MYSQL QUERIES

Software used – MYSQL Workbench

Create Database

CREATE DATABASE plantstore;

```
CREATE DATABASE plantstore;
```

Use Database

use plantstore;

```
use plantstore;
```

Table – Plants

```
CREATE TABLE `plants` (
  `Plant_ID` int NOT NULL AUTO_INCREMENT,
  `Plant_Name` varchar(50) DEFAULT NULL,
  `Flower_Color` varchar(50) DEFAULT NULL,
  `Lifespan` varchar(50) DEFAULT NULL,
  `Plant_Type` varchar(50) DEFAULT NULL,
  `Bloom_Time` varchar(50) DEFAULT NULL,
  `Avg_Plant_Height_cm` int DEFAULT NULL,
  `Leaf_Type` varchar(50) DEFAULT NULL,
  `Price_INR` int DEFAULT NULL,
  PRIMARY KEY (`Plant_ID`)
);
```

```sql
CREATE TABLE `plants` (
  `Plant_ID` int NOT NULL AUTO_INCREMENT,
  `Plant_Name` varchar(50) DEFAULT NULL,
  `Flower_Color` varchar(50) DEFAULT NULL,
  `Lifespan` varchar(50) DEFAULT NULL,
  `Plant_Type` varchar(50) DEFAULT NULL,
  `Bloom_Time` varchar(50) DEFAULT NULL,
  `Avg_Plant_Height_cm` int DEFAULT NULL,
  `Leaf_Type` varchar(50) DEFAULT NULL,
  `Price_INR` int DEFAULT NULL,
  PRIMARY KEY (`Plant_ID`)
);
```

Inserting values-

INSERT INTO plants (Plant_ID, Plant_Name, Flower_Color, Lifespan, Plant_Type, Bloom_Time, Avg_Plant_Height_cm, Leaf_Type, Price_INR) VALUES

(1, 'Spider Plant', 'White', 'Perennial', 'Herb', 'Summer', 37, 'Evergreen', 160),

(2, 'Snake Plant', 'Green, White', 'Perennial', 'Succulent, Herb', 'Spring', 115, 'Evergreen', 150),

(3, 'Aloe vera', 'Yellow, Orange', 'Perennial', 'Succulent, Herb', 'Spring, Summer', 60, 'Evergreen', 200),

(4, 'Ditch lily', 'Orange, Red', 'Perennial', 'Herb', 'Spring, Summer', 95, 'Deciduous', 100),

(5, 'Dragon fruit', 'White, Green', 'Perennial', 'Succulent, Shrub, Vine', 'Spring, Summer, Fall', 900, 'Evergreen', 110),

(6, 'Flamingo flower', 'Red, Yellow', 'Perennial', 'Herb', 'All year round', 45, 'Evergreen', 125),

(7, 'Garden dahlia', 'Red, White', 'Perennial', 'Herb', 'Summer, Fall', 150, 'Deciduous', 195),

(8, 'Horseweed', 'White, Yellow', 'Annual, Biennial', 'Herb', 'Summer, Fall', 100, 'Deciduous', 160),

(9, 'Indian Laurel', 'Green', 'Perennial', 'Tree', 'Summer', 200, 'Evergreen', 175),

(10, 'Lemon balm', 'White, Purple', 'Perennial, Annual', 'Herb', 'Summer, Fall', 90, 'Deciduous', 185);

```sql
INSERT INTO plants (Plant_ID, Plant_Name, Flower_Color, Lifespan, Plant_Type, Bloom_Time, Avg_Plant_Height_cm, Leaf_Type, Price_INR) VALUES
(1, 'Spider Plant', 'White', 'Perennial', 'Herb', 'Summer', 37, 'Evergreen', 160),
(2, 'Snake Plant', 'Green, White', 'Perennial', 'Succulent, Herb', 'Spring', 115, 'Evergreen', 150),
(3, 'Aloe vera', 'Yellow, Orange', 'Perennial', 'Succulent, Herb', 'Spring, Summer', 60, 'Evergreen', 200),
(4, 'Ditch lily', 'Orange, Red', 'Perennial', 'Herb', 'Spring, Summer', 95, 'Deciduous', 100),
(5, 'Dragon fruit', 'White, Green', 'Perennial', 'Succulent, Shrub, Vine', 'Spring, Summer, Fall', 900, 'Evergreen', 110),
(6, 'Flamingo flower', 'Red, Yellow', 'Perennial', 'Herb', 'All year round', 45, 'Evergreen', 125),
(7, 'Garden dahlia', 'Red, White', 'Perennial', 'Herb', 'Summer, Fall', 150, 'Deciduous', 195),
(8, 'Horseweed', 'White, Yellow', 'Annual, Biennial', 'Herb', 'Summer, Fall', 100, 'Deciduous', 160),
(9, 'Indian Laurel', 'Green', 'Perennial', 'Tree', 'Summer', 200, 'Evergreen', 175),
(10, 'Lemon balm', 'White, Purple', 'Perennial, Annual', 'Herb', 'Summer, Fall', 90, 'Deciduous', 185);
```

Select * from plants;

```sql
select * from plants;
```

Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table – Users

```sql
CREATE TABLE `users` (
  `user_id` VARCHAR(255) NOT NULL PRIMARY KEY,
  `username` VARCHAR(50) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `role` VARCHAR(10) DEFAULT NULL
);
```

```sql
CREATE TABLE `users` (
  `user_id` VARCHAR(255) NOT NULL PRIMARY KEY,
  `username` VARCHAR(50) NOT NULL,
  `password` VARCHAR(255) NOT NULL,
  `role` VARCHAR(10) DEFAULT NULL
);
```

Inserting values

INSERT INTO users (user_id, username, password, role) VALUES

('aastha@123', 'Aastha', 'school', 'admin'),

('mansa@124', 'Mansa', 'college', 'admin');

```sql
INSERT INTO users (user_id, username, password, role) VALUES
('aastha@123', 'Aastha', 'school', 'admin'),
('mansa@124', 'Mansa', 'college', 'admin');
```

Note: Users who will sign up will be given customer role by default, as they will be customers.

Select * from users;

```sql
select * from users;
```

user_id	username	password	role
aastha@123	Aastha	school	admin
mansa@124	Mansa	college	admin
NULL	NULL	NULL	NULL

Table – Orders

```
CREATE TABLE orders (
   order_id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(100) NOT NULL,
   transaction_id VARCHAR(50) UNIQUE NOT NULL,
   price DECIMAL(10, 2) NOT NULL,
   pay_mode ENUM('Cash', 'Credit Card', 'Debit Card', 'Online') NOT NULL,
   address TEXT NOT NULL,
   order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (username) REFERENCES Users(username)
);
```

```
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) NOT NULL,
    transaction_id VARCHAR(50) UNIQUE NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    pay_mode ENUM('Cash', 'Credit Card', 'Debit Card', 'Online') NOT NULL,
    address TEXT NOT NULL,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (username) REFERENCES Users(username)
);
```

Note: No need to insert values into orders table as it will be populated automatically afterwards.

select * from orders;

```
select * from orders;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

order_id	username	transaction_id	total_price	payment_mode	delivery_address	order_date

MYSQL TABLES

Plants Table

```
mysql> select * from plants;
+----------+----------------+---------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
| Plant_ID | Plant_Name     | Flower_Color  | Lifespan         | Plant_Type            | Bloom_Time          | Avg_Plant_Height_cm | Leaf_Type | Price_INR |
+----------+----------------+---------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
|        1 | Spider Plant   | White         | Perennial        | Herb                  | Summer              |                  37 | Evergreen |       160 |
|        2 | Snake Plant    | Green, White  | Perennial        | Succulent, Herb       | Spring              |                 115 | Evergreen |       180 |
|        3 | Aloe vera      | Yellow, Orange | Perennial        | Succulent, Herb       | Spring, Summer      |                  60 | Evergreen |       200 |
|        4 | Ditch lily     | Orange, Red   | Perennial        | Herb                  | Spring, Summer      |                  95 | Deciduous |       100 |
|        5 | Dragon fruit   | White, Green  | Perennial        | Succulent, Shrub, Vine | Spring, Summer, Fall |                 900 | Evergreen |       110 |
|        6 | Flamingo flower | Red, Yellow   | Perennial        | Herb                  | All year round      |                  45 | Evergreen |       125 |
|        7 | Garden dahlia  | Red, White    | Perennial        | Herb                  | Summer, Fall        |                 150 | Deciduous |       195 |
|        8 | Horseweed      | White, Yellow | Annual, Biennial | Herb                  | Summer, Fall        |                 100 | Deciduous |       160 |
|        9 | Indian Laurel  | Green         | Perennial        | Tree                  | Summer              |                 200 | Evergreen |       175 |
|       10 | Lemon balm     | White, Purple | Perennial, Annual | Herb                  | Summer, Fall        |                  90 | Deciduous |       185 |
+----------+----------------+---------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
10 rows in set (0.00 sec)
```

Users Table

```
mysql> select * from users;
+-------------+----------+----------+----------+
| user_id     | username | password | role     |
+-------------+----------+----------+----------+
| aastha@123  | Aastha   | school   | admin    |
| bushra@345  | Bushra   | 12345    | customer |
| mansa@124   | Mansa    | college  | admin    |
| subhrat@123 | Subhrat  | 1235     | customer |
| subhrat@678 | Subhrat  | 1234     | customer |
+-------------+----------+----------+----------+
5 rows in set (0.00 sec)
```

Orders Table

```
mysql> select * from orders;
+----------+----------+----------------+-------------+--------------+-----------------+---------------------+
| order_id | username | transaction_id | total_price | payment_mode | delivery_address | order_date          |
+----------+----------+----------------+-------------+--------------+-----------------+---------------------+
|        1 | user1    | YQ4ZAWAJ2E     |      300.00 | cash         | sarita vihar    | 2024-11-12 13:39:33 |
|        2 | Samyak   | P2QNNQ6637     |     1850.00 | Cash         | Sarita Vihar    | 2024-11-14 01:11:38 |
|        3 | Bushra   | 5VF5UE943U     |     3700.00 | credit Card  | Jasola          | 2024-11-14 12:51:08 |
+----------+----------+----------------+-------------+--------------+-----------------+---------------------+
3 rows in set (0.00 sec)
```

SOURCE CODE (IDLE PYTHON 3.13)

FINAL_CS_PROJECT.py - D:/Documents/Nikita project/FINAL TO GIVE/FINAL_CS_PROJECT.py (3.13.0)

File Edit Format Run Options Window Help

```python
import mysql.connector
import random
import string
from tabulate import tabulate
import csv
import pickle

# Establish connection to the MySQL database
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="mysql",
    database="plantstore"
)
cursor = connection.cursor()

# Write a welcome message to 'welcome.txt' file
with open("welcome.txt", "w") as welcome_file:
    welcome_file.write("Welcome to Flora-Flow! Your go-to plant store.")

# Function to check if the user exists in the database
def user_exists(username, user_id=None):
    if user_id:
        cursor.execute("SELECT * FROM users WHERE username = %s AND user_id = %s", (username, user_id))
    else:
        cursor.execute("SELECT * FROM users WHERE username = %s", (username,))
    return cursor.fetchone()

# Function to log activities in binary format to 'activity_log.bin'
def log_activity_binary(activity):
    with open("activity_log.bin", "ab") as f:
        pickle.dump(activity, f)
```

```python
# Function to sign up a new user and automatically assign the 'customer' role
def sign_up():
    print("\n--- Creating Account ---\n")
    username = input("Enter a new username: ")
    user_id = input("Enter your user_id: ")  # Ask for user_id during sign-up
    if user_exists(username, user_id):
        print("Username and user_id combination already exists! Please choose a different username or user_id.")
        return None, None, None  # Return None for both username, user_id, and role if account exists

    password = input("Enter a new password: ")

    # Insert the new user with role set to "customer"
    cursor.execute("INSERT INTO users (user_id, username, password, role) VALUES (%s, %s, %s, %s)",
                   (user_id, username, password, "customer"))
    connection.commit()

    # Log the account creation activity
    log_activity_binary(f"New account created for username: {username} with role 'customer' and user_id {user_id}")
    print("Account created successfully!")

    # Update user details in CSV file
    update_users_csv()  # This will save the latest user details

    return username, user_id, "customer"  # Return both the username, user_id, and 'customer' role


# Function to export user details to a CSV file
def update_users_csv():
    cursor.execute("SELECT * FROM users")
    users = cursor.fetchall()
    with open("users.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["User_ID", "Username", "Password", "Role"])  # Define the headers
        writer.writerows(users)
    print("User details saved in users.csv.")
```

```python
# Function to log in an existing user
def log_in():
    print("\n--- Logging In ---\n")
    username = input("Enter your username: ")
    user_id = input("Enter your user_id: ")  # Ask for user_id during log-in
    password = input("Enter your password: ")
    if user_exists(username, user_id):
        cursor.execute("SELECT * FROM users WHERE username = %s AND user_id = %s AND password = %s",
                       (username, user_id, password))
        user = cursor.fetchone()
        if user:
            print("Login successful!")
            log_activity_binary(f"User {username} logged in with user_id {user_id}.")
            return username, user_id, user[3]  # Return the username, user_id, and role (role is in the 4th column)
        else:
            print("Invalid credentials. Please try again.")
            return None, None, None
    else:
        print("Invalid credentials. Please try again.")
        return None, None, None
# Function to fetch and display available plants
def show_plants():
    print("\n--- Available Plants ---\n")
    cursor.execute("SELECT * FROM plants")
    plants = cursor.fetchall()
    if plants:
        headers = ["Plant_ID", "Plant_Name", "Flower_Color", "Lifespan", "Plant_Type", "Bloom_Time", "Avg_Plant_Height_cm", "Leaf_Type", "Price_INR"]
        print(tabulate(plants, headers=headers, tablefmt="grid"))
    else:
        print("No plants available in the database.")

# Function to update plant information
def update_plant():
    plant_id = int(input("\nEnter the Plant ID to update: "))
    column = input("Enter the column to update (e.g., Price_INR): ")
    new_value = input("Enter the new value: ")
    cursor.execute(f"UPDATE plants SET {column} = %s WHERE Plant_ID = %s", (new_value, plant_id))
    connection.commit()
    print("Plant information updated successfully!")
    update_plants_csv()
    log_activity_binary(f"Plant ID {plant_id} updated. Changed {column} to {new_value}.")
```

```python
# Function to delete a plant
def delete_plant():
    plant_id = int(input("\nEnter the Plant ID to delete: "))
    cursor.execute("DELETE FROM plants WHERE Plant_ID = %s", (plant_id,))
    connection.commit()
    print("Plant deleted successfully!")
    update_plants_csv()
    log_activity_binary(f"Plant ID {plant_id} deleted.")

# Function to add a new plant
def add_plant():
    print("\n--- Add New Plant ---\n")

    # Get details of the new plant from the user
    plant_name = input("Enter Plant Name: ")
    flower_color = input("Enter Flower Color: ")
    lifespan = input("Enter Lifespan (e.g., Perennial, Annual): ")
    plant_type = input("Enter Plant Type (e.g., Herb, Shrub, Tree): ")
    bloom_time = input("Enter Bloom Time (e.g., Spring, Summer, Fall): ")
    avg_height = input("Enter Average Plant Height (in cm): ")
    leaf_type = input("Enter Leaf Type (e.g., Evergreen, Deciduous): ")
    price_inr = input("Enter Price (in INR): ")

    # Fetch all existing Plant_IDs
    cursor.execute("SELECT Plant_ID FROM plants ORDER BY Plant_ID ASC")
    plant_ids = cursor.fetchall()
    plant_ids = [plant[0] for plant in plant_ids]

    # Identify gaps in Plant_ID sequence and reuse the first available missing ID
    missing_plant_ids = [i for i in range(1, len(plant_ids) + 2) if i not in plant_ids]
    new_plant_id = missing_plant_ids[0] if missing_plant_ids else plant_ids[-1] + 1

    # Insert the new plant into the database with the determined Plant_ID
    cursor.execute("""
        INSERT INTO plants (Plant_ID, Plant_Name, Flower_Color, Lifespan, Plant_Type, Bloom_Time,
                            Avg_Plant_Height_cm, Leaf_Type, Price_INR)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """, (new_plant_id, plant_name, flower_color, lifespan, plant_type, bloom_time, avg_height, leaf_type, price_inr))

    connection.commit()
    print("New plant added successfully!")
    update_plants_csv()
    log_activity_binary(f"New plant added: {plant_name}, {plant_type}, {price_inr} INR.")
```

```python
# Function to place an order
def place_order(username):
    print("\n--- Buying Plants ---\n")
    show_plants()
    plant_id = int(input("Enter the Plant ID you want to purchase: "))
    cursor.execute("SELECT Price_INR FROM plants WHERE Plant_ID = %s", (plant_id,))
    result = cursor.fetchone()
    if result:
        price = result[0]
        quantity = int(input("Enter the quantity: "))
        total_price = price * quantity
        print(f"\nTotal Price: {total_price} INR\n")
    else:
        print("Invalid Plant ID.")
        return

    transaction_id = ''.join(random.choices(string.ascii_uppercase + string.digits, k=10))
    payment_mode = input("\nEnter payment mode (e.g., Credit Card, Cash): ")
    delivery_address = input("Enter your delivery address: ")

    # Fetch all existing order_id values and find gaps
    cursor.execute("SELECT order_id FROM orders ORDER BY order_id ASC")
    order_ids = [order[0] for order in cursor.fetchall()]

    # Find the first missing order_id in the sequence for the new order_id
    missing_order_ids = [i for i in range(1, len(order_ids) + 2) if i not in order_ids]
    new_order_id = missing_order_ids[0] if missing_order_ids else (order_ids[-1] + 1 if order_ids else 1)

    # Insert the new order into the orders table with the new_order_id
    cursor.execute("""
        INSERT INTO orders (order_id, username, transaction_id, total_price, payment_mode, delivery_address, order_date)
        VALUES (%s, %s, %s, %s, %s, %s, NOW())
    """, (new_order_id, username, transaction_id, total_price, payment_mode, delivery_address))
    connection.commit()

    print("Order placed successfully! Thank you for shopping with Flora-Flow!")
    log_activity_binary(f"Order placed by {username} with order_id: {new_order_id}, Transaction ID: {transaction_id}, Total Price: {total_price}")

    # Save the order details to orders.csv (only once)
    update_orders_csv()
    print("Order details saved successfully in orders.csv.")  # Only one print message here
```

```python
# Function to export plant details to a CSV file
def update_plants_csv():
    cursor.execute("SELECT * FROM plants")
    plants = cursor.fetchall()
    with open("plants.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["Plant_ID", "Plant_Name", "Flower_Color", "Lifespan", "Plant_Type", "Bloom_Time", "Avg_Plant_Height_cm", "Leaf_Type", "Price_INR"])
        writer.writerows(plants)
    print("Plant details saved in plants.csv.")

# Function to update orders.csv
def update_orders_csv():
    # Logic to update orders.csv
    with open('orders.csv', 'w', newline='') as csvfile:
        fieldnames = ['order_id', 'username', 'transaction_id', 'total_price', 'payment_mode', 'delivery_address', 'order_date']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()

        # Fetch rows from the database
        cursor.execute("SELECT * FROM orders")
        rows = cursor.fetchall()

        # Write each row as a dictionary matching fieldnames
        for row in rows:
            # Creating a dictionary from the row tuple
            row_dict = dict(zip(fieldnames, row))
            writer.writerow(row_dict)
```

```python
# Main program flow
username = None
user_id = None
role = None

while True:
    if username is None:
        print("\n--- Main Menu ---")
        print("1. Sign Up")
        print("2. Log In")
        print("3. Exit")
        choice = input("Select an option: ")
        if choice == "1":
            username, user_id, role = sign_up()  # Get username, user_id, and role after sign-up
            if username and user_id and role:
                print("\n--- Customer Menu ---")  # Automatically show the customer menu after sign-up
                print("1. Show Plants")
                print("2. Place Order")
                print("3. Log Out")
                choice = input("Select an option: ")
                if choice == "1":
                    show_plants()
                elif choice == "2":
                    place_order(username)
                elif choice == "3":
                    print("Logging out...")
                    username = None
                    user_id = None
                    role = None
                else:
                    print("Invalid option! Please try again.")
        elif choice == "2":
            username, user_id, role = log_in()  # Log in with username, user_id, and password
        elif choice == "3":
            print("Thanks for visiting Flora-Flow! Visit Again!")
            break
        else:
            print("Invalid option! Please try again.")
```

```python
    else:
        if role == "admin":
            print("\n--- Admin Menu ---")
            print("1. Show Plants")
            print("2. Add Plant")
            print("3. Update Plant")
            print("4. Delete Plant")
            print("5. View Orders")
            print("6. Log Out")
            choice = input("Select an option: ")
            if choice == "1":
                show_plants()
            elif choice == "2":
                add_plant()
            elif choice == "3":
                update_plant()
            elif choice == "4":
                delete_plant()
            elif choice == "5":
                cursor.execute("SELECT * FROM orders")
                orders = cursor.fetchall()
                print(tabulate(orders, headers=["Order_ID", "Username", "Transaction_ID", "Total_Price", "Payment_Mode", "Delivery_Address", "Order_Date"], tablefmt="grid"))
            elif choice == "6":
                print("Logging out...")
                username = None
                user_id = None
                role = None
            else:
                print("Invalid option! Please try again.")
        elif role == "customer":
            print("\n--- Customer Menu ---")
            print("1. Show Plants")
            print("2. Place Order")
            print("3. Log Out")
            choice = input("Select an option: ")
            if choice == "1":
                show_plants()
            elif choice == "2":
                place_order(username)
            elif choice == "3":
                print("Logging out...")
                username = None
                user_id = None
                role = None
            else:
                print("Invalid option! Please try again.")
```

```
import mysql.connector
import random
import string
from tabulate import tabulate
import csv
import pickle

# Establish connection to the MySQL database
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="mysql",
    database="plantstore"
)
cursor = connection.cursor()

# Write a welcome message to 'welcome.txt' file
with open("welcome.txt", "w") as welcome_file:
    welcome_file.write("Welcome to Flora-Flow! Your go-to plant store.")

# Function to check if the user exists in the database
def user_exists(username, user_id=None):
    if user_id:
        cursor.execute("SELECT * FROM users WHERE username = %s AND user_id = %s", (username,
user_id))
    else:
        cursor.execute("SELECT * FROM users WHERE username = %s", (username,))
    return cursor.fetchone()

# Function to log activities in binary format to 'activity_log.bin'
def log_activity_binary(activity):
```

```
    with open("activity_log.bin", "ab") as f:
        pickle.dump(activity, f)

# Function to sign up a new user and automatically assign the 'customer' role
def sign_up():
    print("\n--- Creating Account ---\n")
    username = input("Enter a new username: ")
    user_id = input("Enter your user_id: ")  # Ask for user_id during sign-up
    if user_exists(username, user_id):
        print("Username and user_id combination already exists! Please choose a different username or
user_id.")
        return None, None, None  # Return None for both username, user_id, and role if account exists

    password = input("Enter a new password: ")

    # Insert the new user with role set to "customer"
    cursor.execute("INSERT INTO users (user_id, username, password, role) VALUES (%s, %s, %s, %s)",
                   (user_id, username, password, "customer"))
    connection.commit()

    # Log the account creation activity
    log_activity_binary(f"New account created for username: {username} with role 'customer' and
user_id {user_id}")
    print("Account created successfully!")

    # Update user details in CSV file
    update_users_csv()  # This will save the latest user details

    return username, user_id, "customer"  # Return both the username, user_id, and 'customer' role

# Function to export user details to a CSV file
```

```
def update_users_csv():
    cursor.execute("SELECT * FROM users")
    users = cursor.fetchall()
    with open("users.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["User_ID", "Username", "Password", "Role"])  # Define the headers
        writer.writerows(users)
    print("User details saved in users.csv.")


# Function to log in an existing user
def log_in():
    print("\n--- Logging In ---\n")
    username = input("Enter your username: ")
    user_id = input("Enter your user_id: ")  # Ask for user_id during log-in
    password = input("Enter your password: ")
    if user_exists(username, user_id):
        cursor.execute("SELECT * FROM users WHERE username = %s AND user_id = %s AND password =
%s",
                       (username, user_id, password))
        user = cursor.fetchone()
        if user:
            print("Login successful!")
            log_activity_binary(f"User {username} logged in with user_id {user_id}.")
            return username, user_id, user[3]  # Return the username, user_id, and role (role is in the 4th
column)
        else:
            print("Invalid credentials. Please try again.")
            return None, None, None
    else:
        print("Invalid credentials. Please try again.")
        return None, None, None
# Function to fetch and display available plants
```

```
def show_plants():
    print("\n--- Available Plants ---\n")
    cursor.execute("SELECT * FROM plants")
    plants = cursor.fetchall()
    if plants:
        headers = ["Plant_ID", "Plant_Name", "Flower_Color", "Lifespan", "Plant_Type", "Bloom_Time",
"Avg_Plant_Height_cm", "Leaf_Type", "Price_INR"]
        print(tabulate(plants, headers=headers, tablefmt="grid"))
    else:
        print("No plants available in the database.")

# Function to update plant information
def update_plant():
    plant_id = int(input("\nEnter the Plant ID to update: "))
    column = input("Enter the column to update (e.g., Price_INR): ")
    new_value = input("Enter the new value: ")
    cursor.execute(f"UPDATE plants SET {column} = %s WHERE Plant_ID = %s", (new_value, plant_id))
    connection.commit()
    print("Plant information updated successfully!")
    update_plants_csv()
    log_activity_binary(f"Plant ID {plant_id} updated. Changed {column} to {new_value}.")

# Function to delete a plant
def delete_plant():
    plant_id = int(input("\nEnter the Plant ID to delete: "))
    cursor.execute("DELETE FROM plants WHERE Plant_ID = %s", (plant_id,))
    connection.commit()
    print("Plant deleted successfully!")
    update_plants_csv()
    log_activity_binary(f"Plant ID {plant_id} deleted.")
```

```
# Function to add a new plant
def add_plant():
    print("\n--- Add New Plant ---\n")

    # Get details of the new plant from the user
    plant_name = input("Enter Plant Name: ")
    flower_color = input("Enter Flower Color: ")
    lifespan = input("Enter Lifespan (e.g., Perennial, Annual): ")
    plant_type = input("Enter Plant Type (e.g., Herb, Shrub, Tree): ")
    bloom_time = input("Enter Bloom Time (e.g., Spring, Summer, Fall): ")
    avg_height = input("Enter Average Plant Height (in cm): ")
    leaf_type = input("Enter Leaf Type (e.g., Evergreen, Deciduous): ")
    price_inr = input("Enter Price (in INR): ")

    # Fetch all existing Plant_IDs
    cursor.execute("SELECT Plant_ID FROM plants ORDER BY Plant_ID ASC")
    plant_ids = cursor.fetchall()
    plant_ids = [plant[0] for plant in plant_ids]

    # Identify gaps in Plant_ID sequence and reuse the first available missing ID
    missing_plant_ids = [i for i in range(1, len(plant_ids) + 2) if i not in plant_ids]
    new_plant_id = missing_plant_ids[0] if missing_plant_ids else plant_ids[-1] + 1

    # Insert the new plant into the database with the determined Plant_ID
    cursor.execute("""
        INSERT INTO plants (Plant_ID, Plant_Name, Flower_Color, Lifespan, Plant_Type, Bloom_Time,
                            Avg_Plant_Height_cm, Leaf_Type, Price_INR)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """, (new_plant_id, plant_name, flower_color, lifespan, plant_type, bloom_time, avg_height,
leaf_type, price_inr))
```

```
    connection.commit()
    print("New plant added successfully!")
    update_plants_csv()
    log_activity_binary(f"New plant added: {plant_name}, {plant_type}, {price_inr} INR.")

# Function to place an order
def place_order(username):
    print("\n--- Buying Plants ---\n")
    show_plants()
    plant_id = int(input("Enter the Plant ID you want to purchase: "))
    cursor.execute("SELECT Price_INR FROM plants WHERE Plant_ID = %s", (plant_id,))
    result = cursor.fetchone()
    if result:
        price = result[0]
        quantity = int(input("Enter the quantity: "))
        total_price = price * quantity
        print(f"\nTotal Price: {total_price} INR\n")
    else:
        print("Invalid Plant ID.")
        return

    transaction_id = ''.join(random.choices(string.ascii_uppercase + string.digits, k=10))
    payment_mode = input("\nEnter payment mode (e.g., Credit Card, Cash): ")
    delivery_address = input("Enter your delivery address: ")

    # Fetch all existing order_id values and find gaps
    cursor.execute("SELECT order_id FROM orders ORDER BY order_id ASC")
    order_ids = [order[0] for order in cursor.fetchall()]

    # Find the first missing order_id in the sequence for the new order_id
    missing_order_ids = [i for i in range(1, len(order_ids) + 2) if i not in order_ids]
```

```
    new_order_id = missing_order_ids[0] if missing_order_ids else (order_ids[-1] + 1 if order_ids else
1)


    # Insert the new order into the orders table with the new_order_id
    cursor.execute("""
        INSERT INTO orders (order_id, username, transaction_id, total_price, payment_mode,
delivery_address, order_date)
        VALUES (%s, %s, %s, %s, %s, %s, NOW())
    """, (new_order_id, username, transaction_id, total_price, payment_mode, delivery_address))
    connection.commit()


    print("Order placed successfully! Thank you for shopping with Flora-Flow!")
    log_activity_binary(f"Order placed by {username} with order_id: {new_order_id}, Transaction ID:
{transaction_id}, Total Price: {total_price}")


    # Save the order details to orders.csv (only once)
    update_orders_csv()
    print("Order details saved successfully in orders.csv.")  # Only one print message here


# Function to export plant details to a CSV file
def update_plants_csv():
    cursor.execute("SELECT * FROM plants")
    plants = cursor.fetchall()
    with open("plants.csv", "w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["Plant_ID", "Plant_Name", "Flower_Color", "Lifespan", "Plant_Type",
"Bloom_Time", "Avg_Plant_Height_cm", "Leaf_Type", "Price_INR"])
        writer.writerows(plants)
    print("Plant details saved in plants.csv.")


# Function to update orders.csv
def update_orders_csv():
```

```
    # Logic to update orders.csv
    with open('orders.csv', 'w', newline='') as csvfile:
        fieldnames = ['order_id', 'username', 'transaction_id', 'total_price', 'payment_mode',
'delivery_address', 'order_date']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()

        # Fetch rows from the database
        cursor.execute("SELECT * FROM orders")
        rows = cursor.fetchall()

        # Write each row as a dictionary matching fieldnames
        for row in rows:
            # Creating a dictionary from the row tuple
            row_dict = dict(zip(fieldnames, row))
            writer.writerow(row_dict)


# Main program flow
username = None
user_id = None
role = None

while True:
    if username is None:
        print("\n--- Main Menu ---")
        print("1. Sign Up")
        print("2. Log In")
        print("3. Exit")
        choice = input("Select an option: ")
        if choice == "1":
```

```
        username, user_id, role = sign_up()  # Get username, user_id, and role after sign-up
        if username and user_id and role:
            print("\n--- Customer Menu ---")  # Automatically show the customer menu after sign-up
            print("1. Show Plants")
            print("2. Place Order")
            print("3. Log Out")
            choice = input("Select an option: ")
            if choice == "1":
                show_plants()
            elif choice == "2":
                place_order(username)
            elif choice == "3":
                print("Logging out...")
                username = None
                user_id = None
                role = None
            else:
                print("Invalid option! Please try again.")
    elif choice == "2":
        username, user_id, role = log_in()  # Log in with username, user_id, and password
    elif choice == "3":
        print("Thanks for visiting Flora-Flow! Visit Again!")
        break
    else:
        print("Invalid option! Please try again.")
else:
    if role == "admin":
        print("\n--- Admin Menu ---")
        print("1. Show Plants")
        print("2. Add Plant")
        print("3. Update Plant")
```

```
        print("4. Delete Plant")
        print("5. View Orders")
        print("6. Log Out")
        choice = input("Select an option: ")
        if choice == "1":
            show_plants()
        elif choice == "2":
            add_plant()
        elif choice == "3":
            update_plant()
        elif choice == "4":
            delete_plant()
        elif choice == "5":
            cursor.execute("SELECT * FROM orders")
            orders = cursor.fetchall()
            print(tabulate(orders, headers=["Order_ID", "Username", "Transaction_ID", "Total_Price",
"Payment_Mode", "Delivery_Address", "Order_Date"], tablefmt="grid"))
        elif choice == "6":
            print("Logging out...")
            username = None
            user_id = None
            role = None
        else:
            print("Invalid option! Please try again.")
    elif role == "customer":
        print("\n--- Customer Menu ---")
        print("1. Show Plants")
        print("2. Place Order")
        print("3. Log Out")
        choice = input("Select an option: ")
        if choice == "1":
```

```
        show_plants()
    elif choice == "2":
        place_order(username)
    elif choice == "3":
        print("Logging out...")
        username = None
        user_id = None
        role = None
    else:
        print("Invalid option! Please try again.")
```

OUTPUT SCREEN

Welcome Screen

```
--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option:
```

The **welcome.txt** file is automatically created when the code is run.

File Edit View

```
Welcome to Flora-Flow! Your go-to plant store.
```

welcome	14-11-2024 12:23	Text Document	1 KB

Logging in Screen

```
--- Logging In ---

Enter your username: Aastha
Enter your user_id: aastha@123
Enter your password: school
Login successful!
```

The **activity_log.bin** file is automatically created during user/admin activities and records sign-ups, orders, and logouts.

activity_log.bin	14-11-2024 12:25	BIN File	7 KB

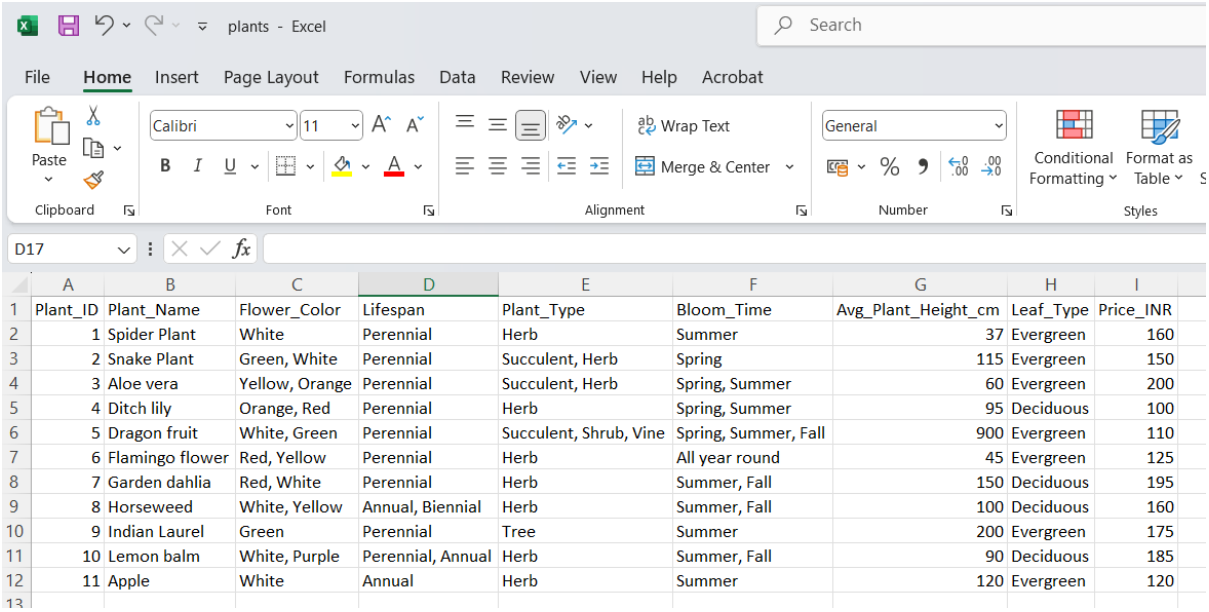
As we can see, Aastha is an admin, so it shows admin menu.

```
--- Admin Menu ---
1. Show Plants
2. Add Plant
3. Update Plant
4. Delete Plant
5. View Orders
6. Log Out
Select an option:
```

For example, I am adding a plant.

```
Select an option: 2

--- Add New Plant ---

Enter Plant Name: Apple
Enter Flower Color: White
Enter Lifespan (e.g., Perennial, Annual): Annual
Enter Plant Type (e.g., Herb, Shrub, Tree): Herb
Enter Bloom Time (e.g., Spring, Summer, Fall): Summer
Enter Average Plant Height (in cm): 120
Enter Leaf Type (e.g., Evergreen, Deciduous): Evergreen
Enter Price (in INR): 120
New plant added successfully!
Plant details saved in plants.csv.
```

Added successfully, we can see Apple is added in plants.csv and in plant details also.



Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185
11	Apple	White	Annual	Herb	Summer	120	Evergreen	120

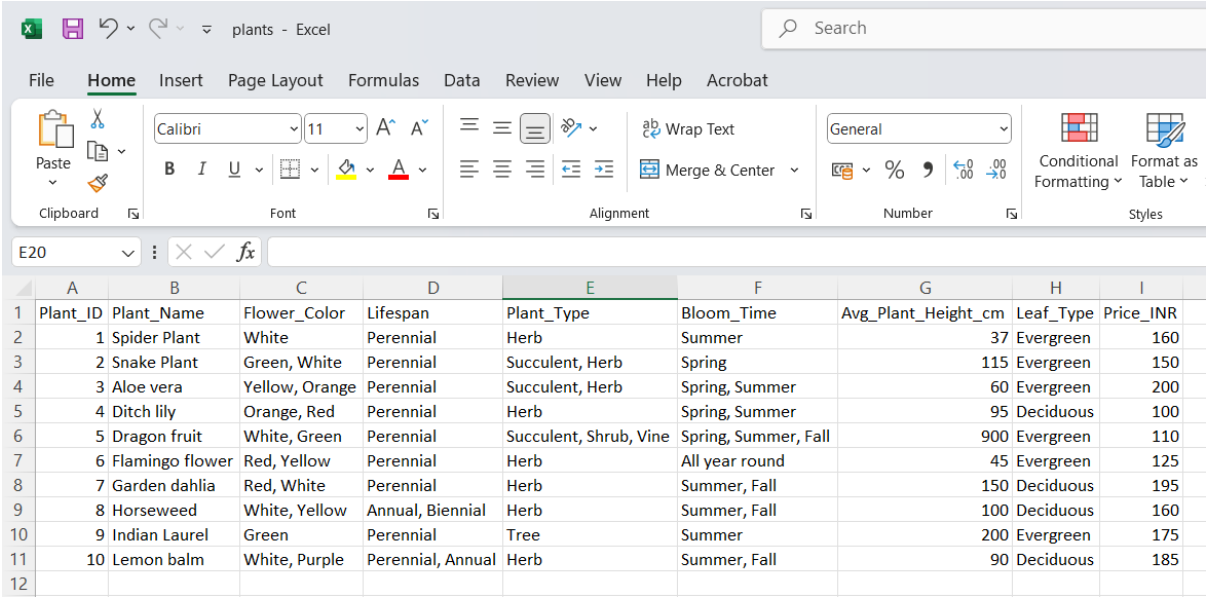
```
Select an option: 1

--- Available Plants ---
```

Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185
11	Apple	White	Annual	Herb	Summer	120	Evergreen	120

I am deleting the same plant as I am logged in as an admin.

```
--- Admin Menu ---
1. Show Plants
2. Add Plant
3. Update Plant
4. Delete Plant
5. View Orders
6. Log Out
Select an option: 4

Enter the Plant ID to delete: 11
Plant deleted successfully!
Plant details saved in plants.csv.
```

We can see that Plant ID 11 is deleted.



Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185

```
Select an option: 1

--- Available Plants ---

+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|   Plant_ID | Plant_Name      | Flower_Color   | Lifespan          | Plant_Type            | Bloom_Time          |   Avg_Plant_Height_cm | Leaf_Type   |   Price_INR |
+============+=================+================+===================+=======================+=====================+=======================+=============+=============+
|          1 | Spider Plant    | White          | Perennial         | Herb                  | Summer              |                    37 | Evergreen   |         160 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          2 | Snake Plant     | Green, White   | Perennial         | Succulent, Herb       | Spring              |                   115 | Evergreen   |         150 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          3 | Aloe vera       | Yellow, Orange | Perennial         | Succulent, Herb       | Spring, Summer      |                    60 | Evergreen   |         200 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          4 | Ditch lily      | Orange, Red    | Perennial         | Herb                  | Spring, Summer      |                    95 | Deciduous   |         100 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          5 | Dragon fruit    | White, Green   | Perennial         | Succulent, Shrub, Vine | Spring, Summer, Fall |                   900 | Evergreen   |         110 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          6 | Flamingo flower | Red, Yellow    | Perennial         | Herb                  | All year round      |                    45 | Evergreen   |         125 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          7 | Garden dahlia   | Red, White     | Perennial         | Herb                  | Summer, Fall        |                   150 | Deciduous   |         195 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          8 | Horseweed       | White, Yellow  | Annual, Biennial  | Herb                  | Summer, Fall        |                   100 | Deciduous   |         160 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|          9 | Indian Laurel   | Green          | Perennial         | Tree                  | Summer              |                   200 | Evergreen   |         175 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
|         10 | Lemon balm      | White, Purple  | Perennial, Annual | Herb                  | Summer, Fall        |                    90 | Deciduous   |         185 |
+------------+-----------------+----------------+-------------------+-----------------------+---------------------+-----------------------+-------------+-------------+
```

Updating plant information

```
--- Admin Menu ---
1. Show Plants
2. Add Plant
3. Update Plant
4. Delete Plant
5. View Orders
6. Log Out
Select an option: 3

Enter the Plant ID to update: 2
Enter the column to update (e.g., Price_INR): Price_INR
Enter the new value: 180
Plant information updated successfully!
Plant details saved in plants.csv.
```

We can see that price of Plant_ID - 2 is updated to 180 in csv and table.

Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	180
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185

```
mysql> select * from plants;
+----------+----------------+----------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
| Plant_ID | Plant_Name     | Flower_Color   | Lifespan         | Plant_Type            | Bloom_Time          | Avg_Plant_Height_cm | Leaf_Type | Price_INR |
+----------+----------------+----------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
|        1 | Spider Plant   | White          | Perennial        | Herb                  | Summer              |                  37 | Evergreen |       160 |
|        2 | Snake Plant    | Green, White   | Perennial        | Succulent, Herb       | Spring              |                 115 | Evergreen |       180 |
|        3 | Aloe vera      | Yellow, Orange | Perennial        | Succulent, Herb       | Spring, Summer      |                  60 | Evergreen |       200 |
|        4 | Ditch lily     | Orange, Red    | Perennial        | Herb                  | Spring, Summer      |                  95 | Deciduous |       100 |
|        5 | Dragon fruit   | White, Green   | Perennial        | Succulent, Shrub, Vine | Spring, Summer, Fall |                 900 | Evergreen |       110 |
|        6 | Flamingo flower | Red, Yellow    | Perennial        | Herb                  | All year round      |                  45 | Evergreen |       125 |
|        7 | Garden dahlia  | Red, White     | Perennial        | Herb                  | Summer, Fall        |                 150 | Deciduous |       195 |
|        8 | Horseweed      | White, Yellow  | Annual, Biennial | Herb                  | Summer, Fall        |                 100 | Deciduous |       160 |
|        9 | Indian Laurel  | Green          | Perennial        | Tree                  | Summer              |                 200 | Evergreen |       175 |
|       10 | Lemon balm     | White, Purple  | Perennial, Annual | Herb                  | Summer, Fall        |                  90 | Deciduous |       185 |
+----------+----------------+----------------+------------------+-----------------------+---------------------+---------------------+-----------+-----------+
10 rows in set (0.00 sec)
```

Admin can also see orders:

```
--- Admin Menu ---
1. Show Plants
2. Add Plant
3. Update Plant
4. Delete Plant
5. View Orders
6. Log Out
Select an option: 5
+------------+------------+------------------+---------------+----------------+--------------------+---------------------+
|   Order_ID | Username   | Transaction_ID   |   Total_Price | Payment_Mode   | Delivery_Address   | Order_Date          |
+============+============+==================+===============+================+====================+=====================+
|          1 | user1      | YQ4ZAWAJ2E       |           300 | cash           | sarita vihar       | 2024-11-12 13:39:33 |
+------------+------------+------------------+---------------+----------------+--------------------+---------------------+
|          2 | Samyak     | P2QNNQ6637       |          1850 | Cash           | Sarita Vihar       | 2024-11-14 01:11:38 |
+------------+------------+------------------+---------------+----------------+--------------------+---------------------+
|          3 | Bushra     | 5VF5UE943U       |          3700 | credit Card    | Jasola             | 2024-11-14 12:51:08 |
+------------+------------+------------------+---------------+----------------+--------------------+---------------------+
```

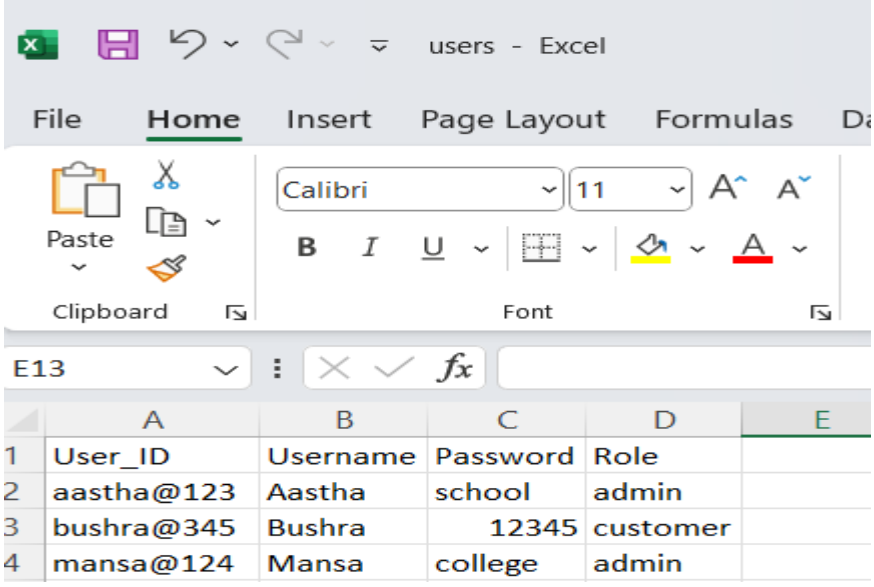
Logging out from admin. It will take us to Main Menu.

```
--- Admin Menu ---
1. Show Plants
2. Add Plant
3. Update Plant
4. Delete Plant
5. View Orders
6. Log Out
Select an option: 6
Logging out...

--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option: |
```

Creating an account

Customer will create an account and its details will be saved in **users.csv** file as well.

```
Welcome to Flora-Flow!

--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Enter your choice: 1
```



	A	B	C	D	E
1	User_ID	Username	Password	Role	
2	aastha@123	Aastha	school	admin	
3	bushra@345	Bushra	12345	customer	
4	mansa@124	Mansa	college	admin	

```
mysql> select * from users;
+-------------+----------+----------+----------+
| user_id     | username | password | role     |
+-------------+----------+----------+----------+
| aastha@123  | Aastha   | school   | admin    |
| bushra@345  | Bushra   | 12345    | customer |
| mansa@124   | Mansa    | college  | admin    |
| subhrat@678 | Subhrat  | 1234     | customer |
+-------------+----------+----------+----------+
4 rows in set (0.00 sec)
```

We can see that account created successfully and it shows customer menu for view and placing an order.

```
--- Creating Account ---

Enter a new username: Bushra
Enter your user_id: bushra@345
Enter a new password: 12345
Account created successfully!
User details saved in users.csv.

--- Customer Menu ---
1. Show Plants
2. Place Order
3. Log Out
Select an option: |
```

When we select 1, it shows all plants.

```
Select an option: 1

--- Available Plants ---
```

Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185

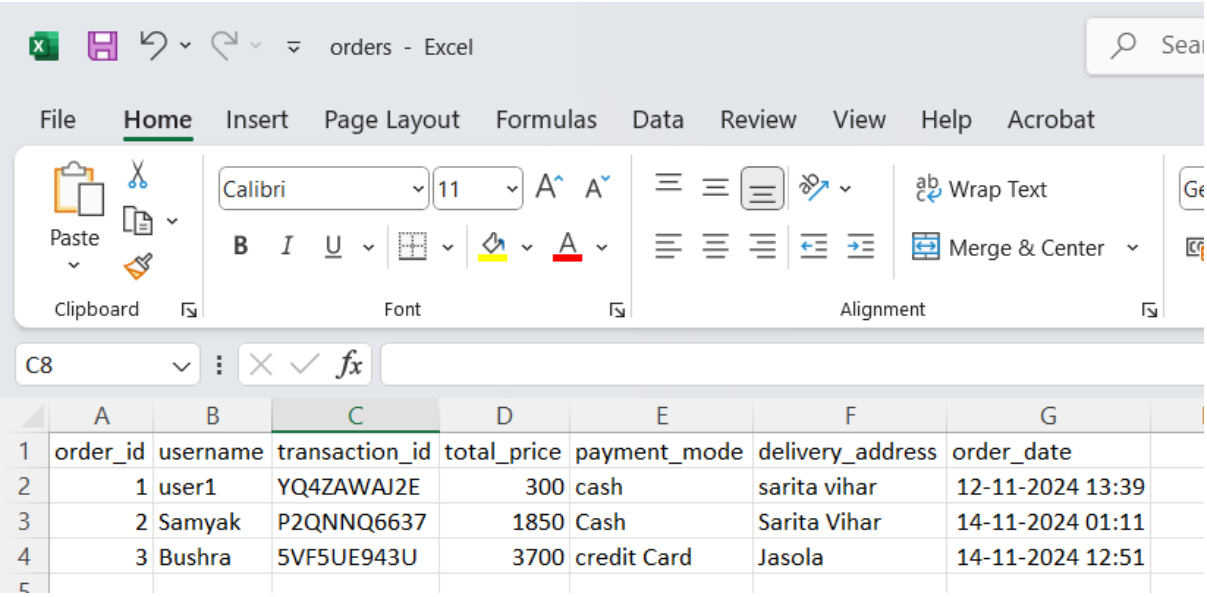
Placing an order

```
--- Customer Menu ---
1. Show Plants
2. Place Order
3. Log Out
Select an option: 2

--- Buying Plants ---


--- Available Plants ---
```

Plant_ID	Plant_Name	Flower_Color	Lifespan	Plant_Type	Bloom_Time	Avg_Plant_Height_cm	Leaf_Type	Price_INR
1	Spider Plant	White	Perennial	Herb	Summer	37	Evergreen	160
2	Snake Plant	Green, White	Perennial	Succulent, Herb	Spring	115	Evergreen	150
3	Aloe vera	Yellow, Orange	Perennial	Succulent, Herb	Spring, Summer	60	Evergreen	200
4	Ditch lily	Orange, Red	Perennial	Herb	Spring, Summer	95	Deciduous	100
5	Dragon fruit	White, Green	Perennial	Succulent, Shrub, Vine	Spring, Summer, Fall	900	Evergreen	110
6	Flamingo flower	Red, Yellow	Perennial	Herb	All year round	45	Evergreen	125
7	Garden dahlia	Red, White	Perennial	Herb	Summer, Fall	150	Deciduous	195
8	Horseweed	White, Yellow	Annual, Biennial	Herb	Summer, Fall	100	Deciduous	160
9	Indian Laurel	Green	Perennial	Tree	Summer	200	Evergreen	175
10	Lemon balm	White, Purple	Perennial, Annual	Herb	Summer, Fall	90	Deciduous	185

```
Enter the Plant ID you want to purchase: 10
Enter the quantity: 20

Total Price: 3700 INR


Enter payment mode (e.g., Credit Card, Cash): credit Card
Enter your delivery address: Jasola
Order placed successfully! Thank you for shopping with Flora-Flow!
Order details saved successfully in orders.csv.
```

Order details saved in order.csv file.



order_id	username	transaction_id	total_price	payment_mode	delivery_address	order_date
1	user1	YQ4ZAWAJ2E	300	cash	sarita vihar	12-11-2024 13:39
2	Samyak	P2QNNQ6637	1850	Cash	Sarita Vihar	14-11-2024 01:11
3	Bushra	5VF5UE943U	3700	credit Card	Jasola	14-11-2024 12:51

```
mysql> select * from orders;
+----------+----------+----------------+-------------+--------------+------------------+---------------------+
| order_id | username | transaction_id | total_price | payment_mode | delivery_address | order_date          |
+----------+----------+----------------+-------------+--------------+------------------+---------------------+
|        1 | user1    | YQ4ZAWAJ2E     |      300.00 | cash         | sarita vihar     | 2024-11-12 13:39:33 |
|        2 | Samyak   | P2QNNQ6637     |     1850.00 | Cash         | Sarita Vihar     | 2024-11-14 01:11:38 |
|        3 | Bushra   | 5VF5UE943U     |     3700.00 | credit Card  | Jasola           | 2024-11-14 12:51:08 |
+----------+----------+----------------+-------------+--------------+------------------+---------------------+
3 rows in set (0.00 sec)
```

Logged out

```
--- Customer Menu ---
1. Show Plants
2. Place Order
3. Log Out
Select an option: 3
Logging out...

--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option:
```

Invalid credentials

It will show invalid credentials if username doesn't exist while logging in.

```
--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option: 2

--- Logging In ---

Enter your username: faraz
Enter your user_id: faraz@123
Enter your password: 1234
Invalid credentials. Please try again.
```

Username exists

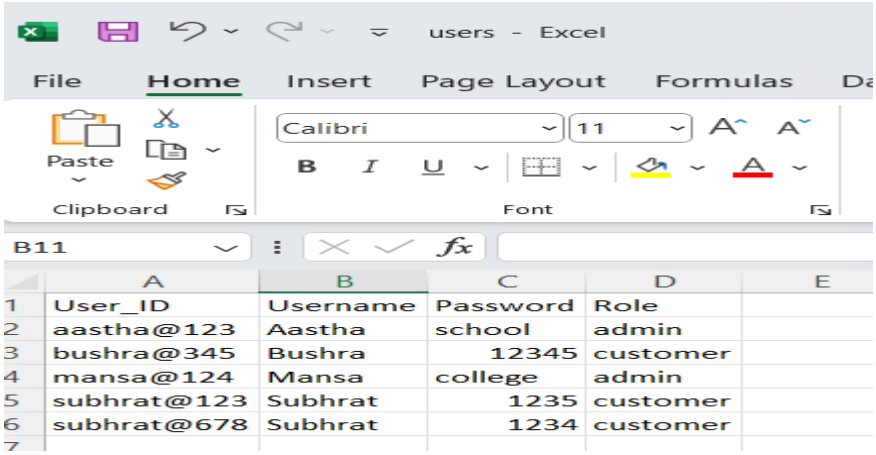
It will show username exists if user_id is same while signing up.

```
--- Creating Account ---

Enter a new username: Aastha
Enter your user_id: aastha@123
Username and user_id combination already exists! Please choose a different username or user_id.
```

Username can be same but user_id should be different

In this, we can see that username Subhrat already there in our users table, but it created an account successfully because user_id is different.

```
--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option: 1

--- Creating Account ---

Enter a new username: Subhrat
Enter your user_id: subhrat@123
Enter a new password: 1235
Account created successfully!
User details saved in users.csv.
```



User_ID	Username	Password	Role
aastha@123	Aastha	school	admin
bushra@345	Bushra	12345	customer
mansa@124	Mansa	college	admin
subhrat@123	Subhrat	1235	customer
subhrat@678	Subhrat	1234	customer

```
mysql> select * from users;
+-------------+----------+----------+----------+
| user_id     | username | password | role     |
+-------------+----------+----------+----------+
| aastha@123  | Aastha   | school   | admin    |
| bushra@345  | Bushra   | 12345    | customer |
| mansa@124   | Mansa    | college  | admin    |
| subhrat@123 | Subhrat  | 1235     | customer |
| subhrat@678 | Subhrat  | 1234     | customer |
+-------------+----------+----------+----------+
5 rows in set (0.00 sec)
```

Exit

This will exit from menu and display the message in the end.

```
--- Main Menu ---
1. Sign Up
2. Log In
3. Exit
Select an option: 3
Thanks for visiting Flora-Flow! Visit Again!
```