[Date]

ahmad hassan

[company name]

Semester project report

NAME AHMAD HASSAN   
ROLL NO 232109

Subject OPP

Semester 2 bscsev

**SUBMITTED TO SIR kaleemullah**

**Problem statement:**

Currently, managing the pharmacy manually is slow and error-prone, leading to issues such as delayed medicine deliveries, inaccurate inventory levels, and potential risks to patient safety due to mistakes. There is a pressing need for a computerized Pharmacy Management System (PMS) to address these challenges. This system should improve efficiency by automating tasks, accurately tracking medicine inventory, simplifying prescription filling, and enhancing overall operations. Additionally, it should be capable of generating daily and monthly reports to provide insights into pharmacy performance and trends. Integration with existing healthcare systems, compliance with regulations, real-time data availability, and user-friendly interfaces are essential features required for the system to optimize workflow and ensure the best possible patient care.

**Proposed Solution:**

The current manual system for managing pharmacy operations lacks efficiency and transparency, leading to issues such as inaccurate stock management, unrecorded sales transactions, and inadequate control over user access. There is a need for a comprehensive Pharmacy Management System (PMS) that facilitates CRUD operations, user authentication, stock tracking, sales recording, and role-based access control. This system should address the challenges faced by administrators and employees in managing medicines, handling sales, and ensuring accountability while providing a user-friendly interface and robust security measures.

**MODULES:**

For the proposed Pharmacy Management System (PMS) with CRUD operations, user authentication, stock management, sales recording, and role-based access control, the system can be organized into several modules:

1. **Authentication Module**:
   * Responsible for user authentication and authorization.
   * Includes functionalities for user login, signup, and password management.
   * Ensures secure access to the system based on user roles.
2. **User Management Module**:
   * Allows administrators to manage user accounts, roles, and permissions.
   * Enables the creation, modification, and deletion of user profiles.
   * Ensures proper segregation of duties and access control.
3. **Inventory Management Module**:
   * Handles CRUD operations for medicines, including adding, updating, and deleting records.
   * Provides functionalities for checking available stock levels, receiving new stock, and adjusting quantities.
   * **Implements low stock alerts and notifications**.
4. **Sales Recording Module**:
   * Records all sales transactions, capturing details such as date, time, medicine sold, quantity, and total amount.
   * Generates sales invoices or receipts for each transaction.
   * Maintains a sales history for reporting and auditing purposes.
5. **Reporting Module**:
   * Generates daily and monthly sales reports.
   * Provides insights into sales performance, trends, and inventory turnover.
   * Allows customization of reports based on user preferences.
6. **Billing and Purchase Module**:
   * Enables users to add medicines to a virtual cart, specify quantities, and generate bills.
   * Calculates total amounts, taxes, and discounts.
   * Tracks purchase history and payments.
7. **Role-Based Access Control (RBAC) Module**:
   * Enforces role-based access control, restricting user access to specific functionalities based on their roles.
   * Defines roles such as administrator, manager, and employee, each with distinct permissions.
   * Manages role assignments and permissions.

**Concept of Opp used in project**

**Encapsulation**: Bundling data and methods within classes, restricting direct access to some components.

**Inheritance**: Allowing a class to inherit properties and methods from another class, promoting code reuse.

**Polymorphism**: Enabling objects of different classes to be treated as objects of a common superclass.

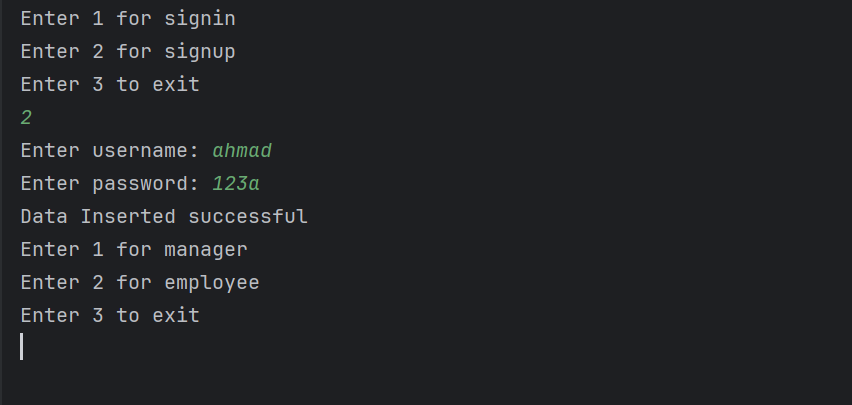
**Abstraction**: Simplifying complex reality by modeling classes based on essential characteristics.

**Composition**: Building complex objects by combining simpler objects, enhancing modularity.

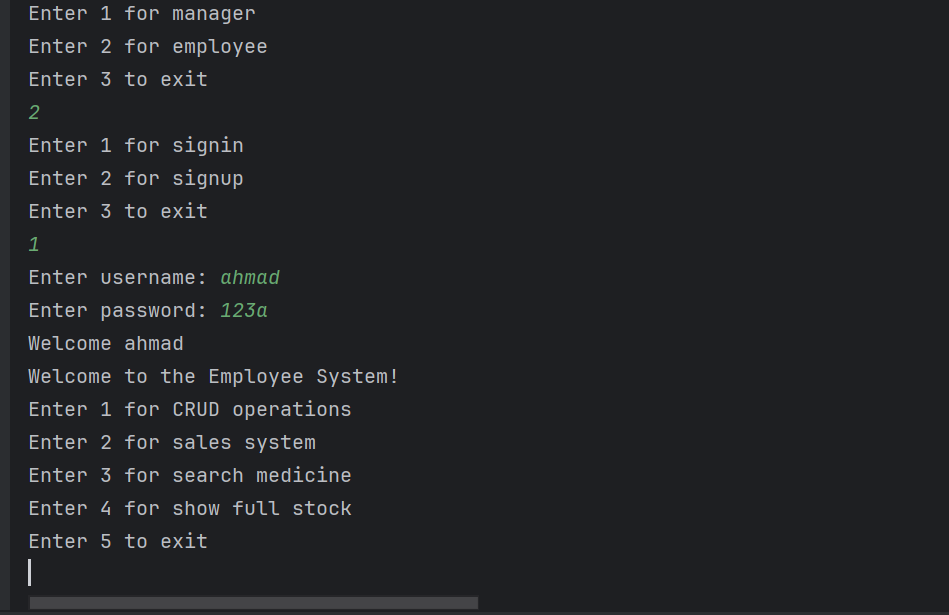
**Exception Handling**: Managing errors and exceptions gracefully to maintain system stability.

**SCREEN SHOTS:**

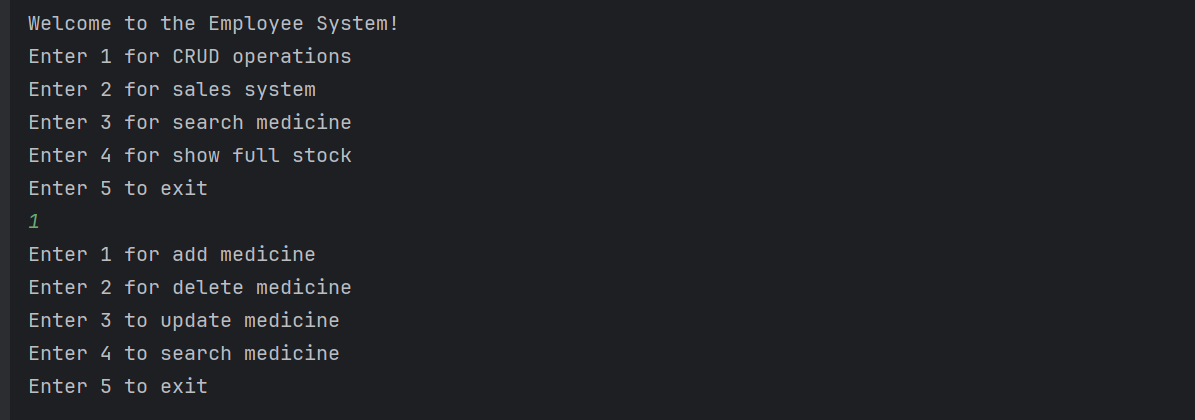
**SIGN IN**

****

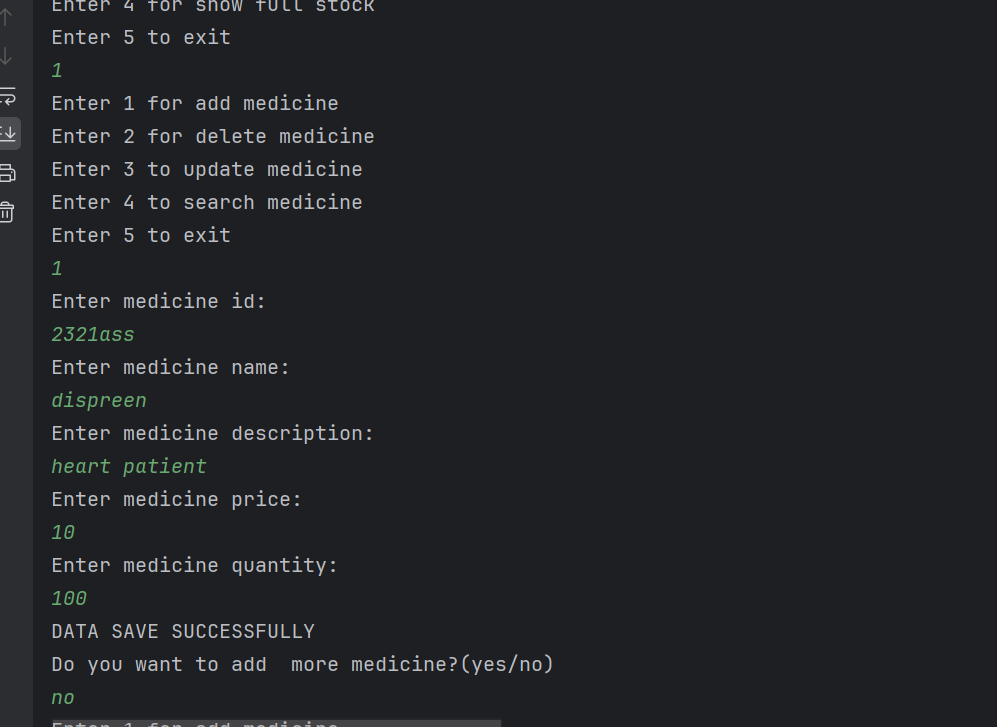
**SIGN UP:**

****

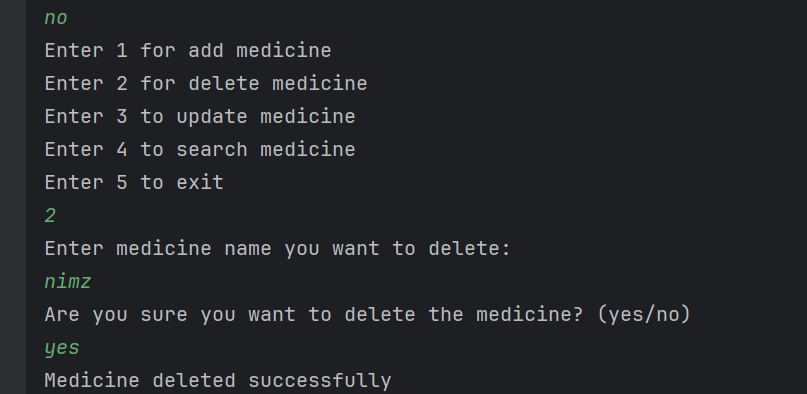
**MENU:**

****

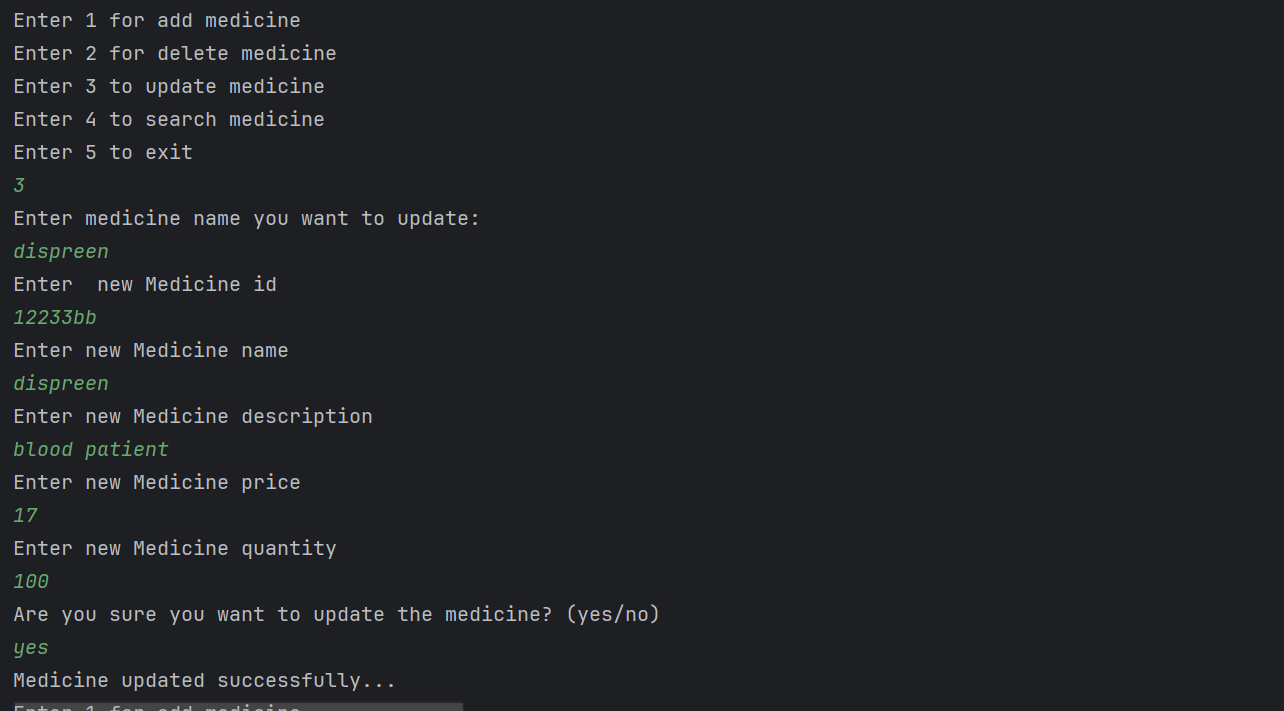
**ADD MEDICINE:**

****

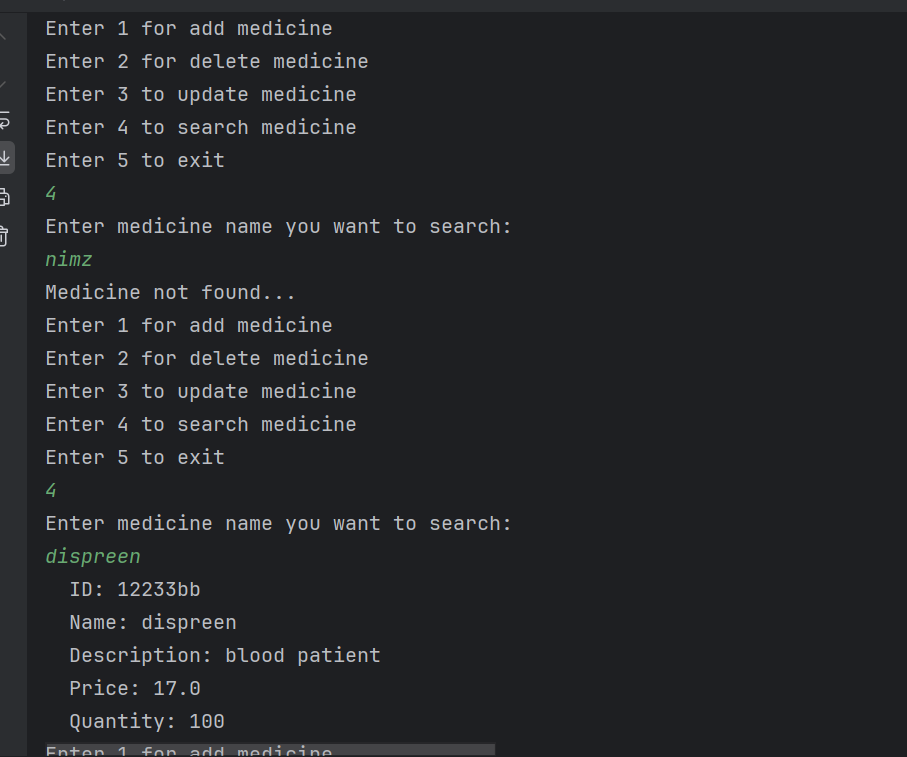
**DELETE MEDICINE:**

****

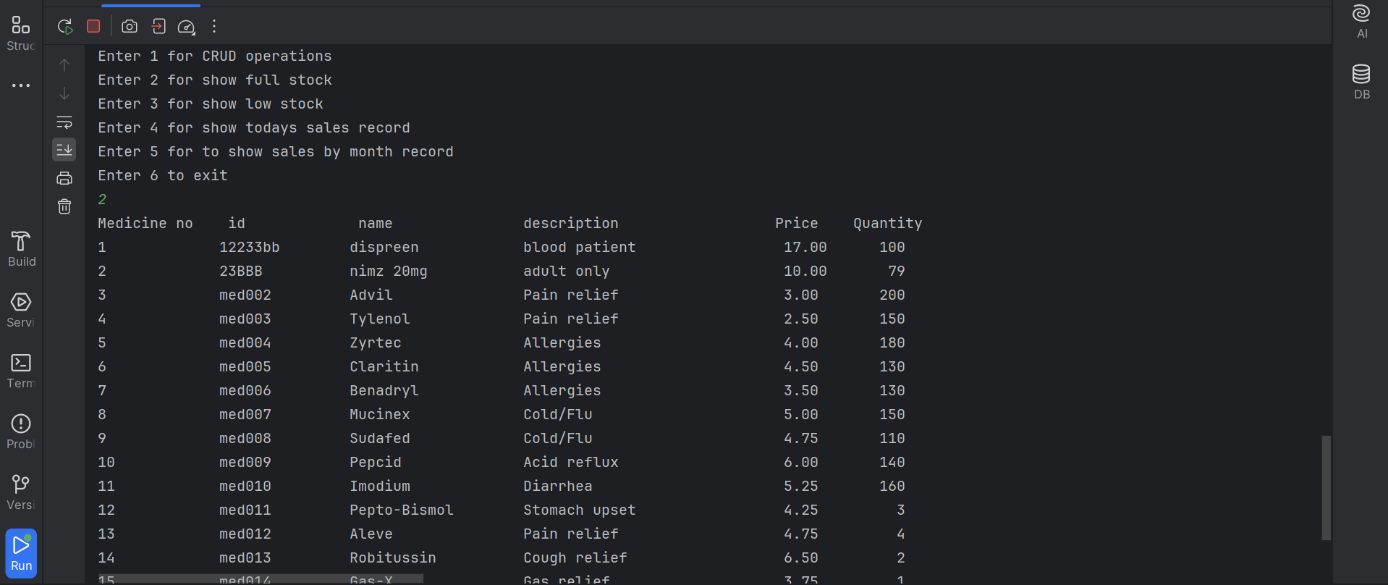
**UPDATE MEDICINE:**

****

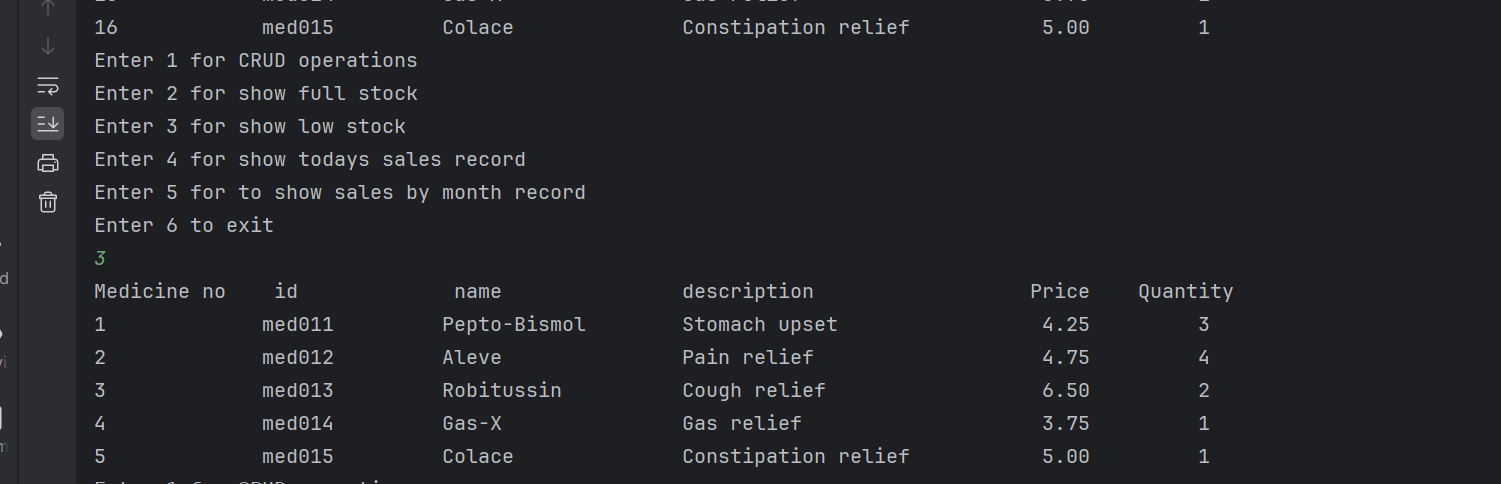
**Search Medicine:**

****

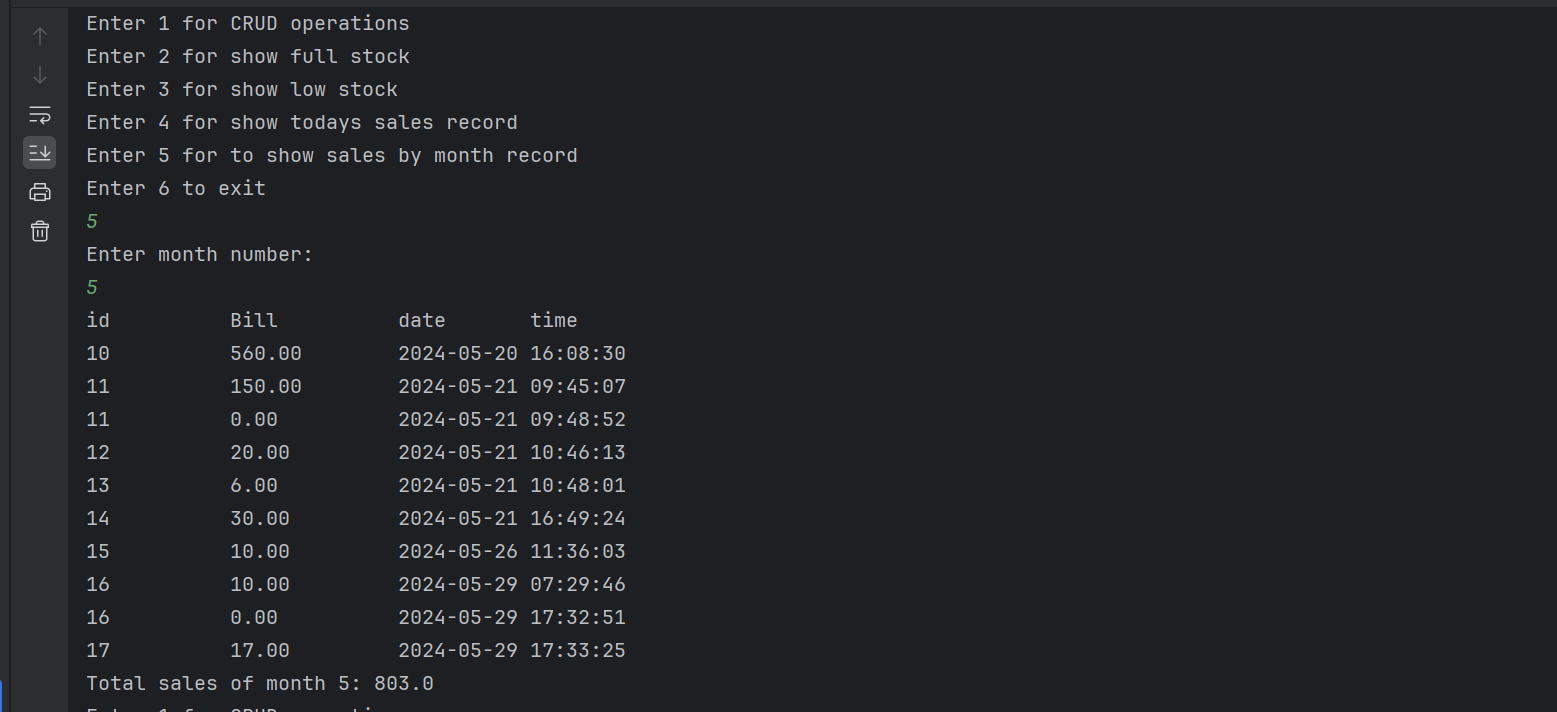
**FULL STOCK:**

****

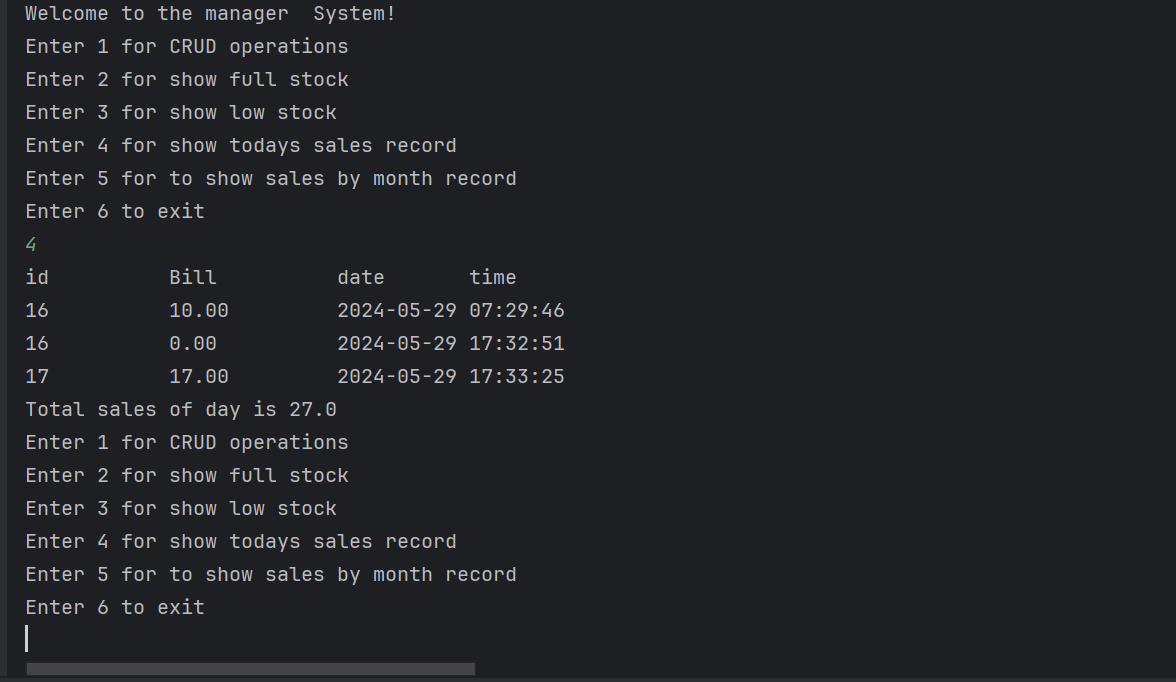
**LOW STOCK:**

****

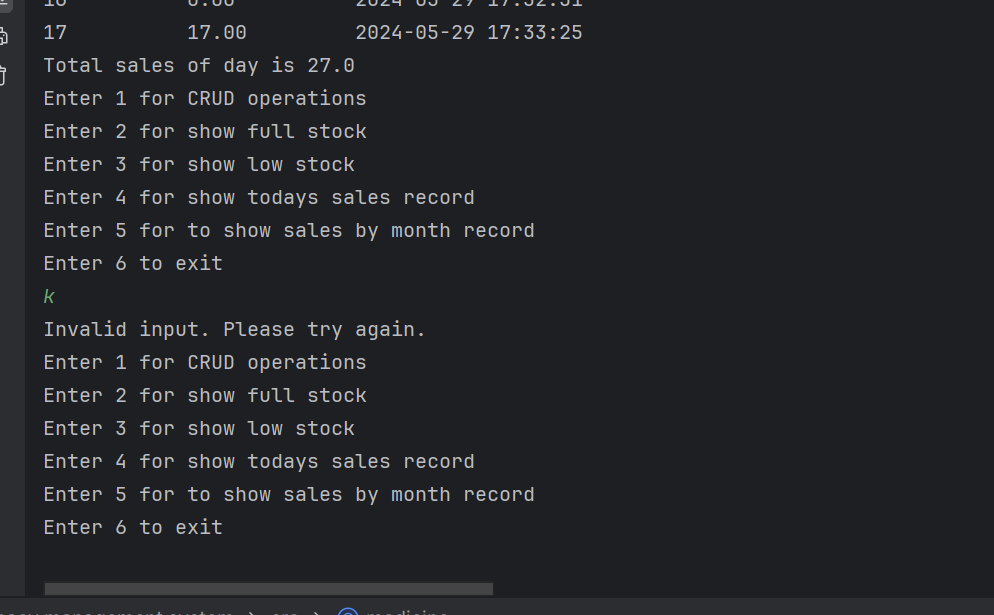
**SHOW MONTHLY REPORT:**

****

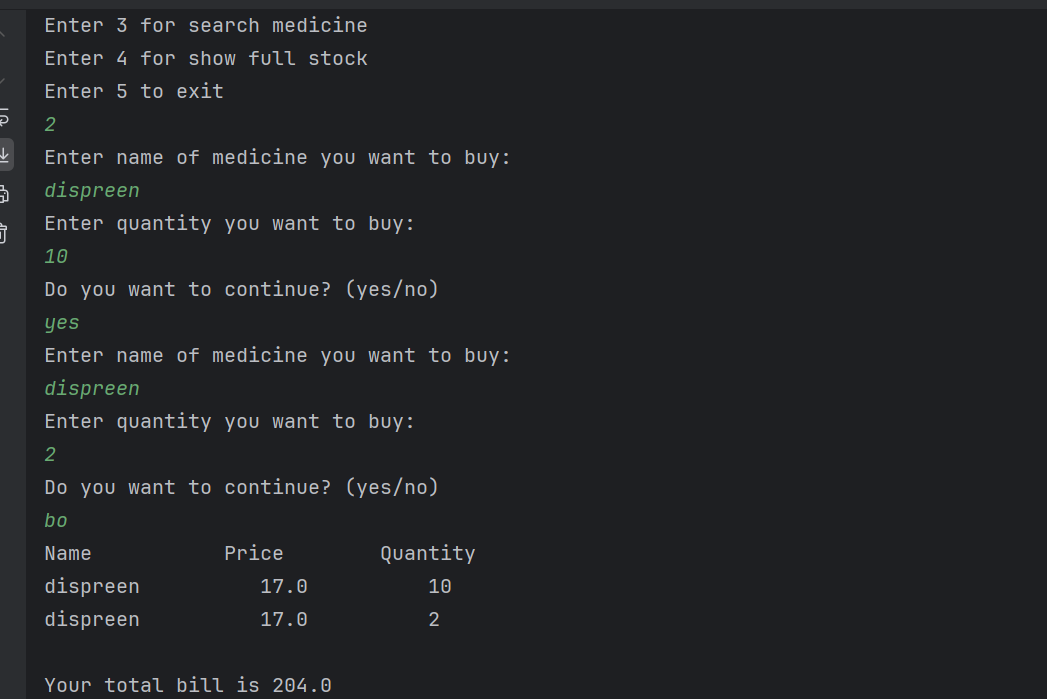
**DAILY SALES:**

****

**EXCEPTIONAL HANDLING:**

****

**SALES:**

****