SALES SAVVY – MODULAR AND SCALABLE E-COMMERCE PLATFORM

Sasmita Sahoo
Student, Department of Computer Science and Engineering
GIFT Autonomous, Bhubaneswar, Odisha, India, sasmita2022@gift.edu.in

Sangita Mahadani
Student, Department of Computer Science and Engineering
GIFT Autonomous, Bhubaneswar, Odisha, India, sangitam2022@gift.edu.in

Prof. Meenakshi Maharana
Assistant Professor, Department of Computer Science and Engineering
GIFT Autonomous, Bhubaneswar, Odisha, India, meenakshi@gift.edu.in

ABSTRACT

The increasing reliance on digital platforms has transformed the way businesses operate, particularly in the retail sector. E-commerce systems have become essential for enabling seamless transactions, efficient inventory management, and enhanced customer engagement. This project, titled **"Sales Savvy – Modular and Scalable E-commerce Platform,"** focuses on designing and developing a full-stack web application that provides a complete online shopping solution for both administrators and customers.

The system is built using a modern technology stack, including ReactJS for frontend development, Spring

Boot for backend services, and MySQL for database management. It incorporates secure authentication using JSON Web Tokens (JWT), ensuring role-based access control for Admin and Customer users. The platform supports essential functionalities such as product management, cart operations, order processing, and secure payment integration using third-party gateways like Razorpay or Stripe.

The architecture of the system is modular, dividing functionalities into distinct services such as Authentication, User Management, Product Management, Cart Management, Order Management, and Payment Service. This approach enhances scalability, maintainability, and flexibility. Additionally, the system is designed to handle multiple users simultaneously with optimized performance and security.

Overall, the Sales Savvy platform demonstrates a real-world implementation of full-stack development concepts and provides a robust solution for small and medium-scale businesses transitioning into digital commerce.

Keywords: E-Commerce, ReactJS, Spring Boot, MySQL, JWT, REST API, Postman, Online Shopping

1. INTRODUCTION

E-commerce enables businesses to sell products globally through web applications. Modern platforms must provide secure authentication, efficient product management, real-time cart updates, and seamless payment processing. Sales Savvy was designed to meet these requirements using a scalable full-stack architecture.Traditional retail systems rely heavily on physical stores, manual record-keeping, and face-to-face transactions. While these systems have served businesses for decades, they are no longer sufficient to meet the growing demands of modern consumers. Customers today expect features such as real-time product availability, secure payment options, order tracking, and personalized shopping experiences.

To address these evolving needs, businesses are adopting digital platforms that automate operations and enhance customer satisfaction. The **Sales Savvy E-commerce Platform** is designed as a comprehensive solution that integrates all essential features of an online marketplace into a single system. It enables businesses to manage products, users, and transactions efficiently while providing customers with a seamless shopping experience.

2. OBJECTIVES OF THE PROJECT

The project aims to develop a secure and user-friendly e-commerce platform. Major objectives include implementing JWT-based authentication, managing products and inventory, supporting cart and checkout operations, processing payments securely.

- To implement a secure authentication system using JWT for role-based access control
- To develop separate interfaces for Admin and Customer users

- To enable efficient product management, including adding, updating, and categorizing products
- To implement cart and order management functionalities
- To integrate secure payment gateways for online transactions
- To ensure scalability and maintainability through modular architecture
- To provide a responsive and user-friendly interface using modern frontend technologies

3. LITERATURE SURVEY

The application follows a three-tier architecture. The presentation layer is developed with ReactJS. The application layer uses Spring Boot and Spring Security to implement business logic and REST APIs. The data layer uses MySQL to store users, products, carts, orders, and payment records.

An e-commerce system typically consists of multiple components, including a user interface, backend services, database management, and payment integration. The frontend is responsible for providing a user- friendly interface, while the backend handles business logic, authentication, and communication with the database. Modern systems use RESTful APIs to ensure seamless interaction between these components. Security is a critical aspect of e-commerce platforms. Technologies such as HTTPS, encryption, and authentication mechanisms like JSON Web Tokens (JWT) are widely used to protect sensitive user data. Additionally, scalability is achieved through modular and micro-services-based architectures, allowing systems to handle large volumes of users and transactions.

4. EXISTING SYSTEM

The platform consists of Authentication, User Management, Product Management, Cart Management, Order Management, Payment, Database, and Admin modules. Each module is independently developed and communicates through standardized REST interfaces.

Amazon is one of the largest e-commerce platforms globally, known for its advanced recommendation system, efficient logistics, and extensive product range. It uses sophisticated algorithms and cloud-based infrastructure to handle millions of transactions daily. Similarly, Flipkart provides a robust platform tailored to the Indian market, offering features like easy returns, multiple payment options, and localized services.

Shopify, on the other hand, is a popular platform for small and medium businesses, enabling them to create their own online stores. It provides customizable templates, payment integration, and inventory management tools.

5. PROPOSED SYSTEM

rendering through the virtual DOM. Spring Boot simplifies backend development using controllers, services, repositories, and entities. Hibernate and JPA manage database relationships, while JWT secures protected endpoints.

The system is divided into multiple modules, such as Authentication Service, User Management, Product Management, Cart Management, Order Management, and Payment Service. This modular approach ensures that each component can be developed, tested, and maintained independently, improving overall system efficiency.

Key Features of the proposed System

- Security is enhanced through the implementation of JWT-based authentication, which ensures that only authorized users can access specific functionalities.
- Role-based access control is implemented to differentiate between Admin and Customer users, providing appropriate permissions to each.
- The system also integrates payment gateways such as Razorpay or Stripe, enabling secure and reliable online transactions.
- it is designed to handle multiple users simultaneously, ensuring high performance and responsiveness.
-

6. SYSTEM REQUIREMENTS

REST APIs use GET, POST, PUT, and DELETE methods. Representative endpoints include /api/auth/login, /api/products, /api/cart, and /api/orders. Postman is used to send requests, verify JSON responses, test authorization headers, and document API collections. Testing includes unit, integration, system, and user acceptance testing.

Hardware Requirements

The system is designed to run on standard computing devices without requiring high-end hardware. The minimum hardware requirements include:

- **Processor:** Intel Core i3 or higher

- **RAM:** Minimum 4GB (8GB recommended for better performance)
- **Storage:** At least 256GB hard disk or SSD
- **Network:** Stable internet connection for accessing web services.

The software requirements include development tools, frameworks, and runtime environments:

- **Operating System:** Windows, Linux, or macOS
- **Frontend Technologies:** ReactJS, HTML, CSS, Bootstrap
- **Backend Technologies:** Spring Boot, Spring Security
- **Database:** MySQL
- **Authentication:** JSON Web Token (JWT)
- **API Communication:** REST APIs
- **Development Tools:** VS Code / IntelliJ IDEA
- **Version Control:** Git
- **Containerization (Optional):** Docker

These tools and technologies are selected to ensure scalability, security, and maintainability of the system.

7. SYSTEM ARCHITECTURE

The Sales Savvy platform follows a **three-tier architecture**, which separates the system into three distinct layers:

1. Presentation Layer (Frontend):

The Presentation Layer is responsible for providing the graphical user interface through which users interact with the system. It is developed using ReactJS, a modern JavaScript library for building dynamic and responsive web applications. ReactJS uses a component-based architecture and Virtual DOM, which improves rendering speed and enhances the overall user experience.

This layer allows customers to register, log in, browse products, search for items, view product details, add products to the shopping cart, and place orders. It also provides an administrative dashboard where administrators can manage products, users, and orders. The frontend communicates with the backend using REST APIs and exchanges data in JSON format. Additional technologies such as HTML, CSS, Bootstrap, and Axios are used for styling and API integration.

2. Application Layer (Backend):

The Application Layer contains the core business logic of the system and is implemented using Spring Boot. Spring Boot simplifies enterprise application development by providing built-in support for REST API creation, dependency injection, database integration, and security configuration.

This layer handles essential functionalities such as user authentication, authorization, product management, cart processing, order management, and payment handling. Spring Security and JSON Web Tokens (JWT) are used to secure the application and provide role-based access control for Admin and Customer users. The backend exposes RESTful APIs that process client requests, validate data, apply business rules, and return structured responses to the frontend.

3. Data Layer (Database):

The MySQL database stores all application data, including user information, product details, orders, and payments. It ensures data consistency and integrity.

This layered architecture improves maintainability and scalability. Each layer operates independently, making it easier to update or modify specific components without affecting the entire system.

Benefits of the Architecture

The three-tier architecture offers several advantages:

- **Modularity:** Each layer is developed independently.
- **Scalability:** New features can be added without affecting existing components.
- **Maintainability:** Code is easier to debug, test, and update.
- **Security:** Sensitive operations are protected using JWT and Spring Security.
- **Performance:** Optimized communication between frontend, backend, and database.
- **Re-usability:** REST APIs can be consumed by web or mobile applications.

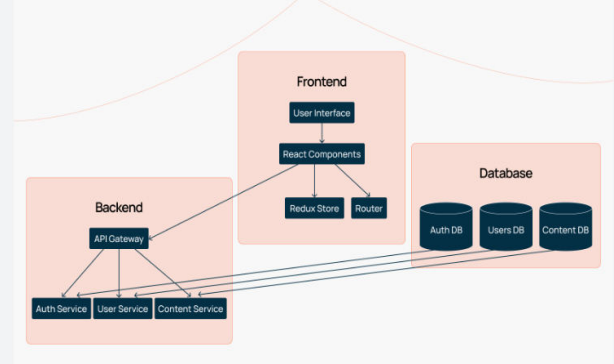

Fig1: System Architecture Diagram

8. DATA FLOW DIAGRAM

A **Data Flow Diagram (DFD)** is a graphical representation that shows how data moves through the e-commerce system. It illustrates the interaction between users, system processes, and databases, helping to understand how information is processed and stored.

In an e-commerce website, the main external entities are **Customer**, **Admin**, and **Payment Gateway**. Customers interact with the system to register, log in, browse products, add items to the cart, place orders, and make payments. Administrators manage products, users, and orders. The payment gateway processes online transactions securely.

The major processes in the DFD include:

1. **User Authentication** – Handles user registration and login by validating credentials with the user database.
2. **Product Management** – Retrieves product details from the product database and allows administrators to add, update, or delete products.
3. **Cart Management** – Stores selected products temporarily in the cart database.
4. **Order Processing** – Creates and updates order records in the order database.
5. **Payment Processing** – Sends payment information to the payment gateway and stores transaction details in the payment database.

The system uses several data stores such as **User Database**, **Product Database**, **Cart Database**, **Order Database**, and **Payment Database** to maintain all necessary information.

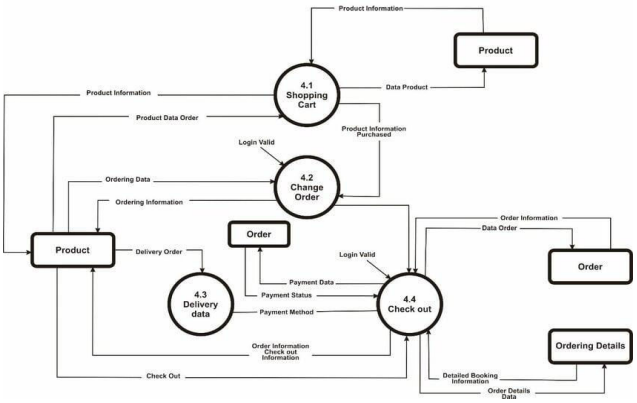

Fig2: Data Flow Diagram

9. DATABASE DESIGN

The database design is a crucial part of an e-commerce website because it defines how data is organized, stored, and related. In the Sales Savvy platform, the backend is developed using Spring Boot and the frontend using React JS, while MySQL is used as the relational database management system. The database is designed to manage users, products, shopping carts, orders, and payments efficiently.

Main Entities in the Database

The database consists of several interconnected tables:

1. **User Table** – Stores customer and administrator information such as user ID, name, email, password, phone number, address, and role (ADMIN or CUSTOMER).
2. **Product Table** – Contains product details including product ID, name, description, category, price, stock quantity, and image URL.
3. **Category Table** – Organizes products into categories such as Electronics, Clothing, and Books.
4. **Cart Table** – Represents the shopping cart associated with each user.
5. **Cart Item Table** – Stores products added to the cart along with quantity and subtotal.
6. **Order Table** – Maintains order information such as order ID, user ID, order date, total amount, shipping address, and order status.
7. **Order Item Table** – Stores each product included in an order.
8. **Payment Table** – Records payment transactions, including payment method, transaction ID, amount, and payment status.
9. **Address Table** – Stores multiple shipping and billing addresses for users.

Relationships Between Tables

- One **User** can have many **Orders**.
- One **User** has one **Cart**.
- One **Cart** contains many **Cart Items**.
- One **Product** can appear in many **Cart Items** and **Order Items**.
- One **Order** contains many **Order Items**.
- One **Order** has one **Payment**.
- One **Category** contains many **Products**.
- One **User** can have multiple **Addresses**.

Technologies Used for Database Integration

Spring Boot uses **Spring Data JPA** and **Hibernate** to map Java entity classes to MySQL tables. Annotations such as @Entity, @Table, @OneToMany, @ManyToOne, and @OneToOne define relationships between entities. Repositories extend JpaRepository to provide automatic CRUD operations without writing SQL queries manually.

Benefits of the Database Design

This database design ensures data consistency, minimizes redundancy, and supports efficient retrieval of information. It allows secure user management, smooth cart and order processing, and reliable payment tracking. The relational structure also makes the system scalable and easy to maintain as new features are added.

10. MODULE DESCRIPTION

The Sales Savvy E-commerce platform is designed using a modular approach, where the system is divided into multiple independent modules. Each module performs a specific function and interacts with other modules through well-defined interfaces. This modular design enhances scalability, maintainability, and flexibility of the system.

Authentication Module :

This module uses **JWT (JSON Web Token)** for secure authentication. When a user logs in, a token is generated and sent to the client. This token is used to verify user identity in subsequent requests. It also supports role-based access control, differentiating between Admin and Customer users.

➢ **User Management Module :**

The User Management module handles all user related operations.

- Register an account
- Login securely
- Update profile information
- Change passwords

➢ **Product Management Module :**

The Product Management Module allows administrators to manage products effectively. It includes functionalities such as:

- Adding new products
- Updating product details
- Deleting products
- Categorizing products

➢ **Cart Management Module**

The Cart Management Module allows users to manage their shopping cart. It provides features such as:

- Adding products to cart
- Updating quantity
- Removing items
- Viewing total cost

This module plays a crucial role in enhancing user experience by providing a smooth shopping process.

The Cart Management Module allows customers to add products to their shopping cart, update quantities, remove items, and view the total cost. It temporarily stores selected products until the customer proceeds to checkout.

➢ **Order Management Module**

The Order Management Module handles the complete order life-cycle. It allows users to place orders and track their status.

Admins can view all orders and update their status, such as:

- Pending
- Shipped
- Delivered

This module ensures efficient order processing and tracking.

➢ **Payment Module**

The Payment Module is responsible for handling financial transactions. It integrates with third-party payment gateways such as Razorpay or Stripe.

It ensures secure payment processing and generates transaction records. This module is essential for completing the purchasing process.

➢ **Database Module**

The Database Module manages data storage and retrieval. It uses MySQL to store:

- User data
- Product details
- Orders
- Payments

Relationships between tables are maintained using primary and foreign keys to ensure data integrity.

➢ **Admin Module**

The Admin Module provides complete control over the system. Admin users can:

- Manage products
- View users
- Monitor orders
- Control system operations

This module ensures efficient system management and smooth operation of the platform.

11. IMPLEMENTATION

Sales Savvy demonstrates the practical application of ReactJS, Spring Boot, and MySQL in developing a secure and scalable e-commerce platform. The project achieves its objectives and provides a strong foundation for future innovation in digital commerce.

Each module of the system is implemented independently to ensure modularity.

- Authentication Module: Implemented using Spring Security and JWT. It handles login, registration, and token validation.
- User Module: Provides functionalities for user registration, profile management, and role assignment.
- Product Module: Allows admins to perform CRUD operations on products. It includes features like product categorization and image handling.
- Cart Module: Enables users to add, update, and remove products from the cart. It calculates total price dynamically.
- Order Module: Processes user orders and manages order status.
- Payment Module: Integrates third-party payment gateways for secure transactions.

The system follows a structured workflow:

1. User registers and logs in
2. Authentication is verified using JWT
3. User browses products
4. Products are added to cart
5. User places order
6. Payment is processed
7. Order confirmation is generated

This workflow ensures smooth interaction between system components.

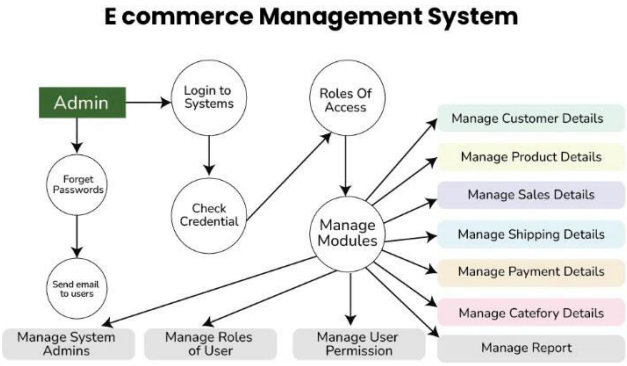

Fig3: Implemenatation Flow

12.TESTING

Testing is an essential phase in software development that ensures the system functions correctly and meets user requirements. In the Sales Savvy platform, testing is performed to identify and fix errors, improve performance, and ensure reliability.

Objectives of Testing

The main objectives of testing include:

- To verify that all functionalities work correctly
- To identify and fix bugs
- To ensure system security
- To validate performance under different conditions
- To improve user experience

Errors Identified and Fixed

Common errors identified include:

- Authentication failures
- API response errors
- Database connection issues
- UI rendering problems

These were fixed through debugging and testing.

Results of Testing

Testing results indicate that the system performs efficiently and meets all functional requirements. The application handles user interactions smoothly and ensures secure transactions.

13. RESULT AND DISCUSSION

The implementation of the Sales Savvy platform successfully achieved all the objectives defined in the project. The system provides a complete e-commerce solution with essential features such as user authentication, product management, cart operations, order processing, and payment integration.

Users are able to register and log in securely using JWT-based authentication. Once authenticated, customers can browse products, view detailed descriptions, and add items to their shopping cart. The cart module functions correctly, allowing users to update quantities and remove items. The system accurately calculates the total cost based on selected products.

The order management module enables users to place orders and track their status. Admin users can view all orders and update their status, ensuring efficient order handling. The payment module successfully integrates with external payment gateways, allowing secure and reliable transactions.

Additionally, the admin dashboard provides complete control over the system, enabling administrators to manage users, products, and orders effectively. All modules interact seamlessly, demonstrating the successful integration of frontend and backend components.

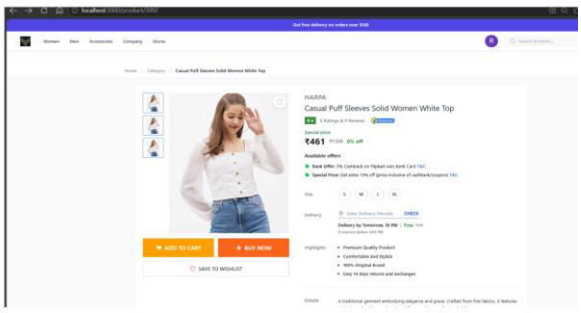

Fig4: A sample result of the website

14. ADVANTAGES OF THE SYSTEM

The Sales Savvy E-commerce platform offers several advantages that make it a reliable, efficient, and scalable solution for modern digital commerce. The system is designed using a modular architecture and modern technologies, which enhances its performance, usability, and security.

One of the major advantages of the system is its **user-friendly interface**. The frontend, developed using ReactJS, provides a clean and intuitive design that allows users to navigate easily through the platform. Customers can browse products, add items to the cart, and complete purchases without any complexity. This improves user satisfaction and encourages repeat usage.

Another significant advantage is **secure authentication and authorization**. The system uses JWT (JSON Web Token) to ensure that only authenticated users can access the platform. Role-based access control differentiates between Admin and Customer users, ensuring that sensitive operations are restricted to authorized personnel. This enhances the overall security of the system and protects user data from unauthorized access.

15. FUTURE ENHANCEMENTS

The Sales Savvy E-commerce platform has significant potential for future enhancements. By incorporating advanced technologies and additional features, the system can be transformed into a more powerful and comprehensive solution.

One of the most important future enhancements is the integration of **Artificial Intelligence (AI) and Machine Learning (ML)**. These technologies can be used to implement personalized product recommendations based on user behavior and preferences. This improves user engagement and increases sales.

The system can also be extended to support a **multi-vendor marketplace**. This feature allows multiple sellers to register and sell their products on the platform. It increases the variety of products available and enhances business opportunities.

Integration of **advanced analytics and reporting tools** is another important enhancement. These tools can provide insights into sales performance, customer behavior, and market trends. Admin users can use this information to make informed business decisions.

The system can also be enhanced with **real-time notifications**. Users can receive updates about order status, payment confirmation, and new product launches. This improves communication and user engagement. Another improvement is the implementation of **cloud-based deployment**. Hosting the system on cloud platforms such as AWS or Azure will improve scalability, reliability, and performance. It also enables the system to handle large volumes of traffic efficiently.

Security can be further enhanced by implementing **multi-factor authentication (MFA)** and advanced encryption techniques. This ensures better protection of user data and prevents unauthorized access.

16.CONCLUSION

The objective of this study was to examine how website design affects the perceived usability and user experience, as well as the online customer journey on that website. The first research question was: Does the new design facilitate good user experience on the website? The case company was looking for someone with UX expertise to evaluate their website and how well the redesigned version supports business objectives, most importantly conversion. Their main goals include also providing information for potential customers and manuals for product owners. The findings suggest that the pragmatic quality (i.e. the usability) of the website is good, but there are some areas of improvement for increasing the hedonic quality of the interaction (i.e. user experience).

- The answer to the first research question is not a certain yes or no: website scored as neutral in user experience questionnaire, so the results indicate that the website's ability to facilitate good UX is not bad – but it could be better.
- The second research question addressed the business objective: Does the new design of the website support conversion, i.e. lead intuitively to purchase? Answering the second research question with certainty is challenging as well.
- Although shopping process on the website is straightforward and received compliments from participants, the most severe usability problem is something that in a real-life setting probably would prevent some users from making a purchase decision.
- Many participants reported that if they were actually shopping for a product and felt uncertain about the details of the product (such as product's compatibility with other devices), they would probably discontinue shopping or search additional information via search engines.

The duality of user needs can be seen e.g. between inexperienced and experienced users: the less experienced users are with an interface, the more important it is to provide well-organized content and product information, whereas experienced users have more hedonic needs and expectations for the interaction. Website quality is also perceived differently by inexperienced and experienced users, as user's expertise affects their expectations of the service quality. In either case, both functional-utilitarian needs and hedonic needs need to be addressed with the design to serve all users.

REFERENCES

1. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
2. Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
3. Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-Down Approach* (7th ed.). Pearson.
4. Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson.
5. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw- Hill.
6. Freeman, E., Robson, E., Sierra, K., & Bates, B. (2014). *Head First Design Patterns*. O'Reilly Media.
7. Walls, C. (2022). *Spring in Action* (6th ed.). Manning Publications. Sharma, S., & Jain, A. (2019).*Spring Boot in Practice*. BPB Publications.
8. Banks, A., & Porcello, E. (2020). *Learning React* (2nd ed.). O'Reilly Media.Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.