Ecommerce

Team Details: Ankit kumar and Md Saheb Ali

Adm.no: 23scse1290048 & 23scse12890007

Section:1

SUMITED TO

Dr Dhirendra Shukla

Objective

Objective:

Introduction to E-commerce Website Development

E-commerce does involve seamless buying experiences thus making digital platforms important than any other medium.
Key functions to implement:

Secure login and registration for users and admins: Authentication could be achieved by using Java frameworks, such as Spring Security.
Product Management: Admin dashboard that allows product CRUD operations with creating new products, updating the inventory of any product by a set of features like price, description, and images, etc.
Shopping Cart: Manage the number of products available in real-time. Customers may add or remove products, increase/decrease quantity, and save their carts.
Checkout and payment gateway integration: Cart-to-order simple placement through any available payment gateway, such as Stripe or PayPal API, with secure checkout.
Technological Stack:

The Spring Boot technology also supports rapid development. RESTful APIs support high-performance communication between frontend and backend.
In rapid prototyping environments, H2 (in-memory) must be used, whereas MySQL would be used for scalable product, user, and order management.
Frontend: This is a responsive and intuitive design using Bootstrap that works well on various devices.
Security: It uses SSL encryption, secure cookies, and JWT for protecting user information and payments.
Easy Development Process:

Frameworks and Tools:
It supports security, databases and APIs, and speeds up the back-end development by minimizing configuration.
H2 Database: High-impact prototyping and testing with setup ease. For production just switch to MySQL or PostgreSQL.
.

Problem Statement

1. Traditional Development Issues:

Designing an e-commerce platform is extremely complicated with the management of the database, that itself requires security protocols and ensures responsive design.
Resources:
Most resources used for new construction-from scratch developments-expensive with many developers, timerequirements for testing much longer than green-field comparison benchmarks.
Security and Scalability: Secure transactions, with increasing traffic of users, require strong infrastructure and sound code; the task for the development team.
2. Rapid Prototyping Requirement:

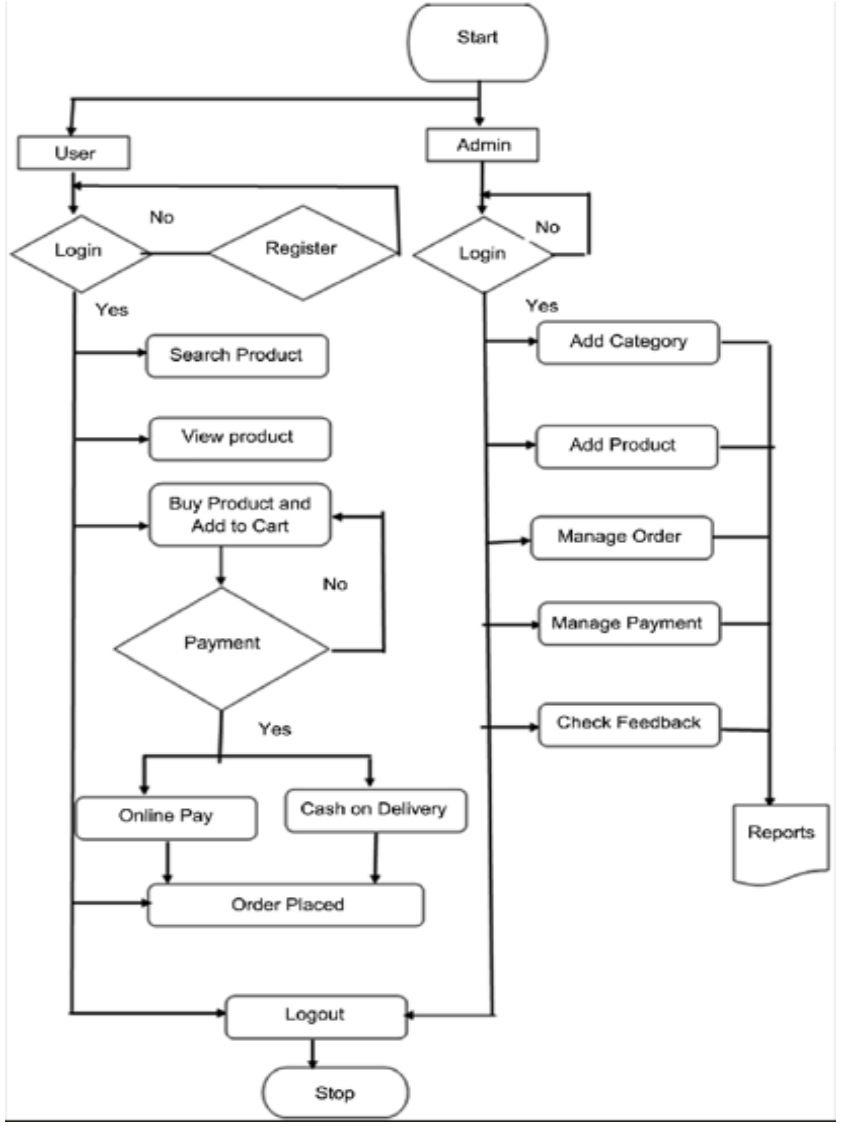
Faster Market Testing: Most businesses need a rapid MVP release for them to test the concept and get some feedback prior to investing heavily.
Adaptability: The prototyping process must be flexible enough to be changed based on user feedback or the company's shifting needs.
Development Overhead Reduced It reduces the infrastructure and boilerplate code. This will allow the core functionalities to be concentrated on more when developing it.
User Experience: 3. Focus

Requirements of the intuitive interface: Smooth navigation of categories of products, smooth management of the cart and checkout. Account Management Should Require No Learning Users should find it extremely easy to manage accounts, orders, and preferences. Features in the eCommerce arena, specifically the secure login, product browsing, cart functionality, and payment gateways, are very important for good user experience and to reduce the cart abandonment rate.

Flow Diagram

Flow Diagram: To illustrate the user journey, a simple flow diagram should cover the following steps:

1. **User Registration/Login**
   - Users create an account or log in to access features.
2. **Browse Product Catalog**
   - Users can view products categorized for easy navigation.
3. **Add Items to Shopping Cart**
   - Users can select products and add them to their shopping cart.
4. **Checkout Process**
   - Users can proceed to checkout, input shipping/payment information.
5. **Order Confirmation**
   - Users receive confirmation and details of their order.



Technical Stacks

Technical Stacks:

- **Backend:**
  - **Framework: Spring Boot**
    - Ideal for building Java applications with minimal configuration. Use Spring Initializr to create a new project with necessary dependencies.
  - **Database: H2 (embedded)**
    - An in-memory database that requires minimal setup, perfect for development and prototyping.
  - **Security: Spring Security**
    - Provides robust security features for user authentication and authorization.
- **Frontend:**
  - **Languages: HTML, CSS, JavaScript**
    - Basic building blocks for creating the web interface.
  - **Frameworks: Bootstrap (optional)**
    - A front-end framework that allows for responsive design, helping make the site user-friendly on various devices.
- **Payment Integration:**
  - **Stripe (sandbox mode)**
    - A simple API for processing payments. Use Stripe's sandbox environment for testing without real transactions.

Summarize Project Phases

Phases:

=>**Installation and Admin Section**

Create a Spring Boot project using spring-initializr.

Configure an Embedded H2 Database.

Develop simple prototypes for User, Product, Order, and Cart.

=>Dev API

RESTful APIs

User Controller: sign-up and login.

Product Controller: Fetch and list the details of products.

Cart Controller deals with add to cart, view cart, and check out.

Use Spring Security for user authentication.

=>**Front-end Development**

Use a Simple HTML template or Bootstrap in UI.

Create major pages

Homepage: Show products in grid.

Login Page Users can login into this page.

Cart page: items in the cart, total, and checkout link.

Checkout Page: User information accompanied by a payment form.

=>**Pumping Money**

Use Stripe to create a basic payment form.

To demonstrate payments, redirect to Stripe's checkout in sandbox mode.

=>**Final Testing and Touches**

Test the entire user flow: Registration and login.

Fix all the bugs, basic logging.

Track the initiation of this project and subsequent improvements..

Key Features Implemented

Key Features Implemented:

- **User Registration/Login:**
  - Basic user registration and authentication using Spring Security.
- **Product Catalog:**
  - Dynamic display of products fetched from the H2 database.
- **Shopping Cart:**
  - Functionality to add/view items in the cart using session storage.
- **Checkout Process:**
  - Integration of Stripe for handling payments.
- **Order Confirmation:**
  - Display a summary of the order after successful checkout.

Testing and Future Improvements

Testing:

- **User Flow Testing:**
  - Manually go through each feature to ensure proper functionality, from signing up to checking out.
- **Debugging:**
  - Use logging to trace issues and correct any bugs identified during testing.

Future Improvements:

- **UI/UX Enhancements:**
  - Improve the design using advanced CSS techniques and JavaScript for interactivity.
- **Advanced Payment Options:**
  - Explore additional payment gateways for flexibility (e.g., PayPal).
- **User Roles and Permissions:**
  - Implement different user roles (admin, customer) for more granular access control.

Conclusion

Conclusion:

- **Feasibility of Rapid Development:**
  - Successfully demonstrated that a functional eCommerce website can be developed quickly.
- **MVP Focus:**
  - Concentrated on essential features to validate the concept.
- **Expansion Opportunities:**
  - There is potential to grow the platform based on user feedback and business needs