

## Table of Contents

<b>Mini ERP / Unified Inventory, Customer, Transaction and Billing System.....</b>	<b>1</b>
1. Project Overview.....	1
2. Objectives.....	1
3. System Architecture.....	2
4. Technologies Used .....	2
5. Folder Structure .....	2
6. Authentication Module.....	3
7. Customers Module.....	3
8. Products Module.....	4
9. Transactions Module .....	5
10. Billing Module .....	5
11. Data Validation & Error Handling .....	6
12. How to Compile & Run.....	6
13. Conclusion.....	6

# Mini ERP / Unified Inventory, Customer, Transaction and Billing System

## 1. Project Overview

The **System** is a console-based application developed in **C++** that simulates core business operations such as **authentication, customer management, product inventory, transactions, and billing**.

The system uses **CSV files** for persistent storage and is designed with a **proper structure**, making it easy to understand, extend, and maintain. This project demonstrates real-world concepts like file handling, data validation, modular programming, and basic ERP workflows.

## 2. Objectives

- Implement a simple ERP-like system using C++
- Practice structured programming and modular design
- Use file handling for persistent data storage

- Apply real-world business logic such as billing, discounts, and stock management
- Build a project suitable for **semester submission** and **portfolio showcase**

### 3. System Architecture

The system is divided into independent modules:

- **Authentication Module** - User signup and login
- **Customers Module** - Customer CRUD operations
- **Products Module** - Inventory management
- **Transactions Module** - Sales processing and their records
- **Billing Module** - Bill generation and discounts

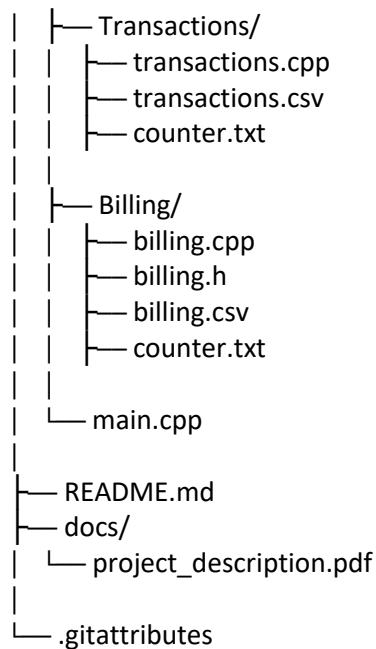
Each module communicates through shared CSV files.

### 4. Technologies Used

- **Language:** C++
- **Storage:** CSV files
- **Compiler:** g++ (GCC)

### 5. Folder Structure

```
Project-Name/
|
|-- src/
|   |-- Authentication/
|   |   |-- authentication.cpp
|   |   |-- data.csv
|   |   |-- counter.txt
|   |-- Customers/
|   |   |-- customers.cpp
|   |   |-- customers.h
|   |   |-- customers.csv
|   |   |-- counter.txt
|   |-- Products/
|   |   |-- products.cpp
|   |   |-- products.h
|   |   |-- products.csv
|   |   |-- counter.txt
```



## 6. Authentication Module

### Purpose

Handles user **sign-up** and **login** functionality.

### Features

- Unique User ID generation using counter.txt
- Duplicate username validation
- Verification during login

### Data Storage

#### Authentication/data.csv

UID, Username, Password

### Key Functions

- signUp() - Registers a new user
- signIn() - Authenticates existing user
- authentication() - Login menu controller

## 7. Customers Module

## **Purpose**

Manages customer records.

## **Features**

- Add, update, delete, search customers
- Customer type handling (General / Regular)

## **Data Storage**

### **Customers/customers.csv**

ID, Name, Phone, Type

## **Business Logic**

- Regular customers receive discounts during billing

## **Key Functions**

- addCustomers()
- updateCustomers()
- deleteCustomers()
- searchCustomers()
- allCustomers()

# **8. Products Module**

## **Purpose**

Manages product inventory.

## **Features**

- Add, update, delete, search products
- Quantity and price management

## **Data Storage**

### **Products/products.csv**

ID, Name, Description, Price, Quantity

## **Key Functions**

- addProducts()
- updateProducts()
- deleteProducts()
- searchProducts()
- allProducts()

## **9. Transactions Module**

### **Purpose**

Handles sales transactions involving customers and products.

### **Features**

- Multiple products per transaction
- Stock validation before sale
- Automatic stock deduction
- Stores current date and time of transactions

### **Data Storage**

#### **Transactions/transactions.csv**

Transaction\_ID, Customer\_ID, Product\_ID, Quantity, Total\_Price, DateTime

### **Key Logic**

- Validates customer existence
- Validates product availability
- Prevents over-selling

## **10. Billing Module**

### **Purpose**

Generates bills from completed transactions.

### **Features**

- Display Product
- Calculates subtotal from transactions
- Applies discounts

- Generates final bill receipt

### Discount Rules

- **Regular customer:** 5% discount
- **General customer:** No discount

### Data Storage

#### Billing/billing.csv

Bill\_ID, Transaction\_ID, Customer\_ID, Sub\_Total, Tax, Discount, Grand\_Total, DateTime

## 11. Data Validation & Error Handling

- Duplicate username prevention
- Invalid customer/product ID handling
- Stock availability checks
- File open failure handling

## 12. How to Compile & Run

- Open terminal
- Go to the directory (unified-inventory-customer-transaction-and-billing-system/src/)
- Then run:
  1. g++ main.cpp -o main.exe
  2. .\main.exe

## 13. Conclusion

This project demonstrates a complete **Mini ERP System** using C++ with real-world business logic. It reflects strong understanding of **file handling, programming, validation, and system design**, making it suitable for academic submission and portfolio presentation.

---

*The End*