Jaypee Institute of Information Technology, Sector - 62, Noida

B.Tech CSE I Semester

SDF Project Report

Ecommerce Management System

Submitted to

Dr. Kavita Pandey

Dr. Shardha Porwal

Submitted by

Harsh Sharma	B5	2401030232
Karvy Singh	B5	2401030234
Rudra Kumar Singh	B5	2401030237

Letter of Transmittal

Dr. Kavita Pandey

Department of Computer Science & Information Technology

Dr. Shardha Porwal

Department of Computer Science & Information Technology

Subject: Submission of Report on "Ecommerce Management System"

Dear Dr. Kavita Pandey & Dr. Shardha Porwal,

We are pleased to submit our report on our Project titled *"Ecommerce Management System"* as part of our coursework. This project is an *Ecommerce Management System* which helps sellers and customers with a platform to sell and buy commodities. It provides several features for customers and sellers as well. Such as graphs, inventory listings, cart system for customers, etc.

We have endeavored to cover the topics in SDF-1 comprehensively and hope that this project meets your expectations.

Thank you for your guidance and the opportunity to work on this project.

Sincerely,

Harsh Sharma (2401030232)
Karvy Singh (2401030234)
Rudra Kumar Singh (2401030237)

Date: November 25, 2024

Contents

1 Abstract

The E-Commerce Management System is a C-based application designed to streamline the operations of both sellers and customers. The system provides separate functionalities for each user type, ensuring a seamless interaction between the two.

- For sellers, the platform allows them to log in securely using an ID and password. Once authenticated, sellers can manage their inventory by adding, updating, or deleting products. They also have access to view detailed sales data, helping them track revenue and manage orders from customers. Sellers can process orders by accepting, declining, or marking them for shipment.
- On the customer side, the system enables users to register or log in securely. Customers can browse the available products, view detailed descriptions, and add items to a shopping cart. Once they are ready to make a purchase, they can proceed to checkout. Additionally, customers can view their order history and track the status of ongoing orders.

2 Topics of SDF-1 Used

The E-commerce Management System utilizes several foundational concepts of C programming taught in the SDF-1 course. Below is a brief overview of the key topics used:

- **Data Types**: The program employs a variety of C data types such as `int`, `float`, `char`, and `double` to represent various elements like product prices, quantities, and user inputs.
- **Variables**: Variables are used extensively to store user credentials, product details, and other intermediate data during program execution.
- **Functions**: The modular structure of the code is achieved by using functions for specific tasks.
- **Pointers**: Pointers are used for efficient memory management, passing data between functions, and dynamic memory allocation. For instance, pointers are utilized to handle strings dynamically and to work with file streams.
- **Structures**: Custom structures like `struct Product` and `struct User` are defined to logically group related data. This makes it easier to handle complex entities like a product with its attributes or a user with their credentials and purchase history.
- **Strings**: Strings are used to manage textual data such as usernames, passwords, product names, and error messages. Functions from the `<string.h>` library, such as `strcmp()` and `strcpy()`, are employed for string manipulation.

These topics collectively form the foundation for the program, ensuring that the system is efficient, organized, and easy to maintain.

3 Design

An Ecommerce Portal is a very graphics heavy site UX wise. A lot of data is involved, so the users need better methods than text to properly understand them. For this reason, we are using a library called `ncurses` rather than building a traditional menu based UI.

`ncurses` allows complete control over the console window. Every block is available to be modified, and user inputs are available without the need to press Enter. So we can build very advanced and fun to use UIs in terminal as well.

The program is divided into mainly three sections:

- Login Page
- Customer Portal
- Seller Portal

3.1 Login Page

The Login Page serves as the entry point of the program. This page can redirect you to either of the portals depending on the input.

In this page, the user inputs a username and a password. If the credentials are correct then the user will be redirected to their account's corresponding portal.

3.2 Customer Portal

The Customer Portal uses a Sidebar and Main Content design. The sidebar controls which page to show on main content and main content follows. It has the following Menus:

- Browse Products
- My Cart
- Manage Account
- Logout

3.3 Seller Portal

The Seller Portal (similar to Customer Portal) uses a Sidebar and Main Content design. The sidebar provides the following menus:

- Orders
- Inventory
- Sales

4 Implementation Detail

4.1 Login Page

The Login Page is implemented in the `main.c` file. This module provides user authentication functionality and acts as the entry point to the system. It includes the following features:

- A menu-based user interface built with `ncurses`.
- Combined login for customers and sellers.
- User validation against stored credentials (e.g., stored in a csv file).
- Error handling for invalid inputs or credentials.

Placing orders and viewing order history **Code Highlights**:

- `int main()`: Entry point of the program, initialises `ncurses` screen and setups up the login screen.
- `int parse_csv(const char *filename, char *user_input1, char *user_input2)`: Checks the credentials against the data stored in the Users csv file.
  returns 0 if credentials are invalid, 1 if its a seller account and 2 if its a customer account.

This module serves as a gateway to the respective Customer and Seller portals, ensuring secure and appropriate access control.

4.2 Customer Portal

The Customer Portal is implemented in the `customer_portal` folder, containing multiple C files and supporting header files. It provides functionality for customer-related operations such as:

- Browsing products and categories.
- Adding products to the cart.
- Viewing and managing the shopping cart.
- Managing user account.

Code Highlights:

4.2.1 customer_portal/main.c

Contains initalization for customer portal.

- `int start_customer_portal()`: Starts the initalization for customer portal. Sets up the top window, sidebar and content window. Checks for user input, calls functions for different screens and menus.

4.2.2 customer_portal/cart.c

Contains functions for displaying the cart, adding items to cart and managing them.

- `struct CartItem`: A linked list which each node representing an item in the cart.
- `Product *copy_product(Product *)`: Creates a malloced copy of a Product struct.
- `void add_to_cart(Product *)`: adds a product to cart.
- `void display_cart(WINDOW *)`: displays the cart in an `ncurses` window.

4.2.3 customer_portal/display.c

Contains functions for displaying the UI and utility.

- `char **wrap_text()`: wraps text bigger than terminal size to next line.
- `void display_products()`: Display products listing menu

4.2.4 customer_portal/product.c

Contains functions for reading and writing products.

- `Product **load_products2()`: Load products from csv.
- `void write_products(Product **)`: Write products to csv.

The user interface leverages `ncurses` to provide a dynamic and user-friendly experience, allowing seamless navigation through menus and product listings.

4.3 Seller Portal

The Seller Portal is implemented in the `seller_portal` folder, which contains dedicated C files and header files to support seller-related functionalities. The main features of this module are:

- Adding new products to the platform.
- Managing existing product listings (update/delete operations).
- Viewing sales reports and order details.
- Editing seller profiles and managing account settings.

Code Highlights:

4.3.1 seller_portal/roughtui.c

Contains initalization for seller portal.

- `int run_tui()`: Starts the initalization for seller portal. Sets up the top window, sidebar and content window. Checks for user input, calls functions for different screens and menus.

4.3.2 seller_portal.wow.c

Contains functions for displaying static menus with beautiful UI.

- `void content_static(WINDOW *)`: displays the static product listings in an `ncurses` window.
- `int handle_menu_input()`: internal function which handles the input inside the static window.

4.3.3 seller_portal/product_new.c

Contains functions for CRUD of products and displaying them for the inventory screen.

- `Product **load_products2()`: Load products from csv.
- `void write_products(Product **)`: Write products to csv.
- `int display_products()`: Displays inventory screen and handles its input.

This module ensures that sellers have comprehensive control over their product inventory and insights into their business performance.

5 References

- NCURSES Docs: https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/
- The C Programming Language by Dennis Ritchie & Brian Kernighan
- This Project is also hosted on GitHub: https://github.com/Karvy-Singh/SdfSuper