Project Title:

BAKERY MANAGEMENT SYSTEM WITH MYSQL INTEGRATION

Objective:
Develop a Python-based Bakery Management System that interacts with a MySQL database to manage inventory, sales, user authentication, and reporting. The system should demonstrate basic CRUD operations, password hashing, user roles, and reporting features.

Key Learning Outcomes:

1. **Understanding of Python and MySQL integration.**
2. **Implementation of CRUD operations.**
3. **Understanding and implementing password hashing for security.**
4. **User role management and permissions.**
5. **Creating reports based on real-time data.**

Task Breakdown:

1. Setup and Environment

- **Install MySQL** and configure a database.
- Install necessary Python libraries:

```
pip install mysql-connector-python bcrypt
```

- **Database Setup**: Create a MySQL database (`bakery_management`) and required tables (`users`, `inventory`, `sales`).

2. User Authentication (Roles and Permissions)

- Implement **User Registration**: Store usernames and hashed passwords (using `bcrypt`).
- Add **User Roles**: Only allow certain actions for specific roles (e.g., `manager`, `cashier`).

3. Inventory Management

- Implement functionality to **Add, Update, and Delete items** from the inventory.
- The manager should be able to add items, and both manager and cashier should update stock quantities after sales.

4. Sales Recording

- Implement a **sales recording system** where each sale reduces inventory.

- Add sales to the database and calculate totals.

5. Report Generation

- Generate **daily, weekly, or custom sales reports** based on data in the sales table.
- Reports should display the total number of items sold and total sales for the period.

6. Password Hashing and Security

- Use `bcrypt` to hash passwords for security.
- Ensure hashed passwords are stored and checked properly during login.

7. Extra Features (Optional)

- Allow custom date range reports.
- Implement user logout functionality.
- Add exception handling for database connectivity and SQL query errors.

Project Submission Guidelines:

1. **Code Submission**: Ensure students submit their Python scripts along with an SQL script for database creation.
2. **Project Demo**: Ask students to present a demo showcasing all the functionalities.
3. **Documentation**: Each student should provide documentation detailing how they solved the problem, challenges faced, and any extra features added.
4. **Evaluation Criteria**:
   - Correctness of the implementation.
   - Quality of code and structure.
   - Report generation accuracy.
   - Security measures (password hashing).
   - Presentation of the final project.

1. Modules and Classes

- **BakeryItem (from items module):**
  - Represents individual bakery items (name, price, stock).
  - Methods:
    - add_to_db(): Adds a new item to the inventory.
    - get_item(name): Retrieves an item by name.
    - get_item_by_id(item_id): Retrieves an item by ID.
    - update_stock(quantity): Updates the stock of an item.
    - display_inventory(): Displays all items in the inventory.

    - remove_item(item_id): Removes an item from the inventory.

- **Reports (from reports module):**
  - Generates sales reports and custom reports.
  - Methods:
    - generate_report(period): Generates sales reports (daily/weekly/monthly).
    - generate_custom_report(start_date, end_date): Generates reports for a custom date range.
    - add_sale(item_name, quantity, price): Records a sale after updating stock.
- **Authentication (from auth module):**
  - Handles user authentication (register, login, logout).
  - Methods:
    - register(username, password, role): Registers a new user with a role (cashier/manager).
    - login(username, password): Authenticates a user based on username and password.
    - logout(): Logs out the current user.
    - get_role(): Retrieves the role of the logged-in user.

2. Core System Flow

- **main() Function:**
  - Handles user input and routes commands.
  - Initializes key components like Reports, Authentication, and cart.
  - Provides options for:
    1. Registration
    2. Login
    3. Add items (for managers)
    4. Update stock (for cashier or manager)
    5. Generate reports (sales/custom)
    6. Display inventory
    7. Remove items (for managers)
    8. Manage cart (add/remove items, generate bill)

9. Logout
10. Exit

3. User Roles and Permissions

- **Manager Role:**
  - Can register users.
  - Add and remove bakery items.
  - Update stock and manage inventory.
  - Generate reports.
- **Cashier Role:**
  - Can update stock when items are sold.
  - Generate sales and custom reports.
  - Manage the cart for sales transactions.

4. Cart Management

- The cart is a dictionary where each entry is an item_id with its corresponding item and quantity.
- Users can add items to the cart, remove them, and generate a bill. The stock is updated after billing.

5. Key Functionalities

- **Inventory Management:**
  - Managers can add or remove bakery items.
  - Both cashiers and managers can update stock after sales.
- **Sales Management:**
  - Reports can be generated by period (daily, weekly, monthly).
  - Custom reports can also be generated for specific date ranges.
- **User Authentication:**
  - Users must log in to access most features.
  - Only certain features (e.g., add/remove items) are restricted based on user roles.

6. Input/Output Structure

- **Input:**
  - User provides input through a series of menu choices (register, login, add items, etc.).

  - Details like item name, price, quantity, report period, etc., are collected interactively.

- **Output:**
  - Displays actions like successful login, item addition, inventory listing, and bill generation.
  - Error messages for invalid actions or insufficient permissions.

7. User Interaction Flow:

1. **Register/Login:**
   - User first registers or logs in.
   - Role (cashier/manager) is assigned, restricting access to certain features.
2. **Inventory and Cart:**
   - Manager can add or remove items.
   - Users can view items, add to the cart, and generate a bill.
3. **Reports and Sales:**
   - Sales reports are generated either for fixed periods (daily, weekly, monthly) or custom date ranges.
4. **Logout and Exit:**
   - User can log out or exit the system.

1. System Welcome Message

```
--- Bakery Management ---
1. Register
2. Login
3. Add Item
4. Update Stock
5. Sales Report
6. Custom Report
7. Display Inventory
8. Remove Item
9. Add to Cart
10. Remove from Cart
11. Generate Bill
12. Logout
13. Exit
Select an option:
```

2. User Registration Example (Option 1)

Input:

```
1 (Register)
Username: johnDoe
Password: pass123
Role (cashier/manager): manager
```

Output:

```
User johnDoe registered successfully as manager.
```

3. User Login Example (Option 2)

Input:

```
2 (Login)
Username: johnDoe
Password: pass123
```

Output:

```
Login successful! Welcome, johnDoe (manager).
```

4. Add Item to Inventory Example (Option 3)

Input (for managers only):

```
3 (Add Item)
Item Name: Croissant
Item Price: 2.5
Item Stock: 50
```

Output:

```
Item added to inventory.
```

---

5. Update Stock Example (Option 4)

Input:

```
4 (Update Stock)
Item Name: Croissant
Quantity Sold: 10
```

Output:

```
Stock updated and sale recorded for Croissant.
```

---

6. Generate Sales Report Example (Option 5)

Input:

```
5 (Sales Report)
Enter report period (daily/weekly/monthly): daily
```

Output:

```
--- Daily Sales Report ---
Croissant: Sold 10 @ $2.5 each
Total Sales: $25.00
```

---

7. Custom Report Example (Option 6)

Input:

```
6 (Custom Report)
Enter start date (YYYY-MM-DD): 2024-09-01
Enter end date (YYYY-MM-DD): 2024-09-15
```

Output:

```
--- Custom Sales Report from 2024-09-01 to 2024-09-15 ---
Croissant: Sold 10 @ $2.5 each on 2024-09-05
Total Sales: $25.00
```

---

8. Display Inventory Example (Option 7)

Input:

```
7 (Display Inventory)
```

Output:

```
--- Inventory ---
Item ID: 1, Name: Croissant, Price: $2.5, Stock: 40
```

---

9. Remove Item from Inventory Example (Option 8)

Input (for managers only):

```
8 (Remove Item)
Enter Item ID to remove: 1
```

Output:

```
Item Croissant removed from inventory.
```

---

10. Add to Cart Example (Option 9)

Input:

```
9 (Add to Cart)
Enter Item ID: 1
Enter quantity: 2
```

Output:

```
2 of Croissant added to cart.
```

---

11. Remove from Cart Example (Option 10)

Input:

```
10 (Remove from Cart)
Enter Item ID to remove from cart: 1
```

Output:

```
Item 1 removed from cart.
```

---

12. Generate Bill Example (Option 11)

Input:

```
11 (Generate Bill)
```

Output:

```
--- Bill ---
```

```
Croissant: 2 x $2.5 = $5.0
Total Amount: $5.0
Thank you for your purchase!
```

---

13. Logout Example (Option 12)

Input:

```
12 (Logout)
```

Output:

```
You have successfully logged out.
```

---

14. Exit Example (Option 13)

Input:

```
13 (Exit)
```

Output:

```
Exiting the system. Goodbye!
```

---

15. Error Handling Example (Invalid Input)

Input:

```
Enter report period (daily/weekly/monthly): yearly
```

Output:

```
Invalid period. Please enter 'daily', 'weekly', or 'monthly'.
```

16. Access Denied Example (For Unauthorized Actions)

Input:

```
3 (Add Item)
```

Output:

```
Access Denied: Only managers can add items.
```

This model output gives a detailed view of the possible interactions and system responses for different scenarios in the Bakery Management System.