

Online Shopping Management System

Tamilmaran R, Dr. Prasanna. S

*Professor, Department of Computer Application-PG, VISTAS,

** Student, Department of Computer Application-PG, VISTAS,
Chennai

ABSTRACT

Online commerce has changed how people shop because it has provided a medium where physical presence is not necessary to make a transaction. This paper covers the design and development of the Online Shopping Management System (OSMS) through programming languages like Python (Flask), MySQL, HTML5, CSS3, and JavaScript. There are three kinds of users in the OSMS system: the Administrator, the Employee, and the Customer. The Administrator manages the accounts and all orders. The Employee works on the product listing. The Customer creates an account, searches for products, adds products to the cart, pays online and checks the order status.

Keywords — e-commerce, online shopping, Flask, MySQL, role-based access, web application, order management

I. INTRODUCTION

Online shopping is popular due to the existence of the internet and its ease for customers. Unlike conventional stores, one needs not to travel to use them; instead, the internet acts as a medium where anyone can shop from wherever they are without moving or queuing.

Project Introduction The current project describes a website application that is specifically designed for the Online Shopping Management System (OSMS). There are three kinds of users that can use the current system; they include the Administrator that controls the system, the Employees that control the products, and the Customers that are the buyers.

The following technologies have been used in building the system: Flask (Python), MySQL, HTML5, CSS3, JS.

The technology stack employed to develop the software application includes Flask (Python), MySQL, and HTML5, CSS3, JS. The next few sections provide detailed information on the manual system, the digital system, methodology, technology stack, system design, implementation, and results.

II. EXISTING SYSTEM

Prior to the introduction of the system, the store would depend on manual methods that included the use of paper records. This involved the use of physical registers for product information, paper slips for orders, and manual billing. Customer information would be stored in documents.

A. Disadvantages

Manual system faced various challenges.

Preparation of report consumed a lot of time. It was hard to respond to clients' inquiries. Cost of paper and stationary was high. Human error was common when keeping the record. Simultaneous access to the records was impossible.

III. PROPOSED SYSTEM

The new OSMS system will automate the entire process of purchasing goods using an online software tool. The application will store all the information in a centralized MySQL database where all users can access the data at once. The system will automate the complete shopping process.

Each module is responsible for specific functions. The Admin module oversees employee management, order tracking, and delivery agent allocation. The Employee module enables the employees to make changes to product entries. The Customer module permits the users to register, login, browse for products, add them to their shopping cart, purchase, and track orders.

B. Advantages

There is less need for paper work. Manual computation of amounts during check out is avoided through automatic computation of amounts hence there will be no mistakes due to manual computations. Reports can be printed immediately. All transactions have booking IDs that help in auditing purposes.

IV. METHODOLOGY

Development started with interviews to determine the requirements and roles. Data Flow Diagrams (DFDs) have been developed in three levels. The Level-0 DFD views the whole system as one process communicating with the Admin, Employee, and User roles and the database. Level-1 splits the Admin process. Level-2 describes the

Customer process from registration to payments.

Normalization was used in designing the database schema, producing five tables, namely regtb (users), protb (products), booktb (orders), addcart (cart items), and reviewtb (reviews). Mapping of routes in Flask was done for all use cases. HTML templates were created using Jinja2. The project underwent unit testing of all modules and end-to-end testing of the entire process.



Fig 1. Data Flow Diagram (DFD)

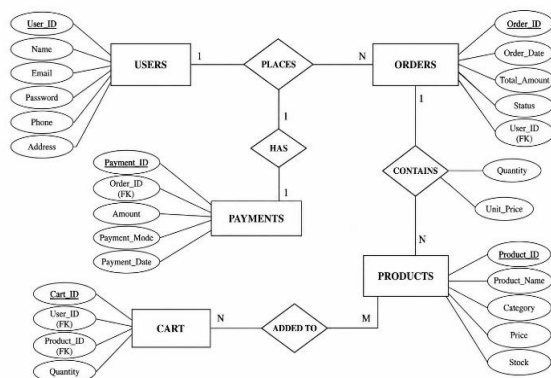


Fig 2. Entity Relationship Diagram (ERD) (Online Shopping Management System)

V. TECHNOLOGIES USED

A. HTML5

Structure is provided to all webpages by HTML5. The use of semantic tags such as header, nav, section, and footer increases clarity. User inputs are obtained through the use of forms for various activities like login, signup, etc.

B. CSS3

The CSS3 is responsible for defining the appearance of the application. Responsive layout is achieved using Flexbox. All styles for all elements are written in an external style sheet file

C. JavaScript

JavaScript is responsible for handling interactions on the client side. It does not require reloading the page for updating the cart total dynamically whenever quantities are changed. Form validation is another task performed by JavaScript before the form is submitted.

D. Python (Flask)

Python 3.7 with Flask micro-web framework

executes all server-side logic. Flask mapping routes URL requests to Python functions. Session management is done for users across request handling. Mysql-connector-python executes SQL queries, and the datetime module adds timestamps to all transactions.

E. MySQL

Application data is stored in well-structured tables in MySQL. All CRUD operations are performed using SQL queries. Queries for counting will produce booking IDs sequentially. The database was hosted on WAMP Server.

VI. SYSTEM DESIGN

A. Data-Flow Diagrams

Level-0 DFD indicates that the system receives input from Admin, Employee, and User objects and communicates with the database. In Level-1 DFD, we have explained the Admin subsystem, which includes Login, Employee Management, and Report Generation. Level-2 DFD explains the process of the Customer, which consists of Register, Login, View Products, Add to Cart, Payment, and Booking Confirmation.

B. System Specifications

Hardware Requirements: Processor: Intel / AMD x64 processor 2 GHz or more; RAM: 4 GB or more; Disk Space: 20 GB or more; Display Resolution: 1024 x 768 pixels or more.

Software Requirements: Operating System: Windows 10/11; Python Version: Python 3.7;

VII. IMPLEMENTATION

The Flask application gets initialized by establishing a connection with the MySQL database named ShopMe. Route functions handle all features. Routes like /Register and /Login authenticate users using the regtb database table and establish sessions. The /AddProduct route function is restricted to employees and inserts products to the protb database table.

The /addcart route enables the addition of the selected product to the addcart table associated with the authenticated user. The /Book route is for completing the payment process. It fetches product information along with user information and computes the sum based on quantity and unit price. The order details will be saved in the booktb table with a unique booking id. Admin routes update the order status from pending to packing, shipping, or delivered.

VIII. RESULTS

The system has been successfully implemented on the local WAMP Server for testing purposes on all three types of users. The customer registration and login functions were working properly. The browse products feature filtered the products

according to their categories and sub-categories. Cart functions gave correct cart totals.

The checkout form collected payment information, created an entry in the database, and showed order confirmation information. Changes of admin status from “booked” to “packing,” “shipment,” and “delivered” could be seen by the user through the tracking page. Submitting a review saved the rating and comment in the review table and also showed in the admin panel. No bugs or errors occurred during testing.

SCREENSHOT:

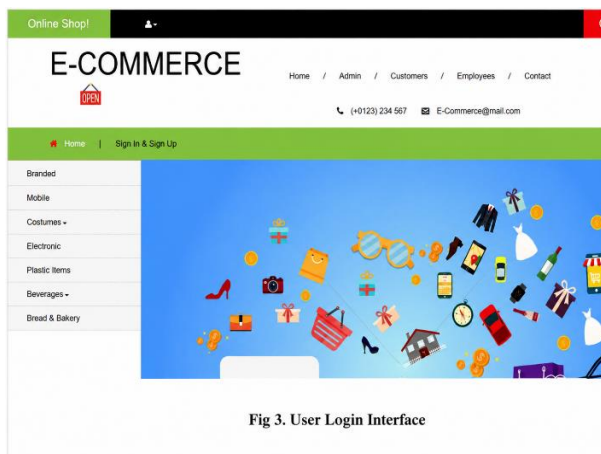


Fig 3. User Login Interface

**TABLE I
PRODUCT DATASET**

Product details that were used in the Online Shopping Management System are illustrated in the table below. This table provides information on product id, product name, category, cost, and stock level of the product.

Product_ID	Product_Name	Category	Price (₹)	Stock
P001	Almond	Dry Fruits	750	25
P002	Cashew Nut	Dry Fruits	650	18
P003	Pistachio	Dry Fruits	900	12
P004	Raisins	Dry Fruits	300	40
P005	Dates	Dry Fruits	200	35

**TABLE II
ORDER DETAILS**

Table shows example data for order tracking kept in the Online Shopping Management

System. The system tracks the details of customer purchases, items purchased, shipping status, and order status data.

Customer_ID	Customer_Name	City	Order_Count	Status
C001	Raja R	Chennai	3	Active
C002	Divya L	Salem	2	Active
C003	Karthik M	Trichy	1	Inactive

**TABLE III
EMPLOYEE DETAILS**

A table showing examples of employee records to be stored in the Online Shopping Management System is shown below. This table consists of information on employee ID, employee designation, department, and contact information.

Employee_ID	Employee_Name	Designation	Contact_No
E001	Arun Kumar	Manager	9876543210
E002	Priya S	Sales Executive	9123456780
E003	Karthik R	Store Assistant	9345678123
E004	Divya M	Inventor y Staff	9789012345

GRAPHICAL ANALYSIS OF MONTHLY SALES AND ORDERS

This chart portrays monthly sales and products ordered data for the Online Shopping Management System. It gives information about the rise in the number of customer orders and sales for various months.

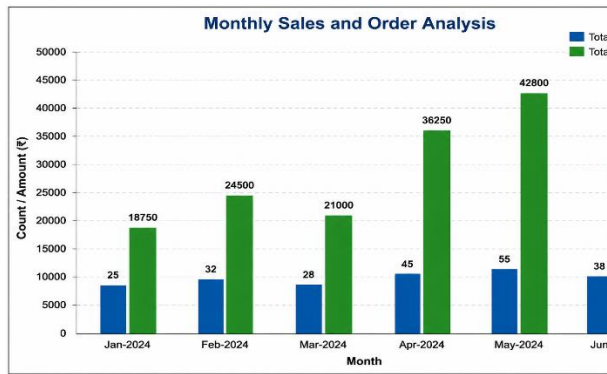


Fig 5. Monthly Sales and Order Analysis Graph

IX. CONCLUSION

The research project introduced an Online Shopping Management System using Python Flask, MySQL, HTML5, CSS3, and JavaScript. This Online Shopping Management System digitizes all the retail processes from managing products to customers purchasing the products, paying online, and order tracking. This has managed to replace the old manual process which was less efficient.

For future work, recommendations would be provided to users, external payment gateway options like Razorpay and PayPal would be integrated, progressive web app support for mobile platforms

would be incorporated, and the application would be deployed on the cloud platform to deal with increased user loads.

REFERENCES

- [1] K. Laudon & C. Traver, E-Commerce: Business, Technology, Society, 16th ed., Hoboken, NJ: Pearson, 2020.
- [2] M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed., Sebastopol, CA: O'Reilly Media, 2018.
- [3] P. DuBois, MySQL Cookbook, 4th ed., Sebastopol, CA: O'Reilly Media, 2020.
- [4] B. Ware, Open-Source Development
- [5] K. Yank, Build Your Own Database Driven Website Using PHP and MySQL. Melbourne: SitePoint, 2004.
- [6] W3Schools. (2024) HTML5 Tutorial. [Online]. Available: <http://www.w3schools.com>
- [7] Python Software Foundation. (2024) Python 3 Documentation. [Online]. Available: <http://www.docs.python>
- [8] MySQL. (2024) MySQL 8.0 Reference Manual. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>