ByteKart Inventory Management System

Project Overview

The **ByteKart Inventory Management System** is a Python-based desktop application designed for small businesses to manage inventory, streamline sales transactions, and generate insightful reports. This system provides role-based access for Admins and Cashiers to ensure secure and efficient operations.

Features

- **User Authentication:** Secure login for Admin and Cashier roles.
- **Inventory Management:** Add, update, delete, and view inventory items.
- **Checkout Process:** Generate bills, update inventory, and record sales.
- **Sales Reporting:** Generate detailed reports of transactions and top-selling items.

Folder and File Structure

```
inventory_management/
├── main.py                 # Main entry point of the application
├── database/
│   └── inventory_db.py       # Database handling for items, users, and sales records
├── ui/
│   ├── login.py              # Login window for Admin/Cashier authentication
│   ├── inventory_ui.py       # Main interface for inventory management
│   └── checkout_ui.py        # Checkout window for sales transactions and bill generation
├── models/
│   ├── item.py               # Item class definition
│   ├── user.py          # User class definition
│   └── sales_record.py     # SalesRecord class to store transaction details
└── reports/
    └── sales_report.py     # Logic for generating sales reports
```

File Descriptions

1. main.py

This is the entry point of the application. Running this script launches the login screen, allowing users to authenticate and access the system's features based on their roles.

2. database/

Contains the logic for database management.

- **inventory_db.py**: Handles database interactions such as CRUD operations for inventory, user authentication, and sales records.

3. ui/

Contains the graphical user interface (GUI) components of the application.

- **login.py**: The login window allows Admin and Cashier users to authenticate.
- **inventory_ui.py**: Provides an interface for managing inventory, including adding, updating, and viewing items.
- **checkout_ui.py**: Enables users to process sales transactions, generate bills, and update inventory accordingly.

4. models/

Contains object-oriented representations of core entities.

- **item.py**: Defines the Item class with attributes like name, price, and quantity.
- **user.py**: Defines the User class for storing user credentials and roles.
- **sales_record.py**: Defines the SalesRecord class for storing transaction details, such as purchased items, quantities, and total price.

5. reports/

Handles report generation functionality.

- **sales_report.py**: Logic for creating and displaying sales reports, including analytics like top-selling items and total revenue.

System Requirements

- **Operating System:** Windows 10 or later
- **Python Version:** Python 3.10+
- **Dependencies:** Install the required libraries using the following command:

bash

Copy code

```
pip install -r requirements.txt
```

How to Run the Application

1. **Clone the Repository:**

bash

Copy code

```
git clone <repository-url>
cd inventory_management
```

2. **Install Dependencies:**

bash

Copy code

```
pip install -r requirements.txt
```

3. **Run the Application:**

bash

Copy code

```
python main.py
```

Key Functionalities by Role

Admin:

- Full access to manage inventory (add, update, delete).
- Generate and view detailed sales reports.

Cashier:

- Access to checkout and billing functions.
- Restricted access to inventory management.

Future Enhancements

- Multi-user concurrency support.
- Cloud-based database integration.
- Advanced analytics and visualizations.

Contributors

- **Developer:** Aakib Kibria Khan, Aritra Nandi
- **Role:** Python Developer

For queries or feedback, please contact [akkhan9@myseneca.ca].

Thank you for using ByteKart!