FLAVOR FINDER

Name: Abdul Rafay

Reg ID: 233679

Department: BS Software Engineering

Technologies uses:

- Frontend: HTML, CSS
- Backend: PHP
- Database: MySQL (Implemented using XAMPP)

Table of Contents

1. Project Scope and Requirements

1.1 Scope of the Platform:

1) Scope:

"Flavor Finder"

2) Description:

This platform allows users to find nearby restaurants based on their GPS location. Users can register, log in, search for restaurants, and leave reviews and ratings for restaurants they have visited.

1.2 Types of Services:

- Restaurant Search
- Restaurant Details
- User Reviews
- Ratings

1.3 Key Functionalities:

- **User Registration:** Users can create accounts and log in.
- **Restaurant Search:** Users can search for nearby restaurants based on their GPS location.
- **Restaurant Details:** View detailed information about restaurants.
- **User Reviews and Ratings:** Users can leave reviews and rate restaurants.
- **User Management:** Admin can manage users and view reviews.

1.4 Advanced Features:

- **Recommendation System:** Recommend restaurants based on user preferences and reviews.
- **User Profile Management:** Users can manage their profiles and view their review history.
- **Geolocation Services:** Use GPS to find nearby restaurants.

2. Database Design

2.1 ER Model:

1) Entities:

- Users, Restaurants, Reviews, Cuisines, Locations

2) Relationships:

- A user can leave multiple reviews for different restaurants.
- A restaurant can have multiple reviews from different users.
- A restaurant can belong to multiple locations and cuisines.

2.2 Attributes:

1) Users:

- **ID** (INT, PRIMARY KEY)
- **Username** (VARCHAR(50), NOT NULL)
- **Email** (VARCHAR(100), NOT NULL, UNIQUE)
- **Password** (VARCHAR(255), NOT NULL)
- **RegistrationDate** (DATETIME, DEFAULT CURRENT_TIMESTAMP)

2) Restaurants:

- **ID** (INT, PRIMARY KEY)
- **Name** (VARCHAR(100), NOT NULL)
- **Address** (VARCHAR(255), NOT NULL)
- **LocationID** (INT, FOREIGN KEY)
- **CuisineID** (INT, FOREIGN KEY)
- **Rating** (FLOAT)

3) Reviews:

- **ID** (INT, PRIMARY KEY)
- **UserID** (INT, FOREIGN KEY)
- **RestaurantID** (INT, FOREIGN KEY)
- **Rating** (INT, CHECK (Rating >= 1 AND Rating <= 5))
- **Comment** (TEXT)
- **ReviewDate** (DATETIME, DEFAULT CURRENT_TIMESTAMP)

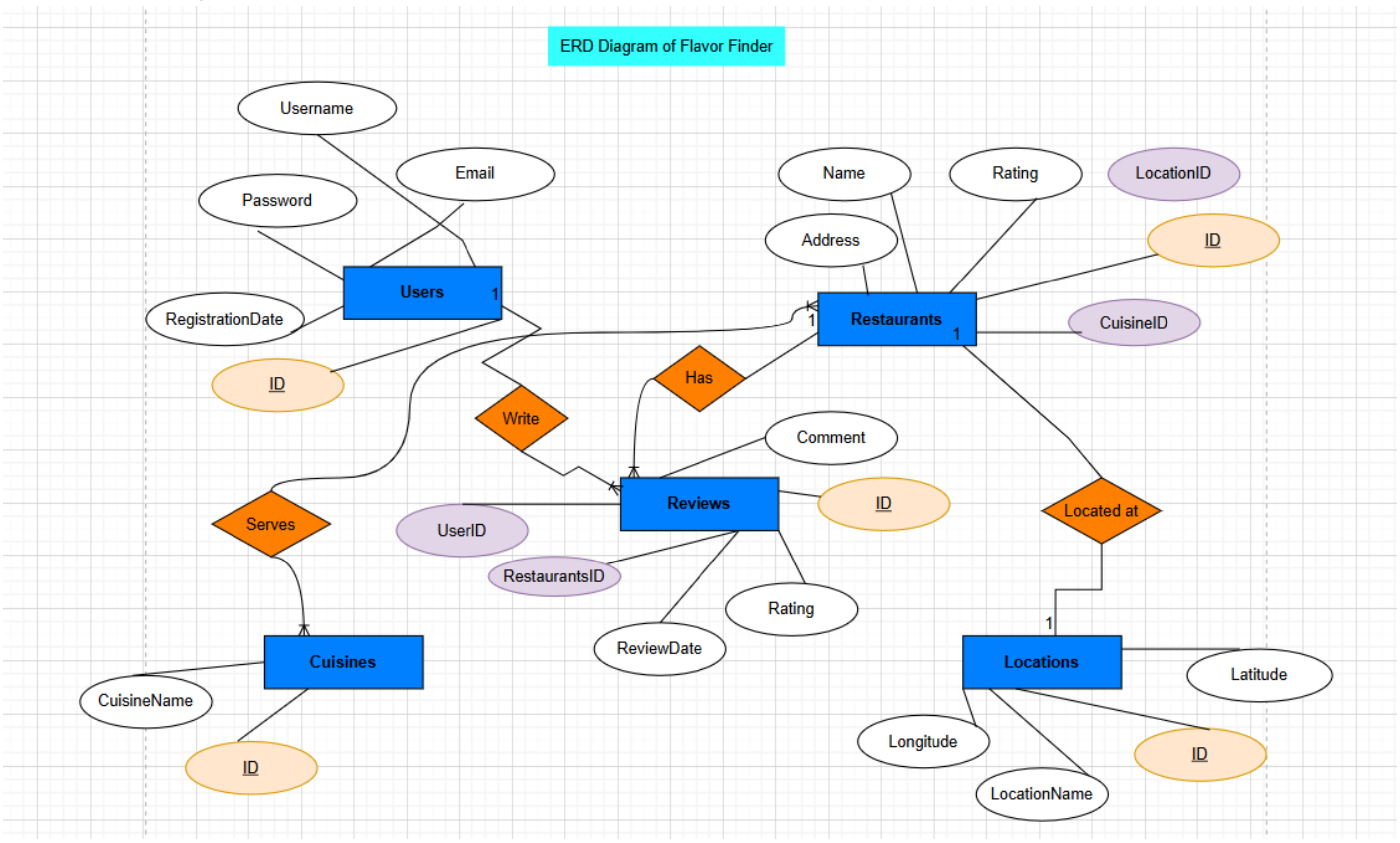
4) Cuisines:

- **ID** (INT, PRIMARY KEY)
- **CuisineName** (VARCHAR(50), NOT NULL)

5) Locations:

- ID (INT, PRIMARY KEY)
- LocationName (VARCHAR(100), NOT NULL)
- Latitude (DECIMAL(10, 8), NOT NULL)
- Longitude (DECIMAL(11, 8), NOT NULL)

2.3 ER Diagram:



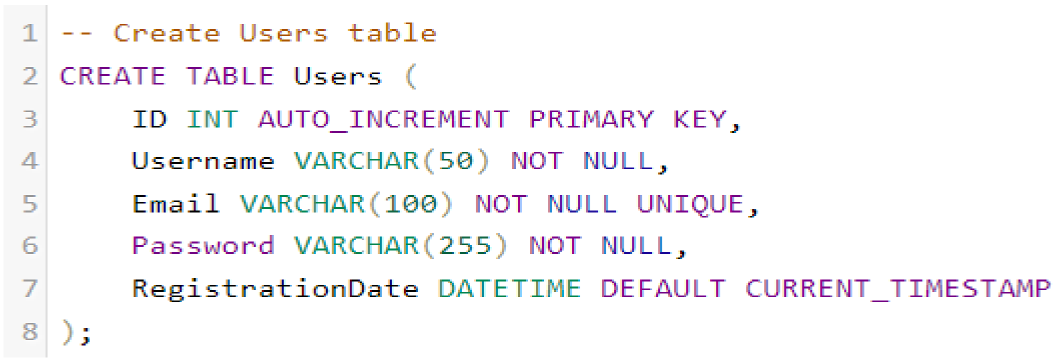
2.4 SQL for Table Creation and Data Management:

1) Creation of Database:

```sql
CREATE DATABASE restaurant_finder;
USE restaurant_finder;
```

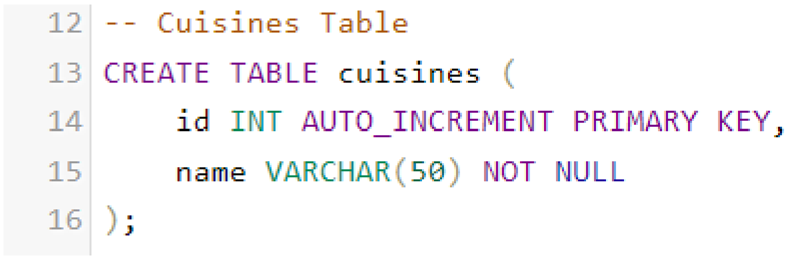
2) Creation of Tables:

- Users:



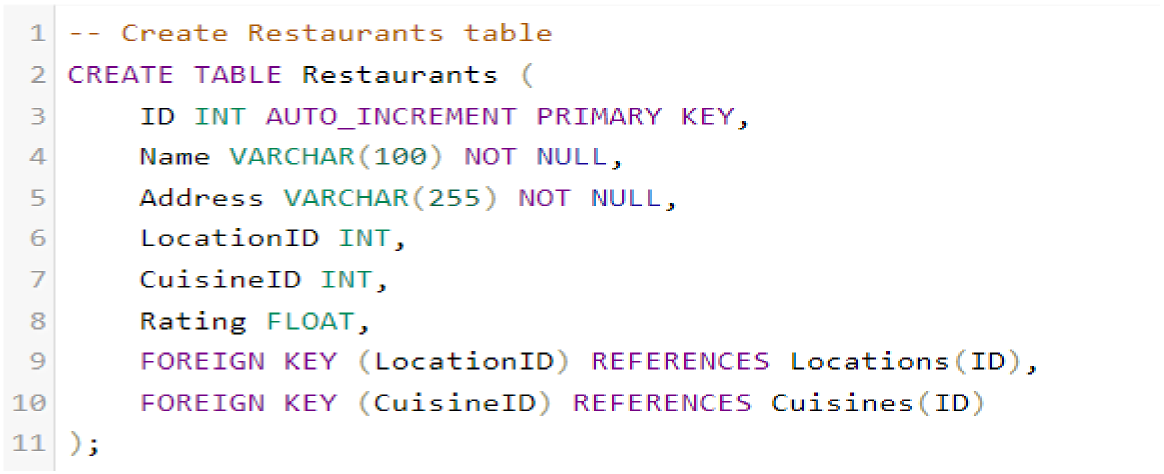
```sql
-- Create Users table
CREATE TABLE Users (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL,
    RegistrationDate DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

- Cuisines:



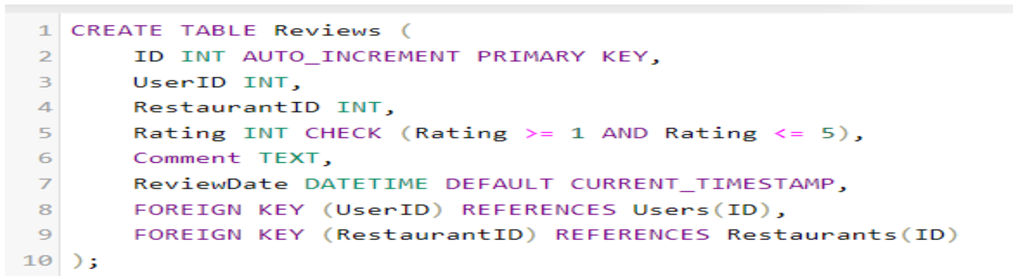
```sql
-- Cuisines Table
CREATE TABLE cuisines (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);
```

- Restaurants:



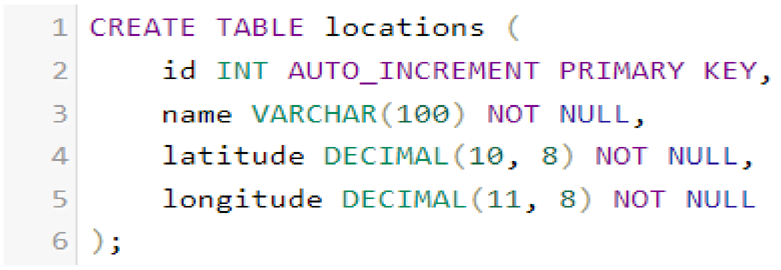
```sql
-- Create Restaurants table
CREATE TABLE Restaurants (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    LocationID INT,
    CuisineID INT,
    Rating FLOAT,
    FOREIGN KEY (LocationID) REFERENCES Locations(ID),
    FOREIGN KEY (CuisineID) REFERENCES Cuisines(ID)
);
```

- Reviews:



```sql
CREATE TABLE Reviews (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    UserID INT,
    RestaurantID INT,
    Rating INT CHECK (Rating >= 1 AND Rating <= 5),
    Comment TEXT,
    ReviewDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES Users(ID),
    FOREIGN KEY (RestaurantID) REFERENCES Restaurants(ID)
);
```

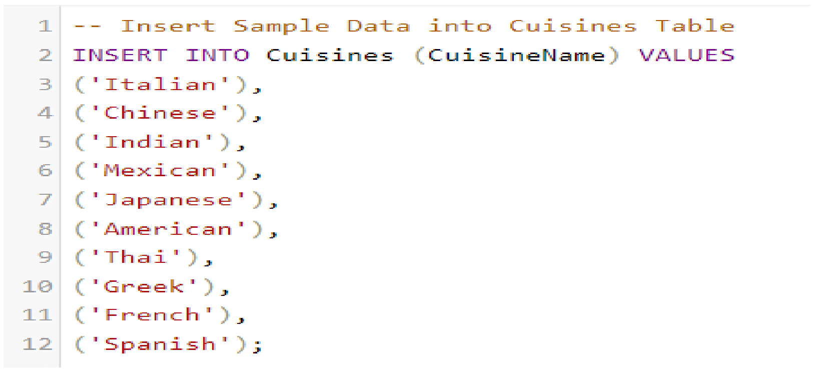
- Locations:



```sql
CREATE TABLE locations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    latitude DECIMAL(10, 8) NOT NULL,
    longitude DECIMAL(11, 8) NOT NULL
);
```

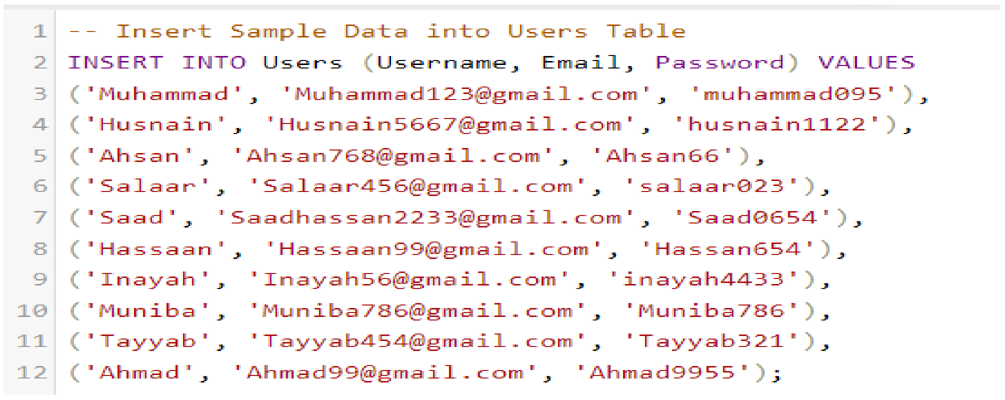
3) Inserting into Tables:

- Cuisines:



```sql
-- Insert Sample Data into Cuisines Table
INSERT INTO Cuisines (CuisineName) VALUES
('Italian'),
('Chinese'),
('Indian'),
('Mexican'),
('Japanese'),
('American'),
('Thai'),
('Greek'),
('French'),
('Spanish');
```

- Users:



```sql
-- Insert Sample Data into Users Table
INSERT INTO Users (Username, Email, Password) VALUES
('Muhammad', 'Muhammad123@gmail.com', 'muhammad095'),
('Husnain', 'Husnain5667@gmail.com', 'husnain1122'),
('Ahsan', 'Ahsan768@gmail.com', 'Ahsan66'),
('Salaar', 'Salaar456@gmail.com', 'salaar023'),
('Saad', 'Saadhassan2233@gmail.com', 'Saad0654'),
('Hassaan', 'Hassaan99@gmail.com', 'Hassan654'),
('Inayah', 'Inayah56@gmail.com', 'inayah4433'),
('Muniba', 'Muniba786@gmail.com', 'Muniba786'),
('Tayyab', 'Tayyab454@gmail.com', 'Tayyab321'),
('Ahmad', 'Ahmad99@gmail.com', 'Ahmad9955');
```

- Restaurants:

```sql
-- Insert Sample Data into Restaurants Table
INSERT INTO Restaurants (Name, Address, LocationID, CuisineID, Rating) VALUES
('Luigi\'s Italian Bistro', '123 Main St', 1, 1, 4.5),
('Dragon Palace', '456 Elm St', 2, 2, 4.0),
('Taj Mahal', '789 Oak St', 3, 3, 3.5),
('El Sombrero', '101 Pine St', 4, 4, 4.2),
('Sakura Sushi', '202 Maple St', 5, 5, 4.8),
('Burger Haven', '303 Birch St', 6, 6, 3.9),
('Thai Spice', '404 Cedar St', 7, 7, 4.7),
('Olympus Taverna', '505 Spruce St', 8, 8, 3.6),
('Café de Paris', '606 Fir St', 9, 9, 4.3),
('Tapas Delight', '707 Ash St', 10, 10, 4.9);
```

- Locations:

```sql
-- Insert Sample Data into Locations Table
INSERT INTO Locations (LocationName, Latitude, Longitude) VALUES
('New York', 40.712776, -74.005974),
('Los Angeles', 34.052235, -118.243683),
('Chicago', 41.878113, -87.629799),
('Houston', 29.760427, -95.369804),
('Denver', 39.739236, -104.990251),
('Phoenix', 33.448376, -112.074036),
('San Diego', 32.715736, -117.161087),
('San Antonio', 29.424349, -98.491142),
('Philadelphia', 39.952583, -75.165222),
('San Francisco', 37.774929, -122.419418);
```

- Reviews:

```sql
-- Insert Sample Data into Reviews Table
INSERT INTO Reviews (UserID, RestaurantID, Rating, Comment) VALUES
(1, 1, 5, 'Authentic and delicious Italian cuisine!'),
(2, 2, 4, 'Great Chinese food, very flavorful.'),
(3, 3, 3, 'Indian food was okay, not very spicy.'),
(4, 4, 5, 'Loved the Mexican dishes, very tasty!'),
(5, 5, 4, 'Fresh and delicious Japanese sushi.'),
(6, 6, 3, 'Typical American food, nothing special.'),
(7, 7, 4, 'Nice and spicy Thai food.'),
(8, 8, 5, 'Amazing Greek food, loved the gyros!'),
(9, 9, 4, 'Elegant French cuisine, very good.'),
(10, 10, 5, 'Fantastic Spanish tapas, highly recommend!');
```

4) Normalization:

1) 1st Normal Form:

- 1NF ensures that the table is flat and has no repeating groups.
- The tables provided are already in 1NF because they have atomic values and each column contains only one value.

2) 2nd Normal Form:

- 2NF ensures that the table is in 1NF and all non-key attributes are fully functional dependent on the primary key.
- The tables provided are already in 2NF because they don't have any partial dependencies.

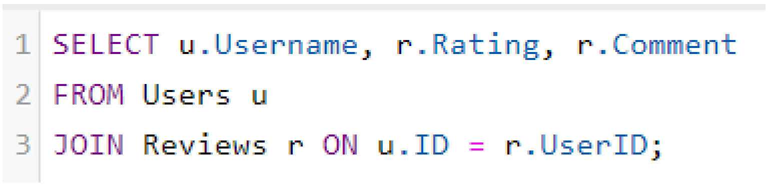
3) 3rd Normal Form:

- 3NF ensures that the table is in 2NF and all non-key attributes are non-transitively dependent on the primary key.
- The tables provided are already in 3NF because they don't have any transitive dependencies.

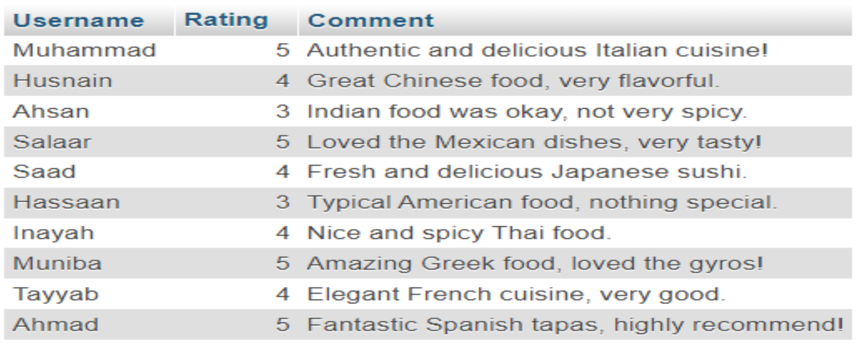
5) Data Manipulation:

1. SQL Queries

- Search:

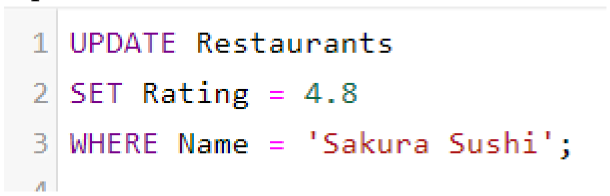


```sql
SELECT u.Username, r.Rating, r.Comment
FROM Users u
JOIN Reviews r ON u.ID = r.UserID;
```

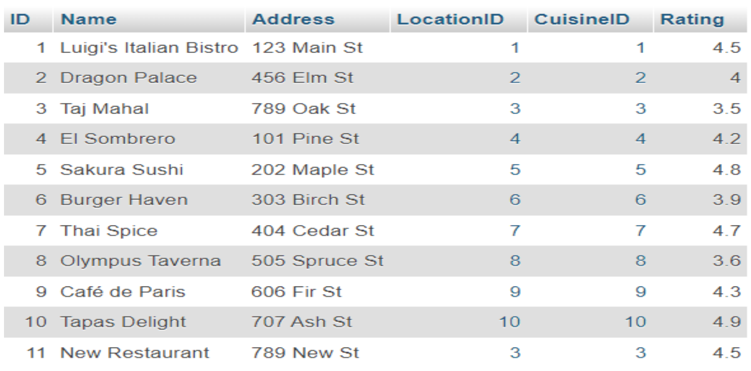


Username	Rating	Comment
Muhammad	5	Authentic and delicious Italian cuisine!
Husnain	4	Great Chinese food, very flavorful.
Ahsan	3	Indian food was okay, not very spicy.
Salaar	5	Loved the Mexican dishes, very tasty!
Saad	4	Fresh and delicious Japanese sushi.
Hassaan	3	Typical American food, nothing special.
Inayah	4	Nice and spicy Thai food.
Muniba	5	Amazing Greek food, loved the gyros!
Tayyab	4	Elegant French cuisine, very good.
Ahmad	5	Fantastic Spanish tapas, highly recommend!

- Update:



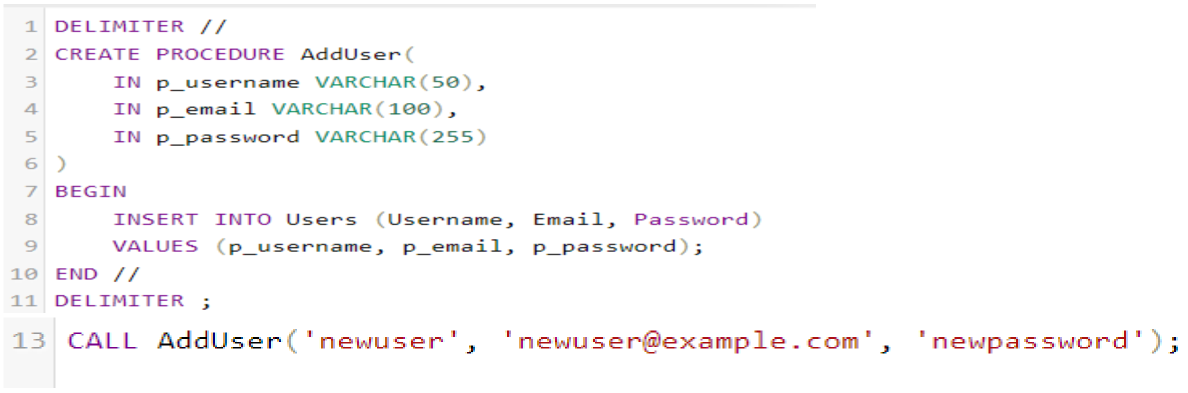
```sql
UPDATE Restaurants
SET Rating = 4.8
WHERE Name = 'Sakura Sushi';
```



ID	Name	Address	LocationID	CuisineID	Rating
1	Luigi's Italian Bistro	123 Main St	1	1	4.5
2	Dragon Palace	456 Elm St	2	2	4
3	Taj Mahal	789 Oak St	3	3	3.5
4	El Sombrero	101 Pine St	4	4	4.2
5	Sakura Sushi	202 Maple St	5	5	4.8
6	Burger Haven	303 Birch St	6	6	3.9
7	Thai Spice	404 Cedar St	7	7	4.7
8	Olympus Taverna	505 Spruce St	8	8	3.6
9	Café de Paris	606 Fir St	9	9	4.3
10	Tapas Delight	707 Ash St	10	10	4.9
11	New Restaurant	789 New St	3	3	4.5

2. Creating Store Procedure:

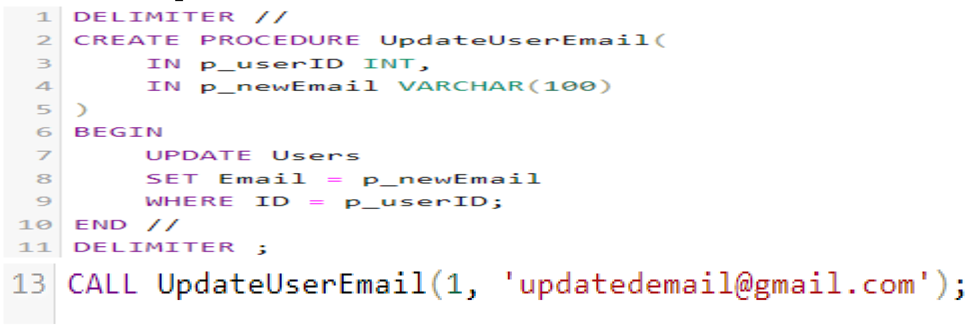
i) AddUser



```sql
DELIMITER //
CREATE PROCEDURE AddUser(
    IN p_username VARCHAR(50),
    IN p_email VARCHAR(100),
    IN p_password VARCHAR(255)
)
BEGIN
    INSERT INTO Users (Username, Email, Password)
    VALUES (p_username, p_email, p_password);
END //
DELIMITER ;
```

```sql
CALL AddUser('newuser', 'newuser@example.com', 'newpassword');
```

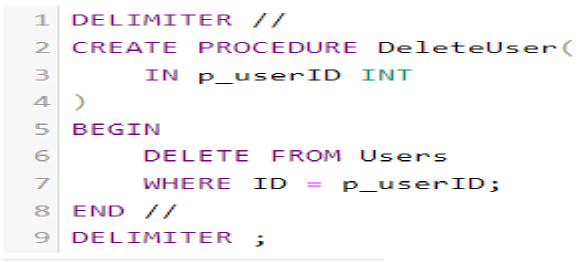
ii) UpdateUserEmail



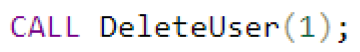
```sql
DELIMITER //
CREATE PROCEDURE UpdateUserEmail(
    IN p_userID INT,
    IN p_newEmail VARCHAR(100)
)
BEGIN
    UPDATE Users
    SET Email = p_newEmail
    WHERE ID = p_userID;
END //
DELIMITER ;
```

```sql
CALL UpdateUserEmail(1, 'updatedemail@gmail.com');
```

iii) DeleteUser

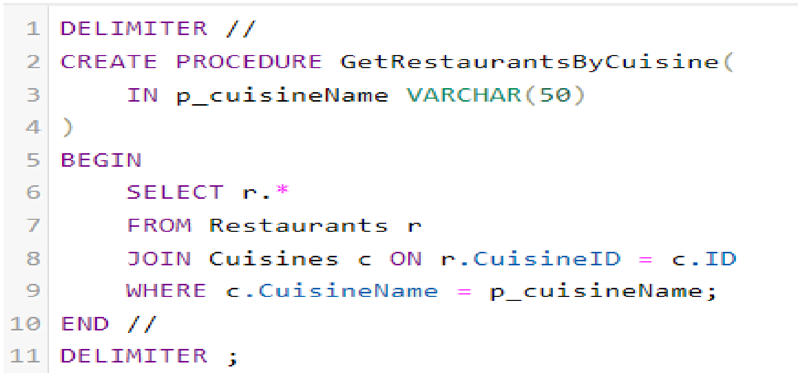


```sql
DELIMITER //
CREATE PROCEDURE DeleteUser(
    IN p_userID INT
)
BEGIN
    DELETE FROM Users
    WHERE ID = p_userID;
END //
DELIMITER ;
```

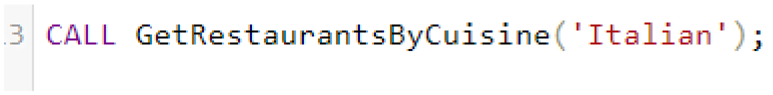


```sql
CALL DeleteUser(1);
```

iv) GetRestaurantsByCuisine



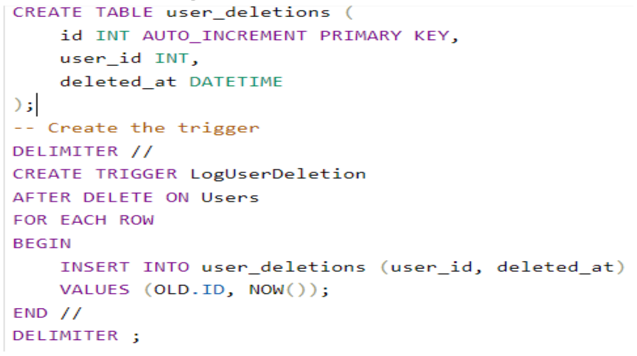
```sql
DELIMITER //
CREATE PROCEDURE GetRestaurantsByCuisine(
    IN p_cuisineName VARCHAR(50)
)
BEGIN
    SELECT r.*
    FROM Restaurants r
    JOIN Cuisines c ON r.CuisineID = c.ID
    WHERE c.CuisineName = p_cuisineName;
END //
DELIMITER ;
```



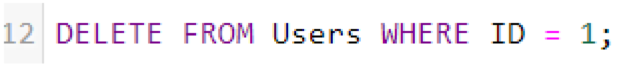
```sql
CALL GetRestaurantsByCuisine('Italian');
```

3. Creating Trigger:

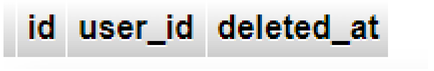
i) LogUserDeletion



```sql
CREATE TABLE user_deletions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    deleted_at DATETIME
);
-- Create the trigger
DELIMITER //
CREATE TRIGGER LogUserDeletion
AFTER DELETE ON Users
FOR EACH ROW
BEGIN
    INSERT INTO user_deletions (user_id, deleted_at)
    VALUES (OLD.ID, NOW());
END //
DELIMITER ;
```

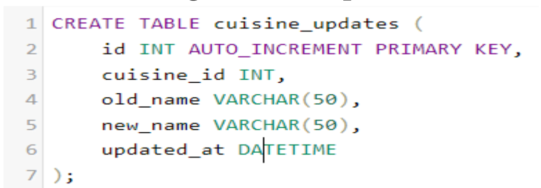


```sql
DELETE FROM Users WHERE ID = 1;
```

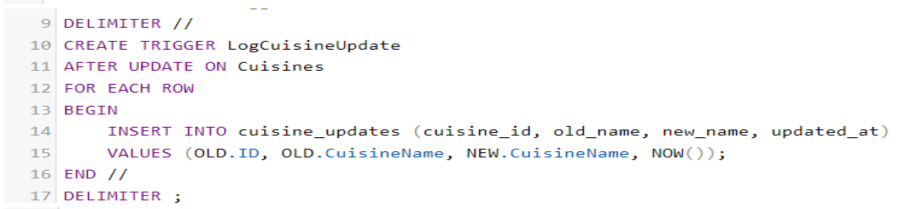


id	user_id	deleted_at

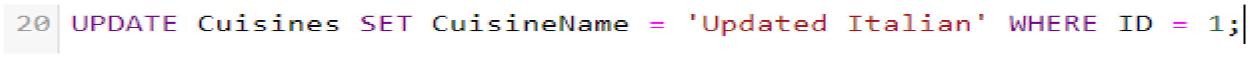
ii) LogCuisineUpdates



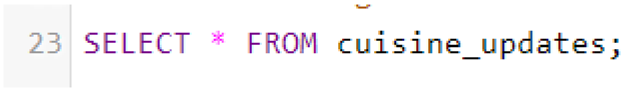
```sql
CREATE TABLE cuisine_updates (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cuisine_id INT,
    old_name VARCHAR(50),
    new_name VARCHAR(50),
    updated_at DATETIME
);
```



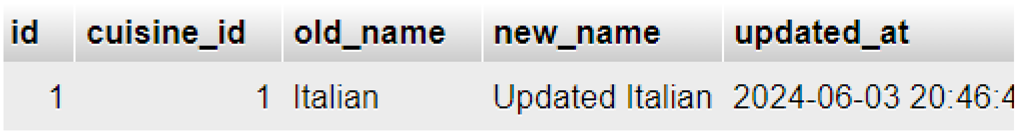
```sql
DELIMITER //
CREATE TRIGGER LogCuisineUpdate
AFTER UPDATE ON Cuisines
FOR EACH ROW
BEGIN
    INSERT INTO cuisine_updates (cuisine_id, old_name, new_name, updated_at)
    VALUES (OLD.ID, OLD.CuisineName, NEW.CuisineName, NOW());
END //
DELIMITER ;
```



```sql
UPDATE Cuisines SET CuisineName = 'Updated Italian' WHERE ID = 1;
```

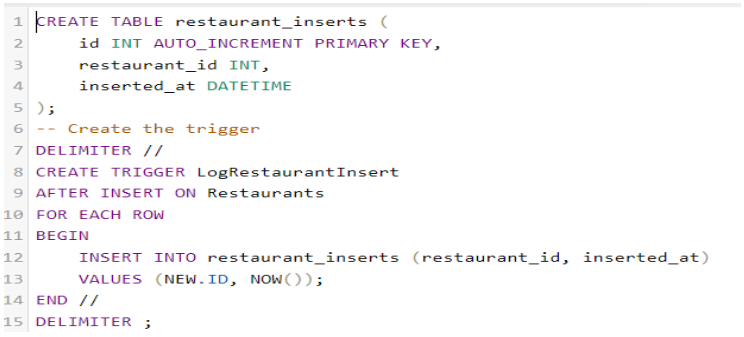


```sql
SELECT * FROM cuisine_updates;
```

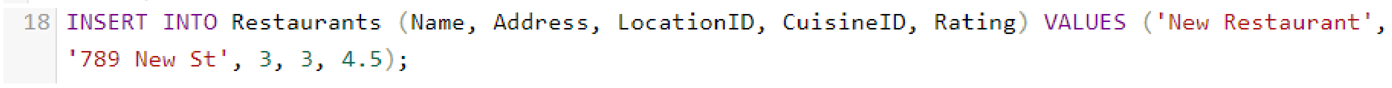


id	cuisine_id	old_name	new_name	updated_at
1	1	Italian	Updated Italian	2024-06-03 20:46:4

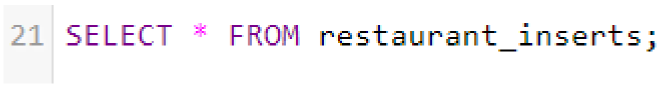
iii) RestaurantInserts



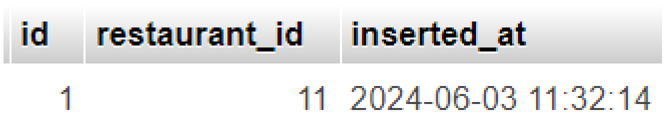
```
CREATE TABLE restaurant_inserts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    restaurant_id INT,
    inserted_at DATETIME
);
-- Create the trigger
DELIMITER //
CREATE TRIGGER LogRestaurantInsert
AFTER INSERT ON Restaurants
FOR EACH ROW
BEGIN
    INSERT INTO restaurant_inserts (restaurant_id, inserted_at)
    VALUES (NEW.ID, NOW());
END //
DELIMITER ;
```



```
INSERT INTO Restaurants (Name, Address, LocationID, CuisineID, Rating) VALUES ('New Restaurant',
'789 New St', 3, 3, 4.5);
```

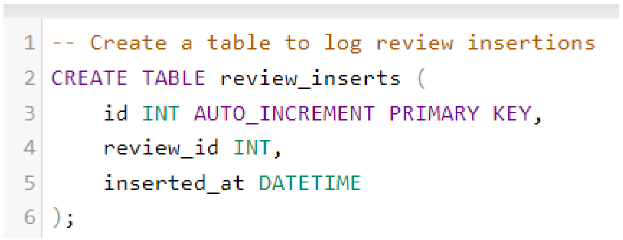


```
SELECT * FROM restaurant_inserts;
```

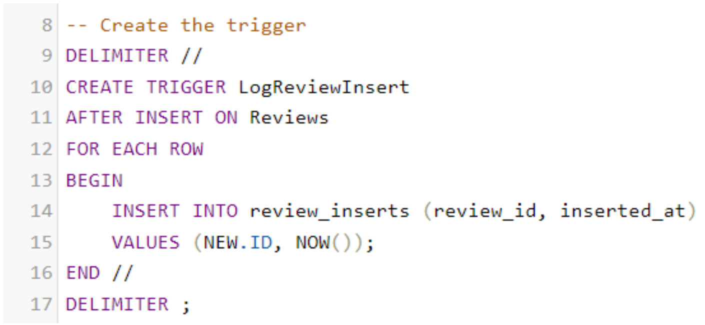


id	restaurant_id	inserted_at
1	11	2024-06-03 11:32:14

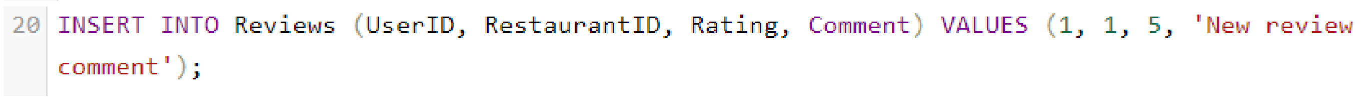
iv) LogReviewInsert



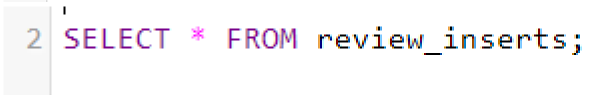
```
-- Create a table to log review insertions
CREATE TABLE review_inserts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    review_id INT,
    inserted_at DATETIME
);
```



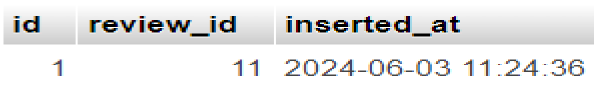
```
-- Create the trigger
DELIMITER //
CREATE TRIGGER LogReviewInsert
AFTER INSERT ON Reviews
FOR EACH ROW
BEGIN
    INSERT INTO review_inserts (review_id, inserted_at)
    VALUES (NEW.ID, NOW());
END //
DELIMITER ;
```



```
INSERT INTO Reviews (UserID, RestaurantID, Rating, Comment) VALUES (1, 1, 5, 'New review
comment');
```



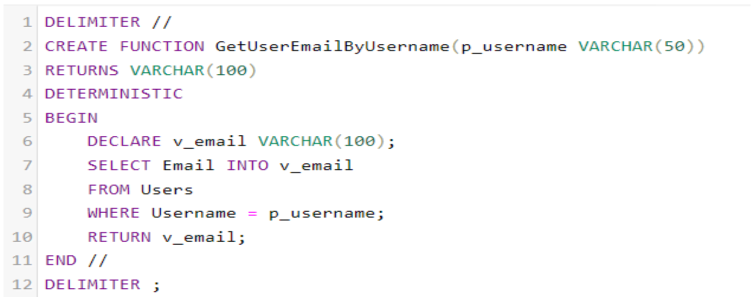
```
SELECT * FROM review_inserts;
```



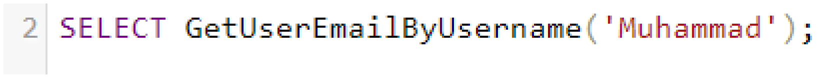
id	review_id	inserted_at
1	11	2024-06-03 11:24:36

4. Creating Function:

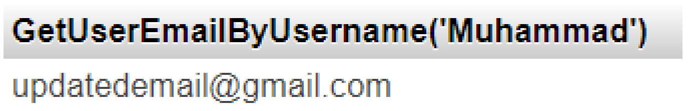
i) GetUserEmailByUsername



```sql
DELIMITER //
CREATE FUNCTION GetUserEmailByUsername(p_username VARCHAR(50))
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE v_email VARCHAR(100);
    SELECT Email INTO v_email
    FROM Users
    WHERE Username = p_username;
    RETURN v_email;
END //
DELIMITER ;
```

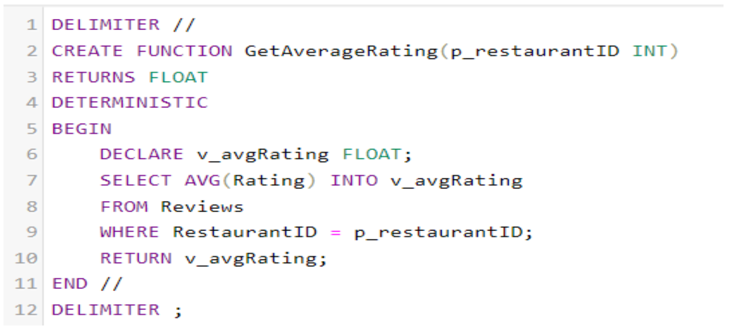


```sql
SELECT GetUserEmailByUsername('Muhammad');
```

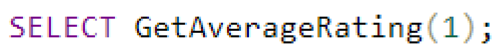


GetUserEmailByUsername('Muhammad')
updatedemail@gmail.com

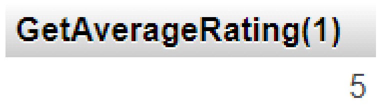
ii) GetAverageRating



```sql
DELIMITER //
CREATE FUNCTION GetAverageRating(p_restaurantID INT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE v_avgRating FLOAT;
    SELECT AVG(Rating) INTO v_avgRating
    FROM Reviews
    WHERE RestaurantID = p_restaurantID;
    RETURN v_avgRating;
END //
DELIMITER ;
```

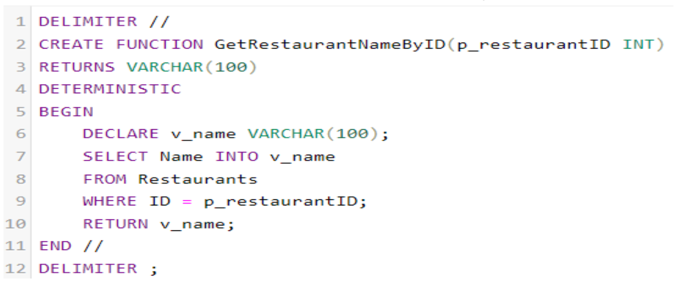


```sql
SELECT GetAverageRating(1);
```

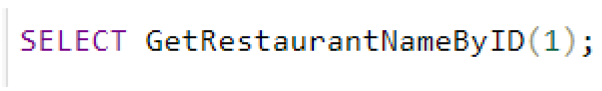


GetAverageRating(1)
5

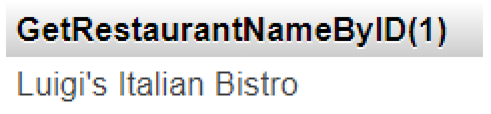
iii) GetRestaurantNameByID



```sql
DELIMITER //
CREATE FUNCTION GetRestaurantNameByID(p_restaurantID INT)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE v_name VARCHAR(100);
    SELECT Name INTO v_name
    FROM Restaurants
    WHERE ID = p_restaurantID;
    RETURN v_name;
END //
DELIMITER ;
```

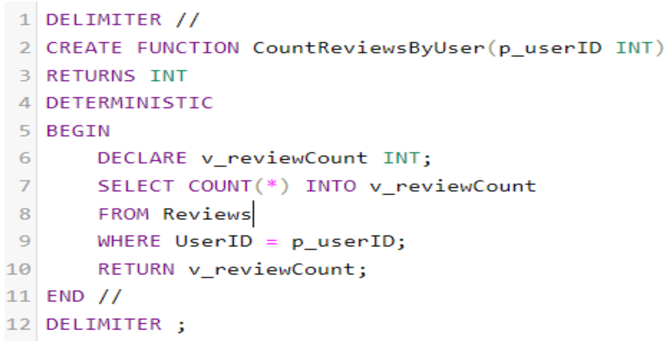


```sql
SELECT GetRestaurantNameByID(1);
```

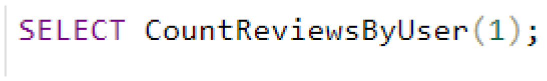


```
GetRestaurantNameByID(1)
Luigi's Italian Bistro
```

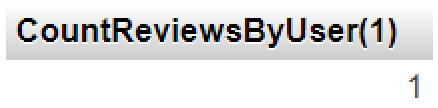
iv) CountReviewByUser



```sql
DELIMITER //
CREATE FUNCTION CountReviewsByUser(p_userID INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE v_reviewCount INT;
    SELECT COUNT(*) INTO v_reviewCount
    FROM Reviews
    WHERE UserID = p_userID;
    RETURN v_reviewCount;
END //
DELIMITER ;
```



```sql
SELECT CountReviewsByUser(1);
```



```
CountReviewsByUser(1)
                    1
```

3. Testing and Validation

3.1 Test Cases:

1) User Registration:

- **Input:** Username, Email, Password
- **Expected Result:** User record created in the Users table.

2) Restaurant Search by Location:

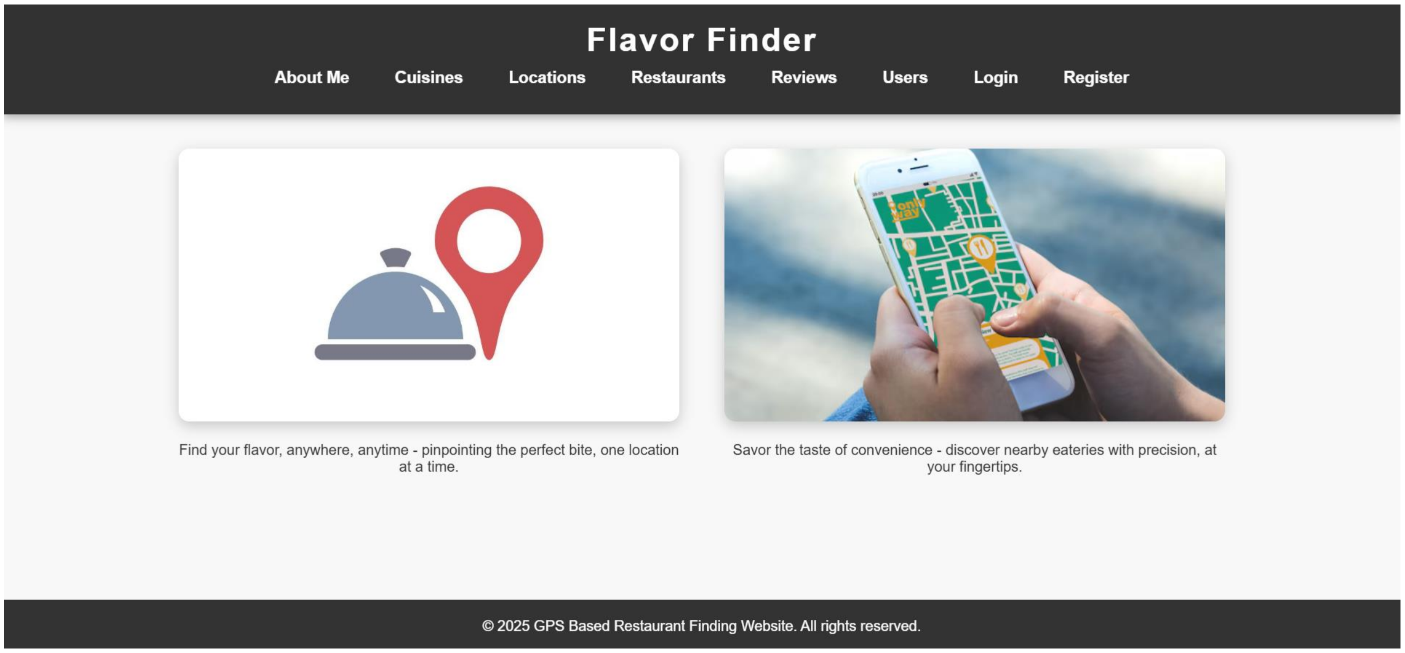
- **Input:** Location Name
- **Expected Result:** List of restaurants in the specified location.

3) Leave a Review:

- **Input:** UserID, RestaurantID, Rating, Comment
- **Expected Result:** Review record created, and restaurant rating updated.

4. Web Interface

4.1 Home Page:

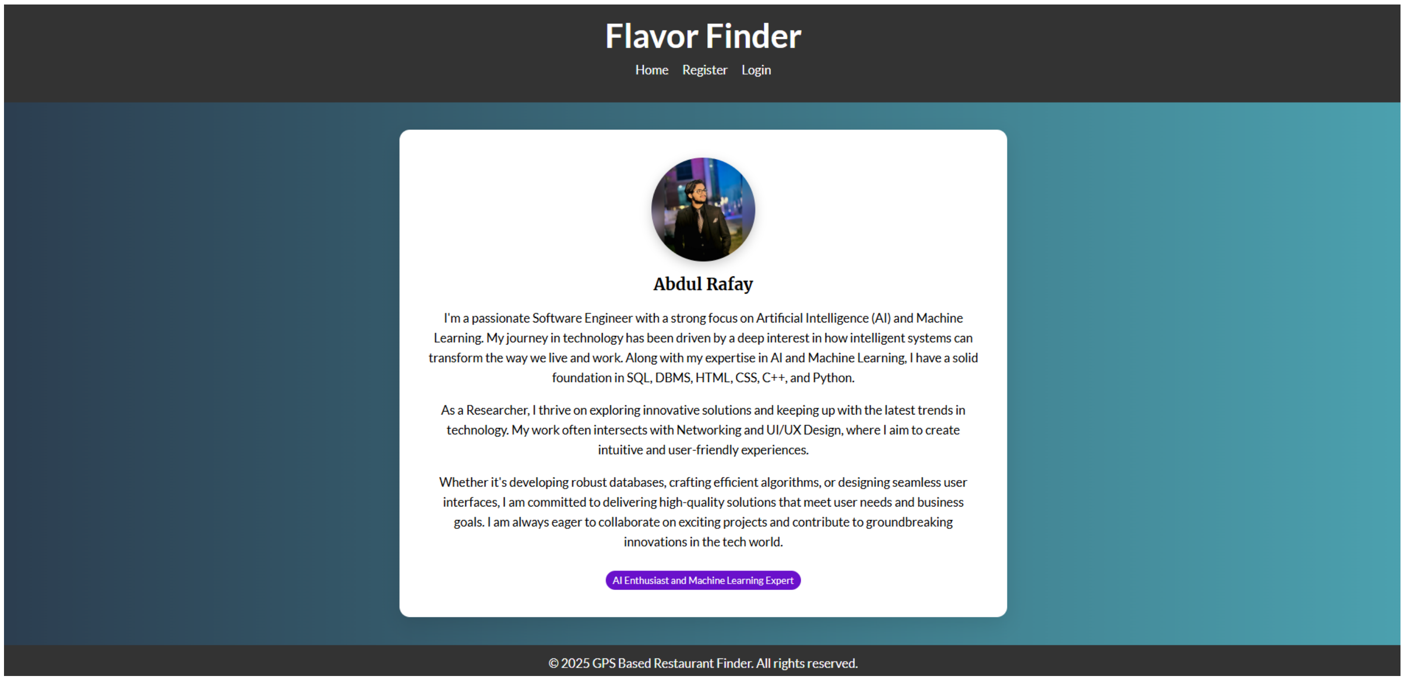


Find your flavor, anywhere, anytime - pinpointing the perfect bite, one location at a time.

Savor the taste of convenience - discover nearby eateries with precision, at your fingertips.

© 2025 GPS Based Restaurant Finding Website. All rights reserved.

4.2 About Me:



Abdul Rafay

I'm a passionate Software Engineer with a strong focus on Artificial Intelligence (AI) and Machine Learning. My journey in technology has been driven by a deep interest in how intelligent systems can transform the way we live and work. Along with my expertise in AI and Machine Learning, I have a solid foundation in SQL, DBMS, HTML, CSS, C++, and Python.

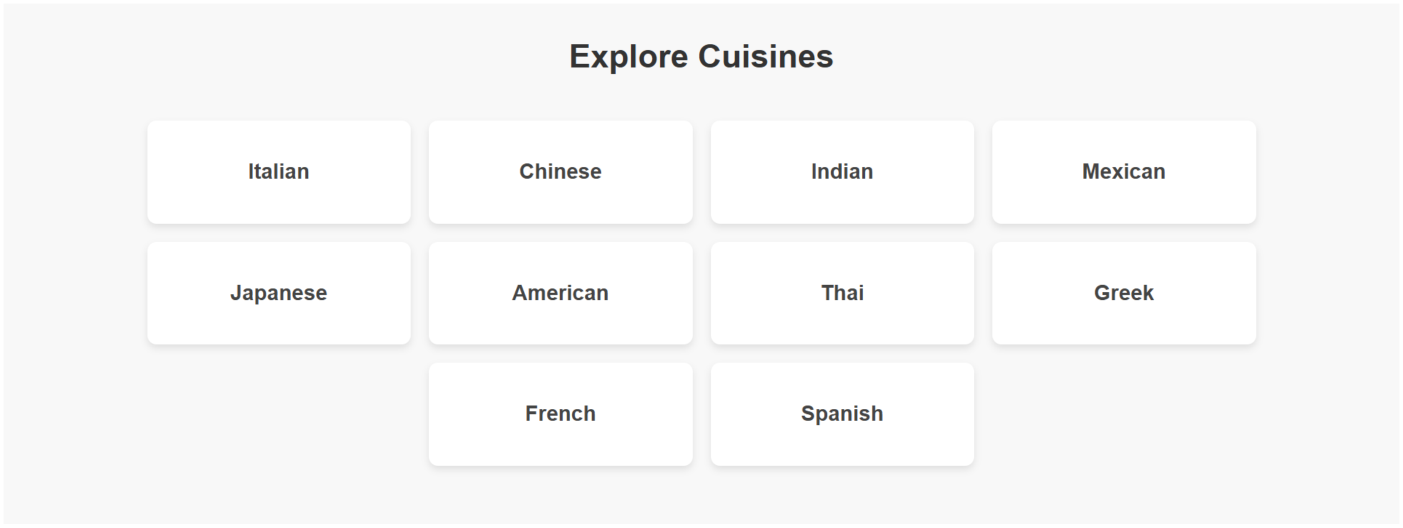
As a Researcher, I thrive on exploring innovative solutions and keeping up with the latest trends in technology. My work often intersects with Networking and UI/UX Design, where I aim to create intuitive and user-friendly experiences.

Whether it's developing robust databases, crafting efficient algorithms, or designing seamless user interfaces, I am committed to delivering high-quality solutions that meet user needs and business goals. I am always eager to collaborate on exciting projects and contribute to groundbreaking innovations in the tech world.

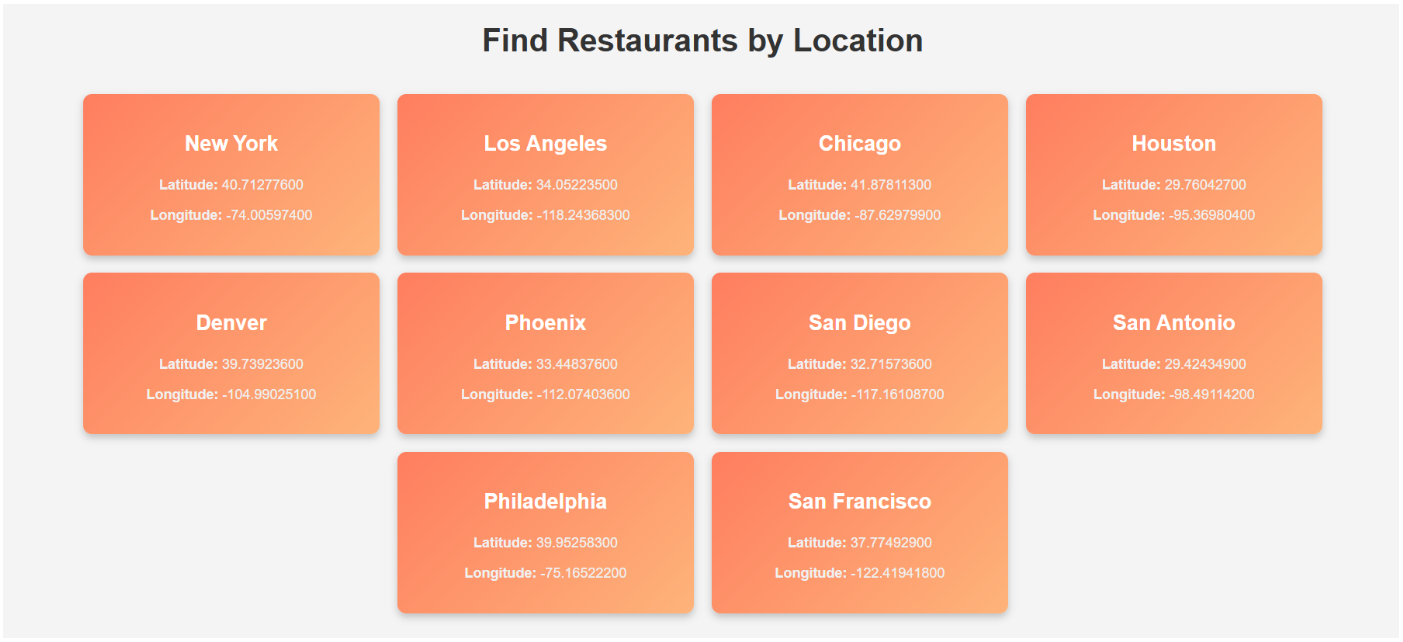
AI Enthusiast and Machine Learning Expert

© 2025 GPS Based Restaurant Finder. All rights reserved.

4.3 Explore Cuisines:



4.4 Find Restaurants by Location:


Find Restaurants by Location

New York
Latitude: 40.71277600
Longitude: -74.00597400

Los Angeles
Latitude: 34.05223500
Longitude: -118.24368300

Chicago
Latitude: 41.87811300
Longitude: -87.62979900

Houston
Latitude: 29.76042700
Longitude: -95.36980400

Denver
Latitude: 39.73923600
Longitude: -104.99025100

Phoenix
Latitude: 33.44837600
Longitude: -112.07403600

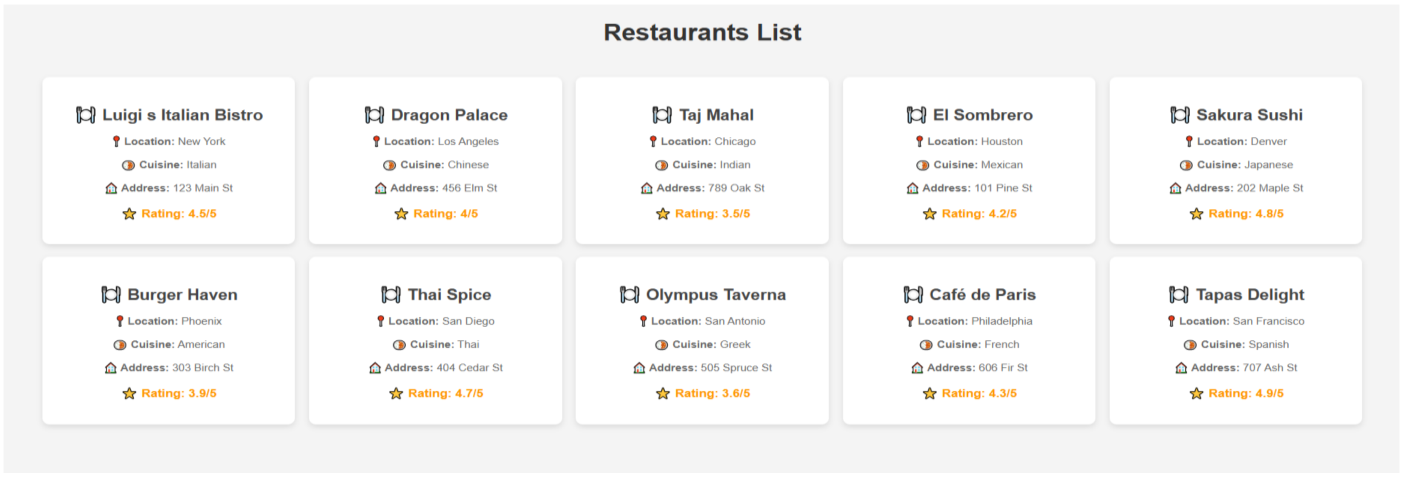
San Diego
Latitude: 32.71573600
Longitude: -117.16108700

San Antonio
Latitude: 29.42434900
Longitude: -98.49114200

Philadelphia
Latitude: 39.95258300
Longitude: -75.16522200

San Francisco
Latitude: 37.77492900
Longitude: -122.41941800

4.5 List of Restaurants:



Restaurants List

Luigi s Italian Bistro
Location: New York
Cuisine: Italian
Address: 123 Main St
Rating: 4.5/5

Dragon Palace
Location: Los Angeles
Cuisine: Chinese
Address: 456 Elm St
Rating: 4/5

Taj Mahal
Location: Chicago
Cuisine: Indian
Address: 789 Oak St
Rating: 3.5/5

El Sombrero
Location: Houston
Cuisine: Mexican
Address: 101 Pine St
Rating: 4.2/5

Sakura Sushi
Location: Denver
Cuisine: Japanese
Address: 202 Maple St
Rating: 4.8/5

Burger Haven
Location: Phoenix
Cuisine: American
Address: 303 Birch St
Rating: 3.9/5

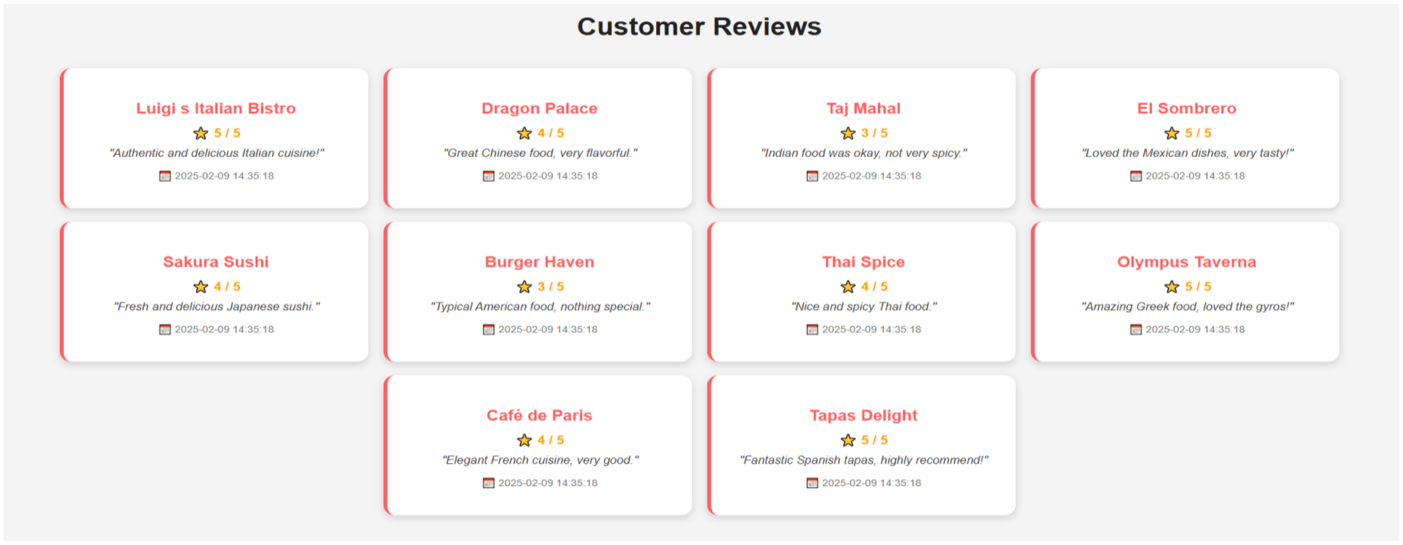
Thai Spice
Location: San Diego
Cuisine: Thai
Address: 404 Cedar St
Rating: 4.7/5

Olympus Taverna
Location: San Antonio
Cuisine: Greek
Address: 505 Spruce St
Rating: 3.6/5

Café de Paris
Location: Philadelphia
Cuisine: French
Address: 606 Fir St
Rating: 4.3/5

Tapas Delight
Location: San Francisco
Cuisine: Spanish
Address: 707 Ash St
Rating: 4.9/5

4.6 Customer Reviews:



Customer Reviews

Luigi s Italian Bistro
5 / 5
"Authentic and delicious Italian cuisine!"
2025-02-09 14:35:18

Dragon Palace
4 / 5
"Great Chinese food, very flavorful."
2025-02-09 14:35:18

Taj Mahal
3 / 5
"Indian food was okay, not very spicy."
2025-02-09 14:35:18

El Sombrero
5 / 5
"Loved the Mexican dishes, very tasty!"
2025-02-09 14:35:18

Sakura Sushi
4 / 5
"Fresh and delicious Japanese sushi."
2025-02-09 14:35:18

Burger Haven
3 / 5
"Typical American food, nothing special."
2025-02-09 14:35:18

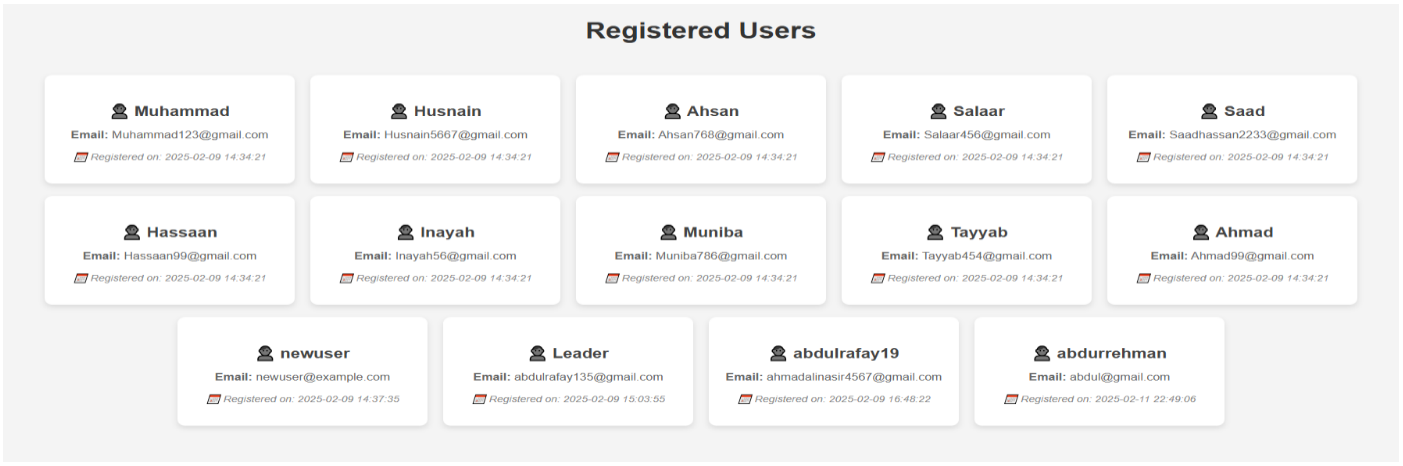
Thai Spice
4 / 5
"Nice and spicy Thai food."
2025-02-09 14:35:18

Olympus Taverna
5 / 5
"Amazing Greek food, loved the gyros!"
2025-02-09 14:35:18

Café de Paris
4 / 5
"Elegant French cuisine, very good."
2025-02-09 14:35:18

Tapas Delight
5 / 5
"Fantastic Spanish tapas, highly recommend!"
2025-02-09 14:35:18

4.7 Registered Users:



Registered Users

Muhammad
Email: Muhammad123@gmail.com
Registered on: 2025-02-09 14:34:21

Husnain
Email: Husnain5667@gmail.com
Registered on: 2025-02-09 14:34:21

Ahsan
Email: Ahsan768@gmail.com
Registered on: 2025-02-09 14:34:21

Salaar
Email: Salaar456@gmail.com
Registered on: 2025-02-09 14:34:21

Saad
Email: Saadhassan2233@gmail.com
Registered on: 2025-02-09 14:34:21

Hassaan
Email: Hassaan99@gmail.com
Registered on: 2025-02-09 14:34:21

Inayah
Email: Inayah56@gmail.com
Registered on: 2025-02-09 14:34:21

Muniba
Email: Muniba786@gmail.com
Registered on: 2025-02-09 14:34:21

Tayyab
Email: Tayyab454@gmail.com
Registered on: 2025-02-09 14:34:21

Ahmad
Email: Ahmad99@gmail.com
Registered on: 2025-02-09 14:34:21

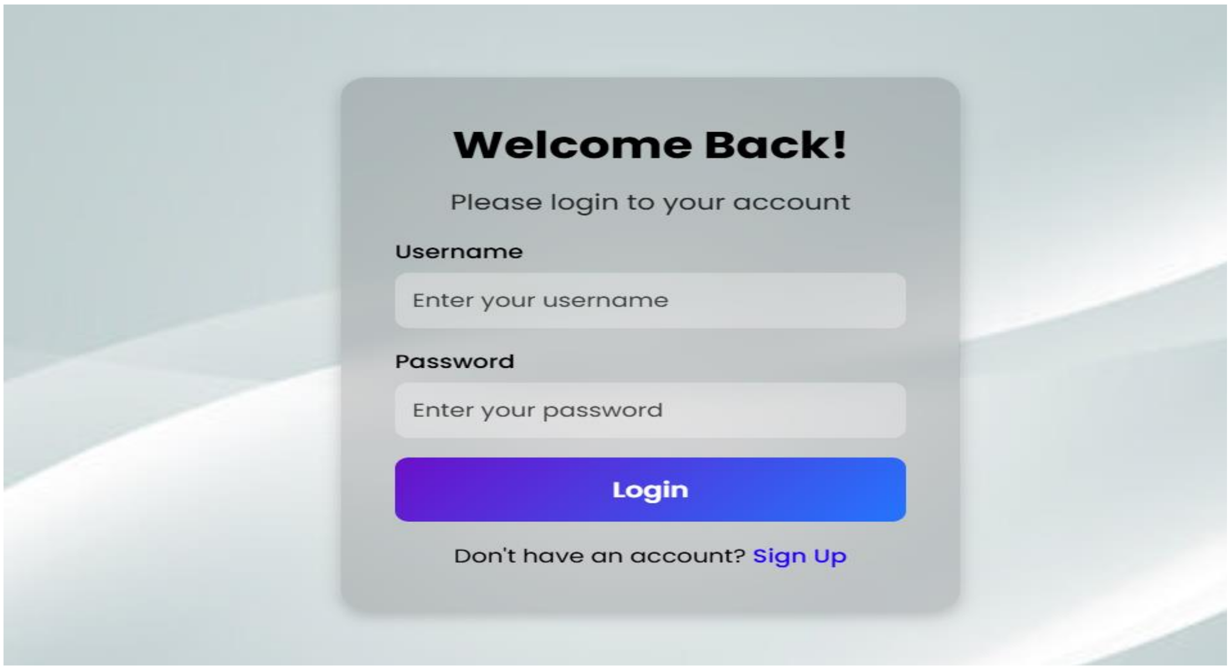
newuser
Email: newuser@example.com
Registered on: 2025-02-09 14:37:35

Leader
Email: abdulrafay135@gmail.com
Registered on: 2025-02-09 15:03:55

abdulrafay19
Email: ahmadalinasir4567@gmail.com
Registered on: 2025-02-09 16:48:22

abdurrehman
Email: abdul@gmail.com
Registered on: 2025-02-11 22:49:06

4.8 Login Page:



Welcome Back!

Please login to your account

Username

Enter your username

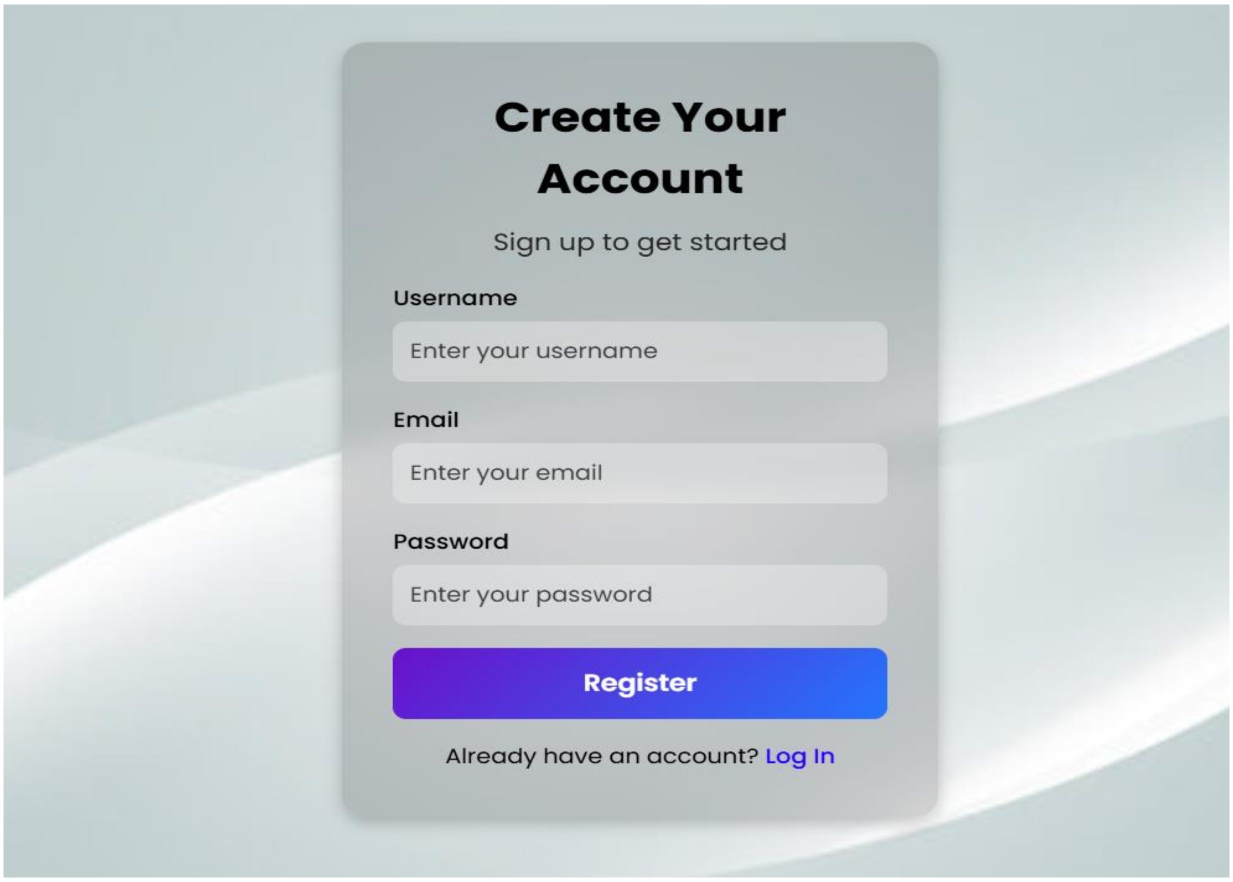
Password

Enter your password

Login

Don't have an account? Sign Up

4.9 Register Page:



Create Your Account

Sign up to get started

Username

Enter your username

Email

Enter your email

Password

Enter your password

Register

Already have an account? Log In