	Dance	1
4	3	Strength Training	4	15-FEB-24	15-MAR-24	Strength	2
5	5	Yoga Meditation	2	25-FEB-24	25-MAR-24	Yoga	3
6	6	Spinning Class	5	05-MAR-24	31-MAR-24	Cardio	4

Payments View

Columns Data Grants Dependencies Details Triggers SQL Errors						
Sort.. Filter:						
	PAYMENTID	MEMBERID	PAYMENTAMOUNT	PAYMENTDATE	PAYMENTMETHOD	
1	1	1	50	15-JAN-24	Credit Card	
2	2	2	75	05-MAR-24	Credit Card	
3	3	3	100	20-FEB-24	PayPal	
4	4	4	60	15-MAR-24	Credit Card	
5	5	5	80	25-FEB-24	Cash	
6	6	6	120	31-MAR-24	Credit Card	

Instructors Schedule View

Columns | Data | Grants | Dependencies | Details | Triggers | SQL | Errors



▼ Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	SCHEDULEID	NUMBER (38)	No	(null)	1 (null)		YES	YES	YES
2	STAFFID	NUMBER (38)	No	(null)	2 (null)		YES	YES	YES
3	CLASSID	NUMBER (38)	No	(null)	3 (null)		YES	YES	YES
4	SCHEDULEDATE	DATE	No	(null)	4 (null)		YES	YES	YES

Workout Plans View

Columns Data Grants Dependencies Details Triggers SQL Errors				
Sort.. Filter:				
	WORKOUTPLANID	MEMBERID	PLANNAME	DESCRIPTION
1	1	1	1 Weight Loss Plan	Custom workout plan for weight loss.
2	2	2	2 Muscle Gain Plan	Custom workout plan for muscle gain.
3	3	3	3 Flexibility Plan	Tailored exercises to improve flexibility.
4	4	4	4 Cardiovascular Plan	High-intensity cardio workouts for improved cardiovascular health.
5	5	5	5 Weight Management Plan	Balanced workouts for weight management.
6	6	6	6 Strength and Endurance Plan	Combination of strength training and endurance exercises.


Health Assessment View

Columns Data Grants Dependencies Details Triggers SQL Errors						
Sort.. Filter:						
	ASSESSMENTID	MEMBERID	HEIGHT	WEIGHT	BODYFATPERCENTAGE	BLOODPRESSURE
1	1	1	160.5	65.2	20	120/80
2	2	2	175	70.5	15.5	130/75
3	3	3	160	55	22	118/82
4	4	4	185.5	80	18	140/90
5	5	5	150	65	20	125/80
6	6	6	162.5	68	25	110/70


Feedback View

Columns Data Grants Dependencies Details Triggers SQL Errors				
Sort.. Filter:				
	FEEDBACKID	MEMBERID	FEEDBACKDATE	FEEDBACKTEXT
1	1	1	15-JAN-24	Great class!
2	2	2	205-MAR-24	Fantastic trainer! Enjoyed every session.
3	3	3	320-FEB-24	Great variety in class routines. Fun atmosphere.
4	4	4	415-MAR-24	Helpful staff, clean equipment. Enjoying my workouts.
5	5	5	525-FEB-24	Friendly environment. Love the yoga classes.
6	6	6	631-MAR-24	Excellent facilities. Professional trainers.


Gym Locations View

Columns Data Grants Dependencies Details Triggers SQL Errors							
 Sort.. Filter: <input type="text"/>							
	LOCATIONID	LOCATIONNAME	ADDRESS	CITY	STATE	ZIPCODE	EQUIPMENTID
1		1 Main Gym	123 Main St	City	CA	12345	1
2		2 Downtown Gym	456 Center St	City	CA	56789	2
3		3 Eastside Fitness	789 East Ave	Town	CA	67890	3
4		4 West End Health	890 West Blvd	Village	CA	78901	4
5		5 North Fitness Center	123 North St	City	CA	89012	5
6		6 Southside Gym	234 South Ave	Town	CA	90123	1
7		7 Central Fitness	567 Center Blvd	Metropolis	CA	34567	2
8		8 Northside Gym	890 North Ave	Town	CA	45678	3
9		9 South Fitness Center	123 South St	City	CA	56789	4
10		10 West Gym	234 West Ave	Village	CA	67890	5
11		11 East Fitness	789 East Blvd	City	CA	78901	1

Membership Types View

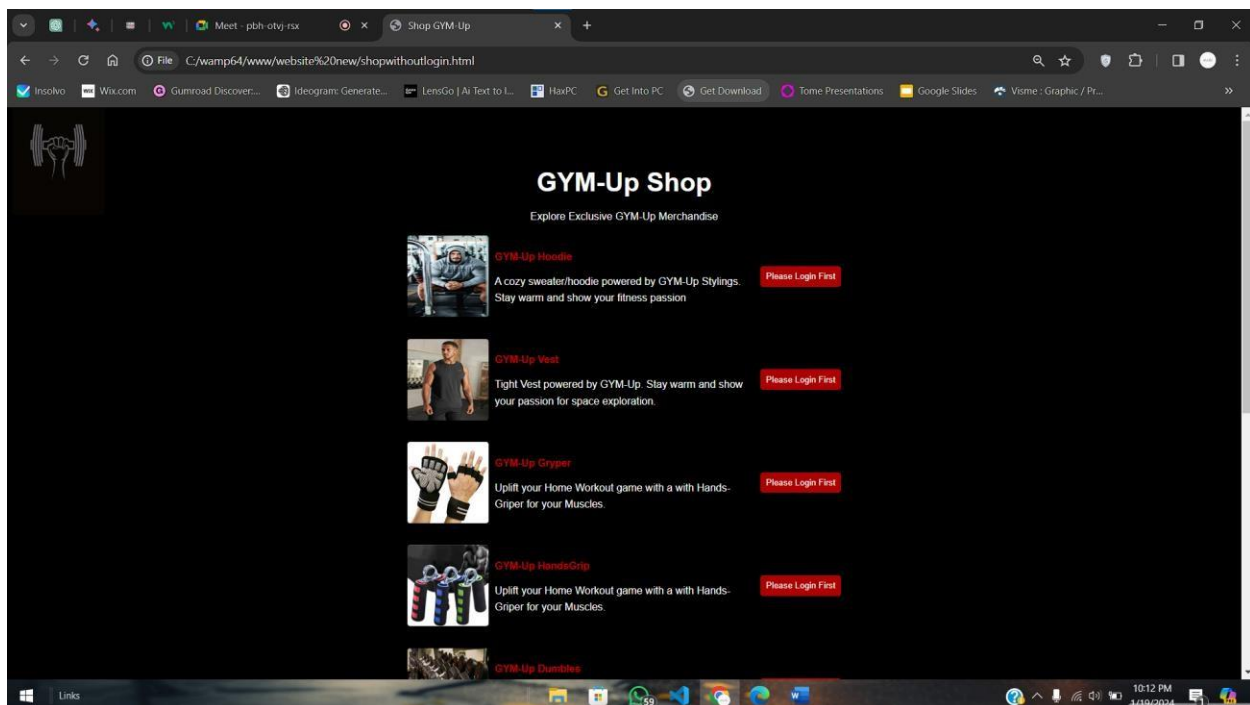
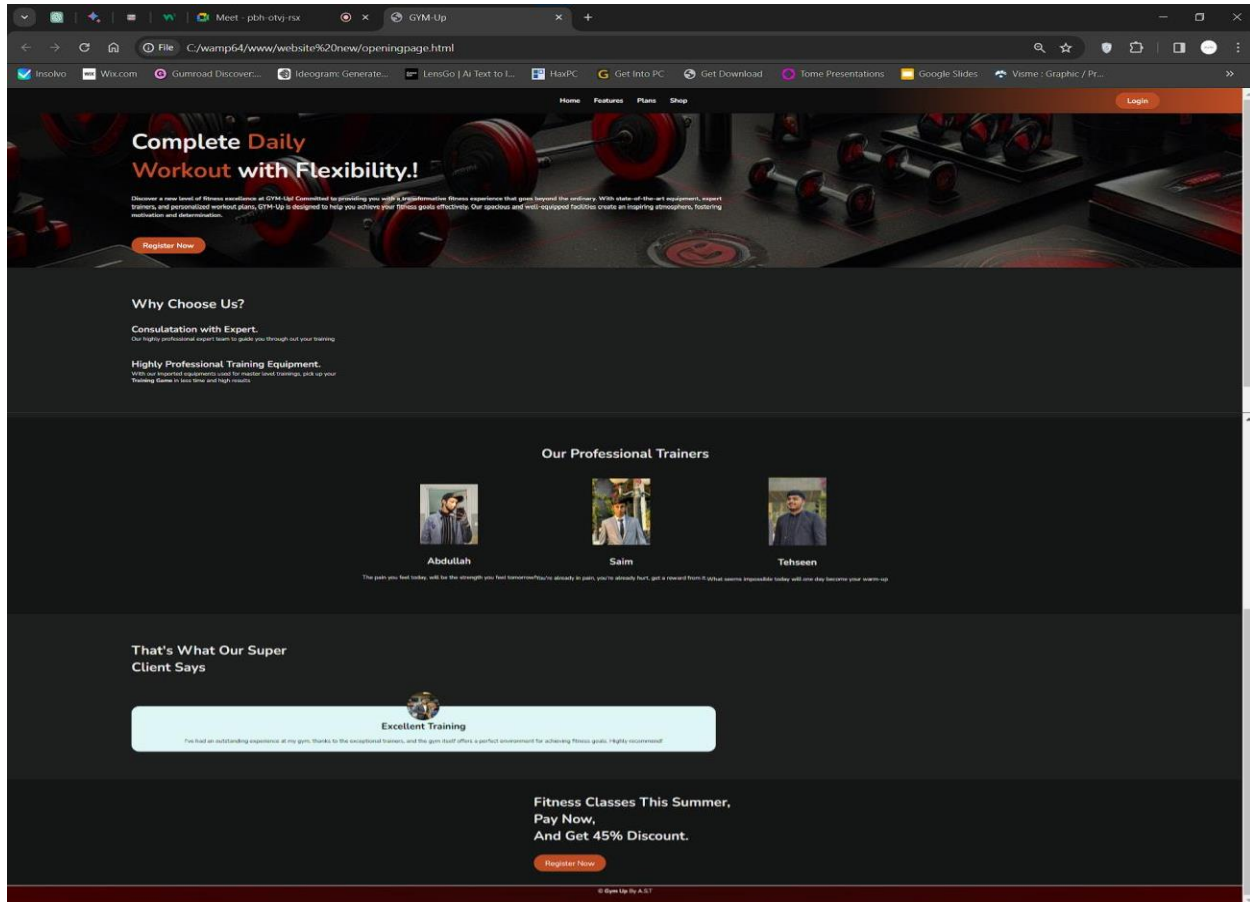
Columns Data Grants Dependencies Details Triggers SQL Errors		
 Sort.. Filter: <input type="text"/>		
	MEMBERSHIPTYPEID	MEMBERSHIPTYPE
1		1 Regular
2		2 Premium
3		3 VIP
4		4 Student
5		5 Senior

MemberPaymentWorkoutClasses View

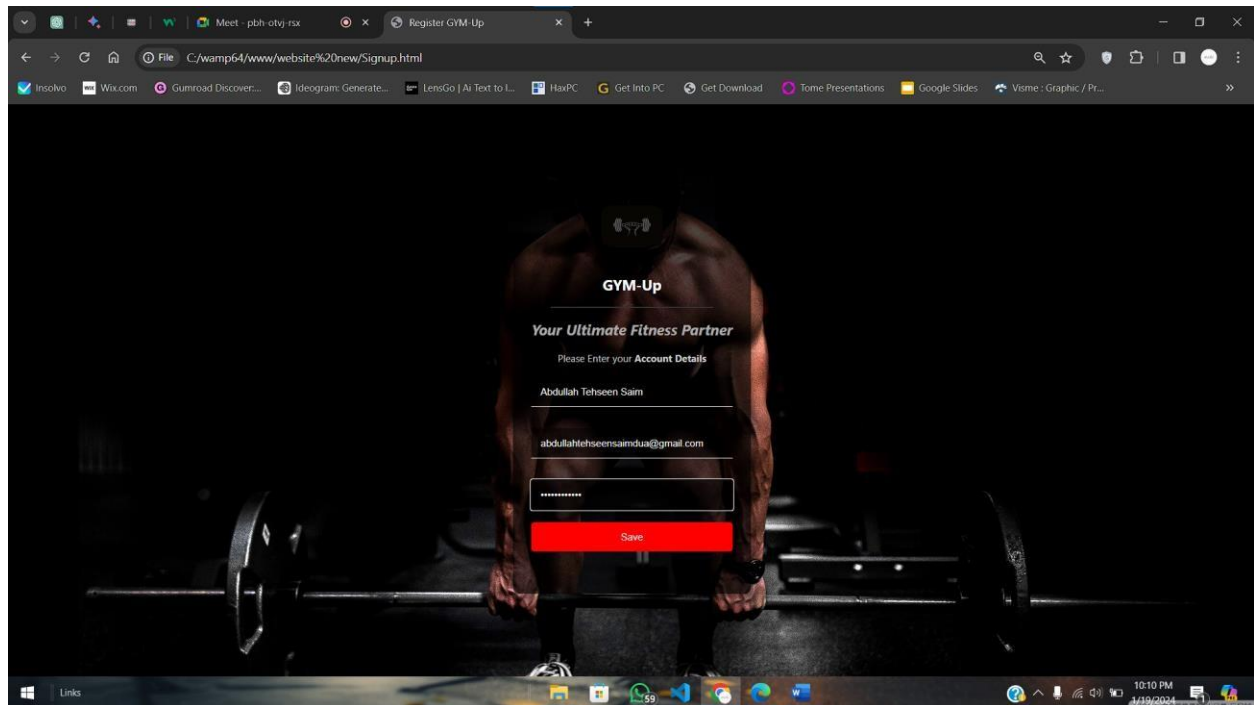
Columns Data Grants Dependencies Details Triggers SQL Errors										
<div> Sort... Filter: <input type="text"/> ▼ Actions...</div>										
	FIRSTNAME	LASTNAME	EMAIL	PAYMENTAMOUNT	PAYMENTDATE	WORKOUTPLANNAME	WORKOUTPLANDESCRIPTION	CLASSNAME	CLASSSTARTDATE	CLASSENDDATE
1	John	Doe	john@example.com	50	15-JAN-24	Weight Loss Plan	Custom workout plan for weight loss.	Yoga Class	01-JAN-24	01-FEB-24
2	Sophia	Anderson	sophia@example.com	75	05-MAR-24	Muscle Gain Plan	Custom workout plan for muscle gain.	Cardio Blast	01-MAR-24	30-MAR-24
3	James	Wilson	james@example.com	100	20-FEB-24	Flexibility Plan	Tailored exercises to improve flexibility.	Zumba Dance	20-FEB-24	20-MAR-24
4	Emma	Taylor	emma@example.com	60	15-MAR-24	Cardiovascular Plan	High-intensity cardio workouts for improved cardiovascular health.	Strength Training	15-FEB-24	15-MAR-24
5	Sophia	Anderson	sophia@example.com	75	05-MAR-24	Muscle Gain Plan	Custom workout plan for muscle gain.	Yoga Meditation	25-FEB-24	25-MAR-24
6	Oliver	Clark	oliver@example.com	80	25-FEB-24	Weight Management Plan	Balanced workouts for weight management.	Spinning Class	05-MAR-24	31-MAR-24
7	Ava	Martinez	ava@example.com	120	31-MAR-24	Strength and Endurance Plan	Combination of strength training and endurance exercises.	(null)	(null)	(null)

Website Details

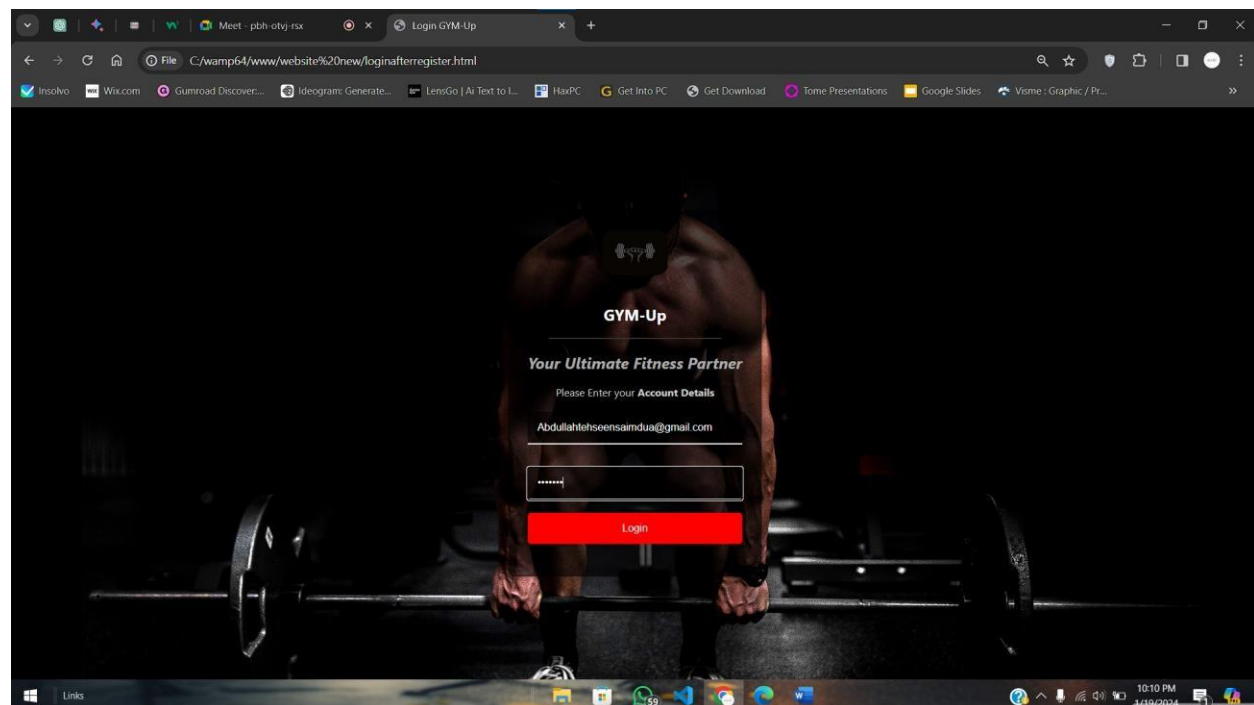
Web starting Page



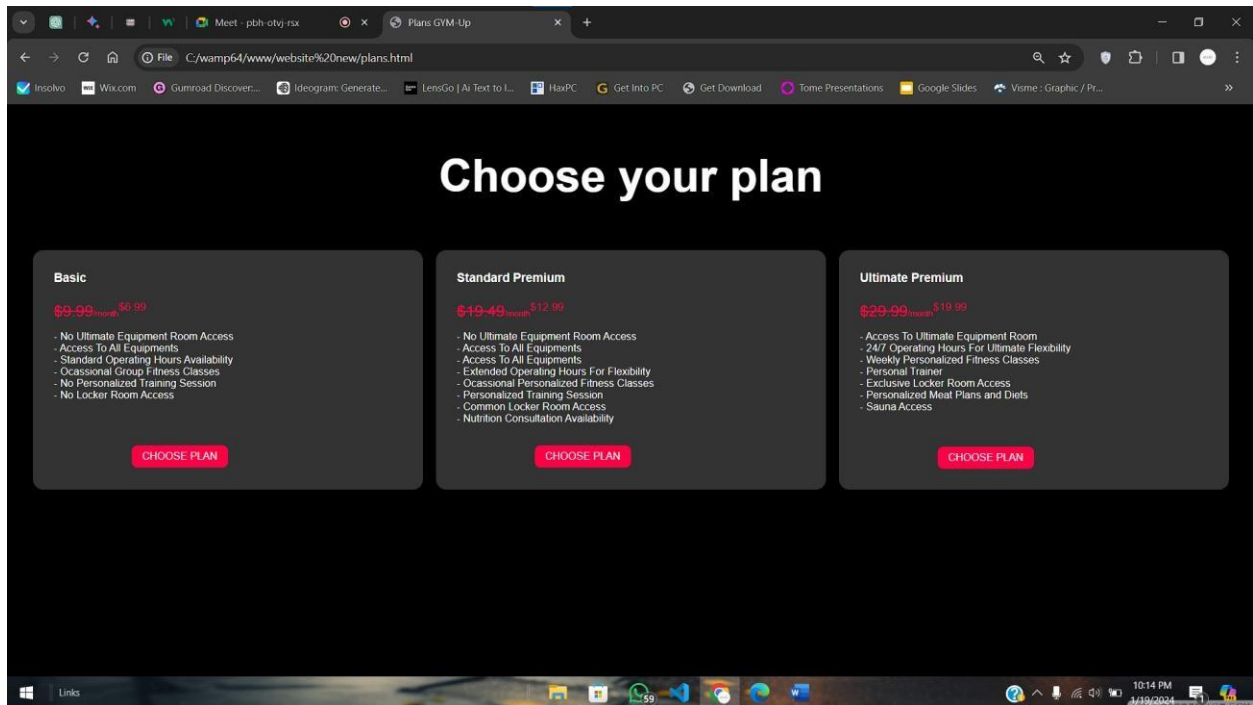
First, you have to Register yourself



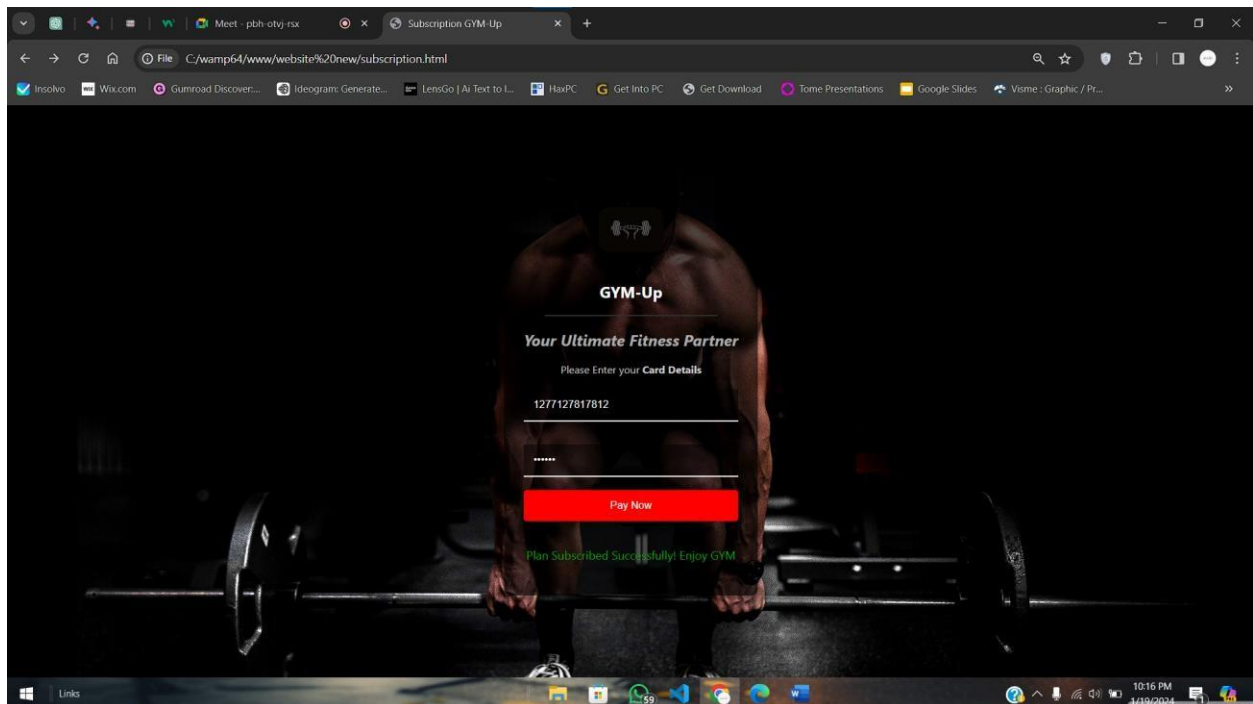
After Registering, you will be redirected to the LOGIN Page



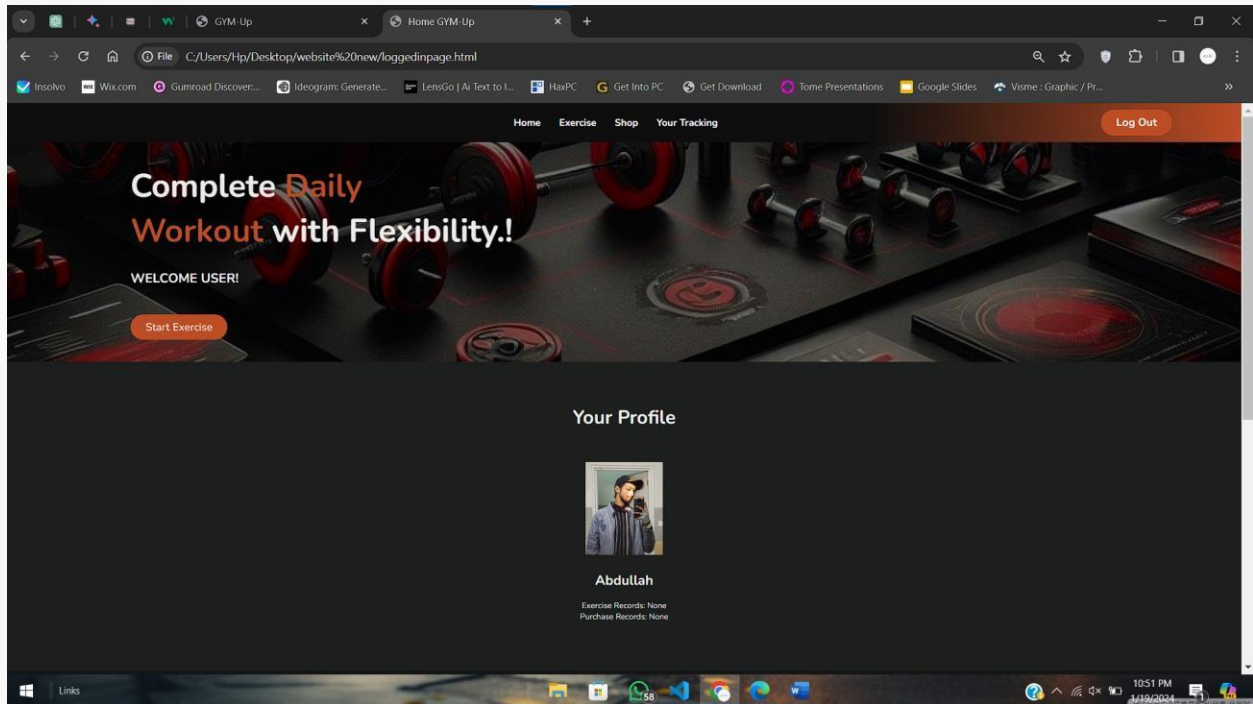
If logging in as a New account / subscription ends, you will need to subscribe a plan
Select a plan and it will redirect you to the subscription/payment page.



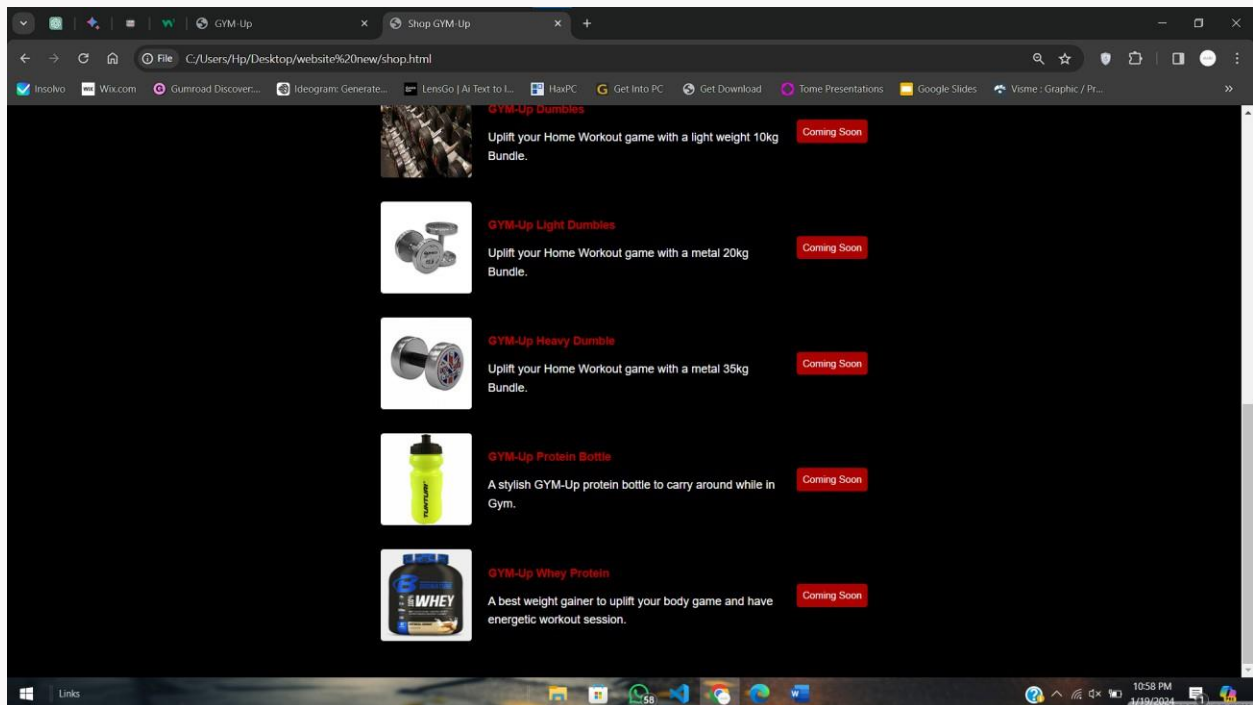
Enter card details, and a payment success or failed message will appear.



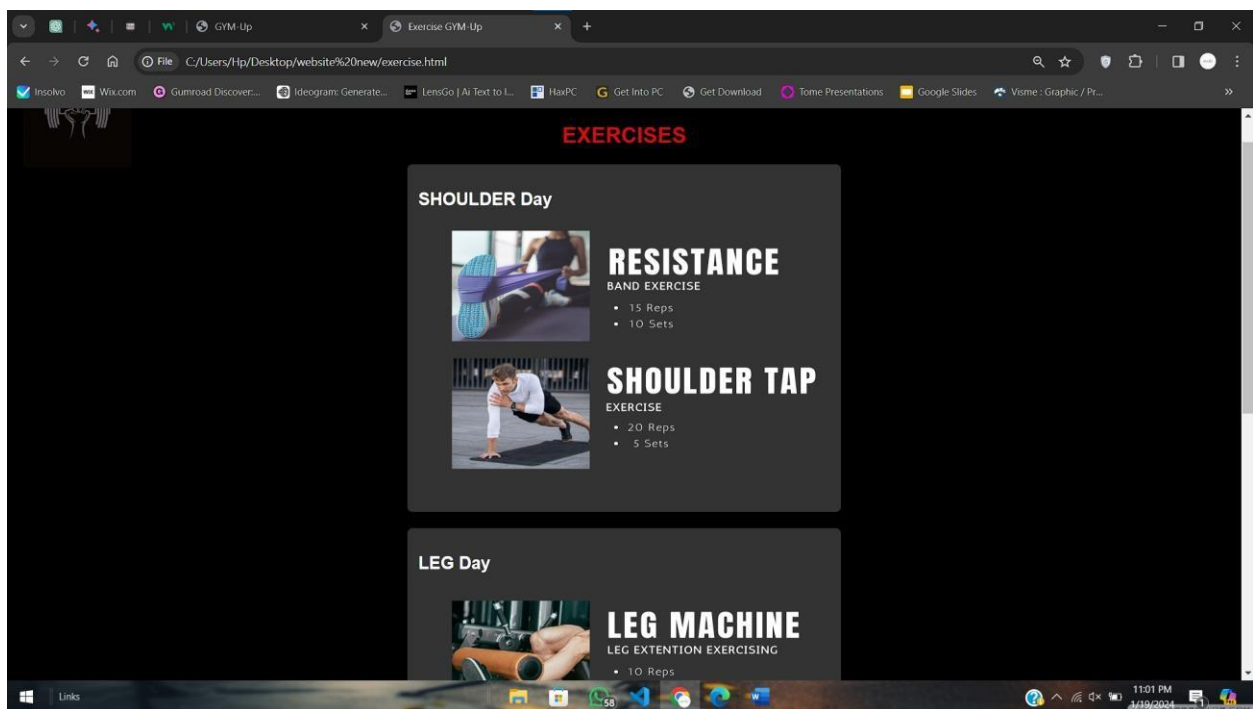
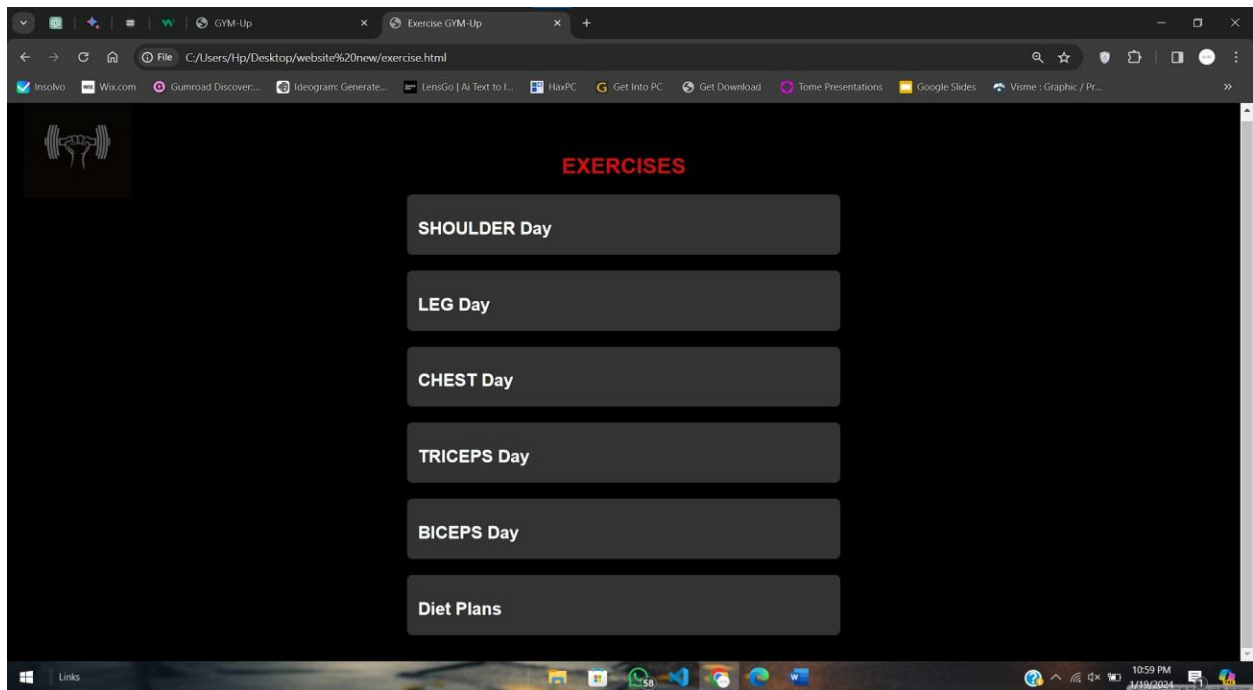
And then, you will be redirected to your account.



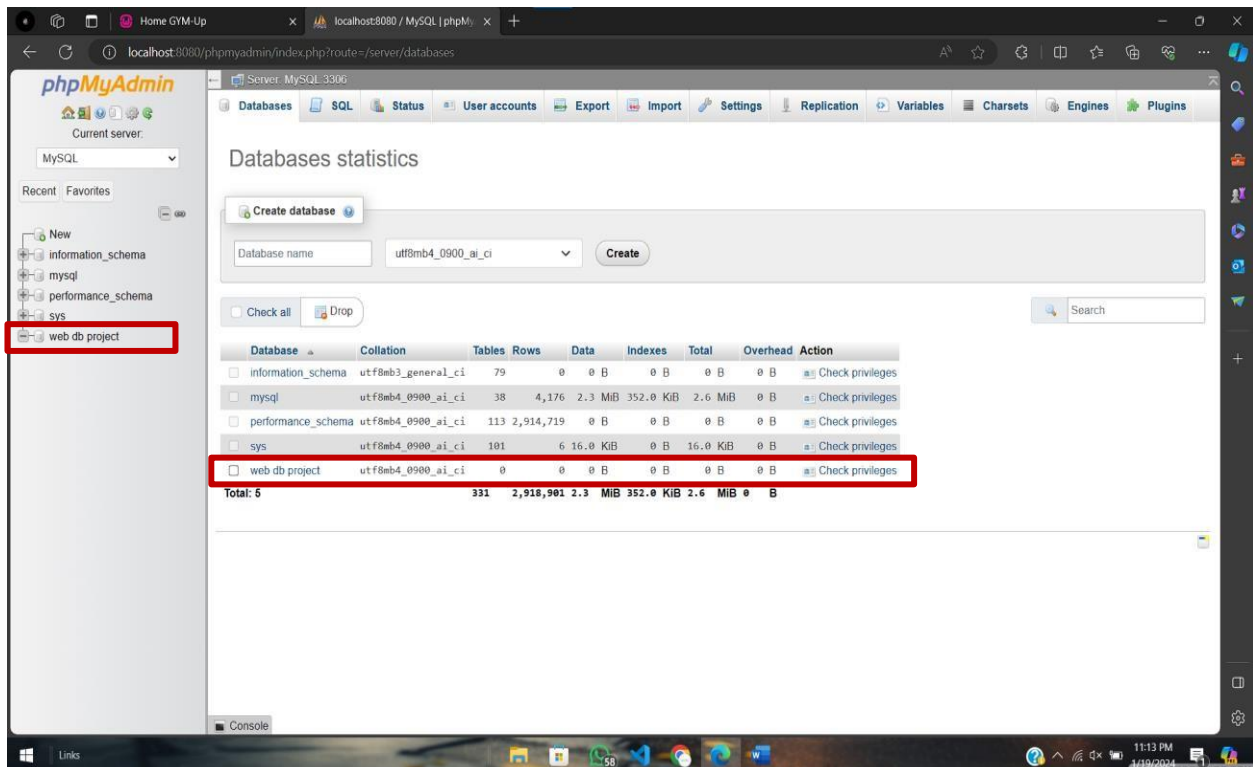
Under Shop Section, you can buy our stuff.



Under Exercise Section,



Web Server and Database Running on WAMP



The screenshot displays the phpMyAdmin interface for a MySQL server. The left sidebar shows a tree view of databases, with 'web db project' highlighted. The main panel shows the 'Databases statistics' page. A table lists the databases and their statistics. The 'web db project' database is highlighted in red.

Database	Collation	Tables	Rows	Data	Indexes	Total	Overhead	Action
information_schema	utf8mb3_general_ci	79	0	0 B	0 B	0 B	0 B	Check privileges
mysql	utf8mb4_0900_ai_ci	38	4,176	2.3 MiB	352.0 KiB	2.6 MiB	0 B	Check privileges
performance_schema	utf8mb4_0900_ai_ci	113	2,914,719	0 B	0 B	0 B	0 B	Check privileges
sys	utf8mb4_0900_ai_ci	101	6	16.0 KiB	0 B	16.0 KiB	0 B	Check privileges
web db project	utf8mb4_0900_ai_ci	0	0	0 B	0 B	0 B	0 B	Check privileges
Total: 5		331	2,918,901	2.3 MiB	352.0 KiB	2.6 MiB	0 B	