Menu Options

Upon running the script, you'll encounter the main menu with the following options:

1. **List all available products**: Displays a list of all available products along with their details.
2. **User Signup/Login**: Allows users to sign up for a new account or login with an existing one.
3. **Exit**: Closes the application.

User Signup/Login

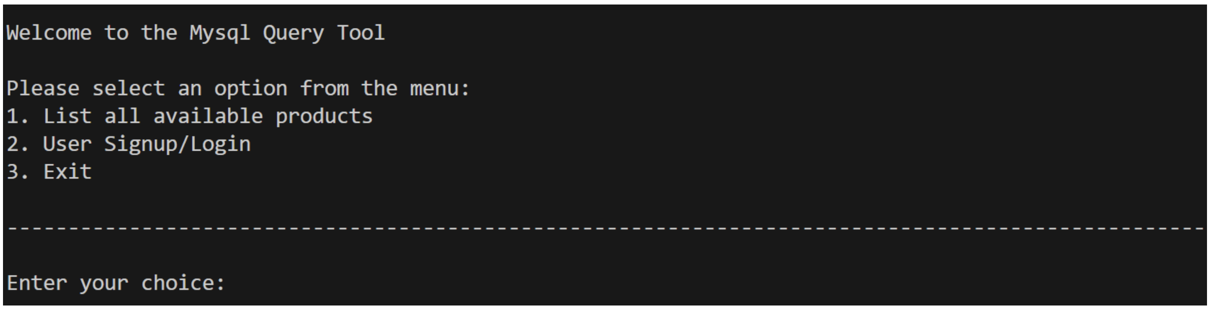
- **Signup**: Users can sign up for a new account by providing their details such as first name, last name, phone number, email, address, wallet balance, and password.
- **Login**: Existing users can log in by entering their user ID and password.

User Menu

Once logged in, users have access to the following options:

1. **View available products**: Displays a list of available products for purchase.
2. **View previous transactions**: Shows a history of previous transactions made by the user.
3. **View account balance**: Displays the current balance in the user's wallet.
4. **Buy Product**: Allows users to purchase a product by specifying the product ID.
5. **Put product up for sale**: Enables users to add a new product for sale by providing details such as name, price, description, quantity, and category.
6. **Upgrade customer status**: Allows users to upgrade their customer status to prime by paying a fee.
7. **View account details**: Shows the user's account details.
8. **Update account details**: Allows users to update their account information such as name, phone number, email, address, password, and wallet balance.
9. **Add money to wallet**: Enables users to add money to their wallet balance.
10. **Logout**: Logs the user out of the application.

CLI Screenshots

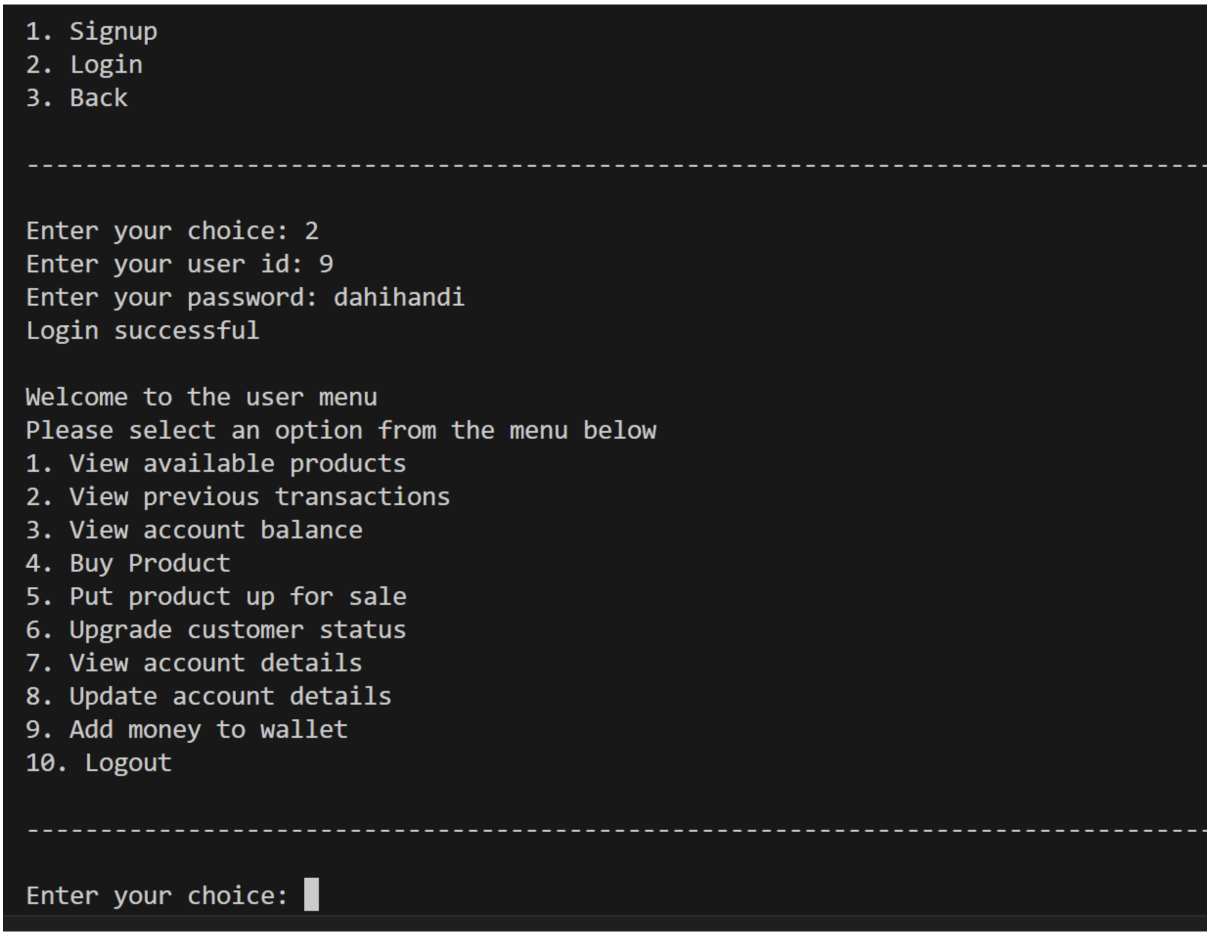


```
Welcome to the Mysql Query Tool

Please select an option from the menu:
1. List all available products
2. User Signup/Login
3. Exit

-------------------------------------------------------------------------------------------------

Enter your choice:
```



```
1. Signup
2. Login
3. Back

--------------------------------------------------------------------------------

Enter your choice: 2
Enter your user id: 9
Enter your password: dahihandi
Login successful

Welcome to the user menu
Please select an option from the menu below
1. View available products
2. View previous transactions
3. View account balance
4. Buy Product
5. Put product up for sale
6. Upgrade customer status
7. View account details
8. Update account details
9. Add money to wallet
10. Logout

--------------------------------------------------------------------------------

Enter your choice: 
```

Conflicting Transactions-

-- Conflicting Transaction 1 Simulate negative wallet balance

-- START TRANSACTION;

-- SELECT wallet INTO @current_balance FROM User WHERE user_id = 4;

-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id) VALUES ('2024-04-20', 49999, 0, 4, 1);

-- COMMIT;

-- Conflicting Transaction 2 Simulate price change

-- START TRANSACTION;

-- UPDATE Product

-- SET price = 50000

-- WHERE product_id = 1;

```
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 49999, 0, 1, 1);
-- COMMIT;

-- Conflicting Transaction 3 Simulate out of stock
-- START TRANSACTION;
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)VALUES ('2024-04-20', 49999, 0, 1, 1);
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)VALUES ('2024-04-20', 49999, 0, 1, 1);
-- select * from product;
-- COMMIT;
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)VALUES ('2024-04-20', 49999, 0, 1, 1);

-- Non Conflicting Transaction 1 different users buy item and update their wallet balnce
-- START TRANSACTION;
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 18999, 0, 1, 7);
-- UPDATE User SET wallet = wallet + 5000 WHERE user_id = 5;
-- COMMIT;

-- Non Conflicting Transaction 2 Seller price change and user buys some other product
-- START TRANSACTION;
-- UPDATE Product
-- SET price = 55000
-- WHERE product_id = 1;
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 100, 0, 1, 13);
-- COMMIT;

-- Non Conflicting Transaction 3 user buys product before Seller price change
```

```sql
-- START TRANSACTION;
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 49999, 0, 1, 1);
-- UPDATE Product
-- SET avl_status = false
-- WHERE product_id = 1;
-- COMMIT;


-- Non Conflicting Transaction 4 user lists a product to sell, someone buys it and seller is credited
-- INSERT INTO Product (name, price, description,  quantity, avl_status, seller_id, category_id, user_id)
-- VALUES ('Pixel 6a', '24999', 'Best of Google', '3', true, 19, '1',1);
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 24999, 0, 6, 14);
-- INSERT INTO Payments (Payment_Date, amount, sellbuy, user_id, product_id)
-- VALUES ('2024-04-20', 24999, 1, 1, 14);
```