



BLUSHEE: An E-Commerce Platform for Makeup and Skincare Products Using MERN Stack

Krishna Nayak

Department of Master in Computer Application

GIFT Autonomous, Bhubaneswar, India

krishnanayak2024@gift.edu.in

Nandan Kumar

Department of Master in Computer Application

GIFT Autonomous, Bhubaneswar, India

nandankumar2024@gift.edu.in

Shubhendu Sekhar Sahoo

Assistant Professor

Department of Master in Computer Application

GIFT Autonomous, Bhubaneswar, India

ssahoo@gift.edu.in



<https://doi.org/10.55041/ijst.v2i6.063>

Cite this Article: Nayak, K. & Kumar, N. (2026). BLUSHEE: An E-Commerce Platform for Makeup and Skincare Products Using MERN Stack. International Journal of Science, Strategic Management and Technology, 02(6). <https://doi.org/10.55041/ijst.v2i6.063>

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

ABSTRACT- The rapid growth of the beauty and skincare industry has significantly increased the demand for online shopping platforms that provide seamless user experiences, personalized product recommendations, and secure transactions. This paper presents Blushee, a modern e-commerce platform developed using the MERN Stack (MongoDB, Express.js, React.js, and Node.js) for makeup and skincare product management and online purchasing. The platform provides features such as user authentication, product categorization, shopping cart management, wishlist functionality, secure payment

integration, order tracking, and an admin dashboard for inventory management. The frontend is designed with a responsive and aesthetic user interface to enhance customer engagement, while the backend ensures scalability, security, and efficient database management. The project demonstrates how full-stack web technologies can be utilized to build a robust and user-friendly beauty product marketplace.

Keywords: MERN Stack, E-Commerce, Makeup Products, Skincare Platform, React.js, MongoDB, Node.js, Express.js, Online Shopping.



I. INTRODUCTION

The rapid growth of internet technologies and digital commerce has significantly changed the way consumers purchase beauty and skincare products. Traditional shopping methods require customers to physically visit stores, which can be time-consuming and less convenient. In contrast, online shopping platforms provide users with the ability to browse products, compare prices, read reviews, and purchase items from the comfort of their homes. The beauty and skincare industry, in particular, has witnessed substantial growth in online sales due to increasing customer awareness regarding cosmetics, self-care products, and skincare routines.

Despite the availability of various e-commerce platforms, many systems are designed for general retail products and do not specifically focus on delivering a personalized and aesthetically pleasing experience for makeup and skincare consumers. Existing systems may also lack efficient product categorization, modern user interfaces, and advanced management functionalities for vendors and administrators. Therefore, there is a need for a specialized beauty-focused e-commerce platform that combines functionality, security, and modern design principles.

To address these issues, the project Blushee was developed using the MERN Stack technology. The platform aims to provide a complete online marketplace for beauty and skincare products while ensuring smooth navigation, responsive design, secure authentication, and scalable system architecture. The application includes separate modules for users, vendors, and administrators to simplify product management, order processing, and customer interactions.

II. LITERATURE REVIEW

Several online shopping platforms and research studies have contributed to the development of modern e-commerce systems. Popular beauty platforms such as Nykaa and Sephora provide a wide range of cosmetic and skincare products with online purchasing facilities. These platforms have successfully improved digital beauty shopping experiences through product recommendations, online reviews, and secure payment systems. Similarly, general e-commerce platforms like Amazon offer scalable shopping infrastructures with advanced logistics and order management features.

Research on modern web application development highlights the importance of full-stack JavaScript technologies such as the MERN Stack. React.js is widely used for creating responsive and dynamic user interfaces, while Node.js and Express.js support scalable backend services and efficient API handling. MongoDB is commonly used for flexible and efficient NoSQL database management. Existing research also emphasizes the growing importance of secure authentication systems, responsive design, cloud-based deployment, and real-time data processing in e-commerce applications.

Although many existing systems provide efficient online shopping services, several limitations still exist, including poor UI personalization, lack of dedicated vendor management, limited scalability, and complex user experiences. Blushee aims to overcome these limitations by integrating modern technologies, responsive aesthetics, secure backend architecture, and role-based dashboards specifically designed for beauty and skincare product management.



III. PROBLEM STATEMENT

Traditional cosmetic and skincare shopping methods often require customers to visit multiple physical stores to compare products, prices, and reviews. This process can consume significant time and effort, especially when customers seek personalized beauty products suitable for their skin type and preferences. Additionally, many general e-commerce platforms fail to provide a specialized shopping experience tailored specifically to beauty and skincare products.

Existing systems may also suffer from poor product organization, limited filtering options, lack of wishlist functionality, inefficient order management, and non-responsive user interfaces. Vendors often face challenges in managing inventory and customer orders efficiently, while administrators require secure and scalable systems to monitor platform activities and product performance.

Therefore, there is a need for an advanced e-commerce platform that provides a user-friendly interface, secure authentication, effective product management, responsive design, and dedicated modules for users, vendors, and administrators. The proposed system, Blushee, aims to solve these problems using MERN Stack technologies and modern web development practices.

IV. SYSTEM ARCHITECTURE

The architecture of Blushee follows a three-tier MERN Stack architecture:

4.1 Frontend Layer

- Developed using React.js
- Responsive UI using Tailwind CSS
- Dynamic routing using React Router

4.2 Backend Layer

- Node.js runtime environment

- Express.js REST APIs
- JWT authentication
- Middleware for authorization and security

4.3 Database Layer

- MongoDB Atlas cloud database
- Product collections
- User collections
- Order management collections

V. OBJECTIVES

The primary objective of the Blushee platform is to develop a secure and responsive online marketplace dedicated to makeup and skincare products. The system aims to simplify the online shopping experience for users while providing efficient management functionalities for vendors and administrators. Another major objective is to create an aesthetically appealing user interface that attracts beauty and skincare consumers and enhances user engagement.

The platform also aims to implement secure authentication and authorization mechanisms using JWT-based login systems. Product categorization, wishlist management, shopping cart functionality, and order tracking are included to improve overall usability. Furthermore, the project focuses on building a scalable architecture capable of handling large datasets and multiple user interactions efficiently using MERN Stack technologies.

VI. PROPOSED SYSTEM

The proposed system, Blushee, is a full-stack e-commerce platform developed using MongoDB, Express.js, React.js, and Node.js. The platform is designed to provide a smooth and visually appealing shopping experience for users interested in beauty and skincare products. The frontend interface is built using React.js and



Tailwind CSS to ensure responsiveness and aesthetic consistency across devices.

The backend of the system is developed using Node.js and Express.js, which handle RESTful APIs, authentication, product management, order processing, and payment integration. MongoDB is used as the primary database for storing user information, product details, order history, and vendor data. JWT authentication is implemented to secure user sessions and protect sensitive information.

The platform consists of three major panels: user, vendor, and admin. Users can browse products, add items to their cart or wishlist, place orders, and track deliveries. Vendors can manage their products and inventory, while administrators can monitor users, products, and overall platform performance through a centralized dashboard.

VII. MODULES OF THE SYSTEM

The Blushee platform is divided into multiple functional modules to ensure smooth operation and efficient management of the e-commerce system. Each module is responsible for handling specific functionalities related to users, vendors, and administrators.

A. User Module

The user module is responsible for managing customer interactions with the platform. Users can register and log in securely using email and password authentication. After successful authentication, users can browse products, search for items based on categories, brands, or price ranges, and view detailed product descriptions.

Users can add products to their shopping cart or wishlist for future purchases. The cart module dynamically updates product quantities and calculates total prices. Customers can also place orders, make online payments, and track their order status. Additionally, users can manage

their profile information, addresses, and order history through the user dashboard.

B. Vendor Module

The vendor module allows sellers to manage their products and monitor sales activities. Vendors can upload product images, descriptions, prices, and stock details through a dedicated dashboard. They can also update inventory information and track customer orders related to their products.

This module improves product management efficiency and enables vendors to maintain accurate stock records. Vendors can also analyze sales performance and customer engagement through dashboard analytics.

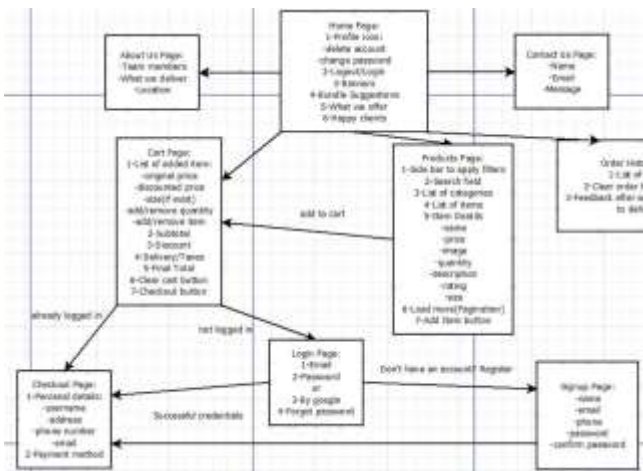
C. Admin Module

The admin module provides complete control over the platform. Administrators can manage users, vendors, products, categories, and customer orders through a centralized dashboard. The admin panel allows administrators to approve or remove products, monitor transactions, and maintain platform security.

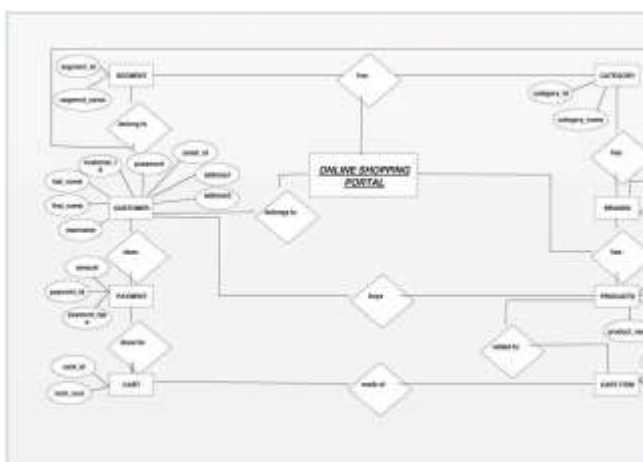
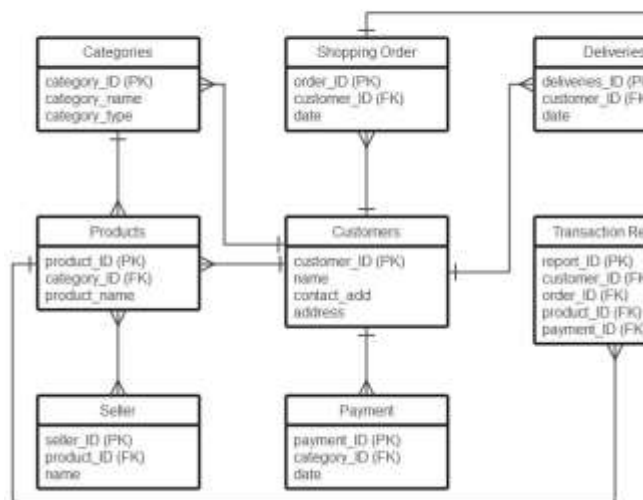
Administrators can also generate reports related to sales performance, user activities, and inventory management. This module ensures proper monitoring and smooth operation of the entire system.

VIII. DATABASE DESIGN

Add schema diagrams or ER diagrams.



ERD for Online Shopping System



The database design of Blushee is developed using MongoDB, which is a NoSQL database management system capable of handling large amounts of dynamic and unstructured data efficiently. The database consists of multiple collections such as Users, Products, Orders, Wishlist, Reviews, Vendors, and Payments.

The Users collection stores customer and administrator information, including login credentials, addresses, and order history. The Products collection contains product details such as product name, category, price, stock availability, brand information, and images. The Orders collection manages customer purchases, payment status, shipping details, and order tracking information.

MongoDB provides flexibility in schema design, making it suitable for handling frequently changing e-commerce product data. The database design also ensures efficient retrieval and storage of records while supporting scalability for future system growth.

IX. FEATURES OF BLUSHEE

Blushee provides several advanced features that improve customer experience and simplify platform management. The system is designed with a modern pastel-themed user interface that attracts beauty and skincare consumers while maintaining responsive performance across multiple devices.

The platform includes secure JWT-based authentication to protect user information and maintain session security. Product categorization and search filtering allow users to easily discover relevant products based on their preferences. Wishlist functionality enables customers to save products for future purchases.

The shopping cart system dynamically updates selected products and calculates total order prices. Integrated payment gateways ensure secure online transactions, while order tracking functionality helps users monitor delivery



progress. The admin dashboard provides centralized management for products, orders, and users.

The system also supports responsive design principles, allowing users to access the platform efficiently through desktop, tablet, and mobile devices.

X. ADVANTAGES OF THE SYSTEM

The Blushee platform provides multiple advantages for users, vendors, and administrators. The responsive and visually appealing user interface enhances customer engagement and improves shopping experiences. Secure authentication and payment integration increase platform reliability and customer trust.

The MERN Stack architecture ensures scalability and efficient performance even when handling large datasets and multiple concurrent users. Vendors benefit from efficient inventory management and sales monitoring features, while administrators gain centralized control over the platform.

The platform also reduces manual efforts involved in traditional shopping methods and provides customers with convenient access to beauty and skincare products from any location.

XI. FUTURE SCOPE

The future scope of Blushee includes several advanced features and technological improvements. Artificial Intelligence and Machine Learning algorithms can be integrated to provide personalized skincare and makeup recommendations based on user preferences and skin types.

Augmented Reality technology can be implemented to enable virtual makeup try-on experiences for customers. Voice-based product search and chatbot support can further improve user interaction and accessibility. Mobile applications for Android and iOS platforms can

also be developed to expand the platform's accessibility.

Additional future enhancements may include multi-language support, advanced analytics dashboards, loyalty reward systems, subscription-based beauty boxes, and social media integration for product promotions and customer engagement.

XII. IMPLEMENTATION METHODOLOGY

The implementation of the Blushee platform follows a systematic software development methodology to ensure efficient project execution and high-quality output. The development process includes requirement analysis, system design, frontend development, backend implementation, database integration, testing, deployment, and maintenance phases.

Initially, user requirements and functional specifications were collected to identify the core functionalities required for the e-commerce platform. These requirements included user authentication, product management, shopping cart functionality, order processing, payment integration, and administrative control systems. Based on these requirements, a detailed system architecture and database design were prepared.

The frontend development phase was carried out using React.js and Tailwind CSS. React.js enabled the creation of reusable components, efficient state management, and responsive interfaces. Tailwind CSS was used to design a visually attractive pastel-themed user interface suitable for beauty and skincare products. Responsive web design principles were implemented to ensure compatibility across desktop, tablet, and mobile devices.

The backend system was developed using Node.js and Express.js. RESTful APIs were created for handling authentication, product management, cart operations, payment processing, and order tracking. Middleware functionalities were implemented to ensure



secure communication between the client and server. JWT authentication mechanisms were integrated to secure user sessions and prevent unauthorized access.

MongoDB was used as the primary database management system due to its flexibility and scalability. Separate collections were created for users, products, orders, vendors, and reviews. Cloudinary services were integrated for efficient image storage and retrieval of product images.

After development, the system underwent multiple testing phases including unit testing, integration testing, and user acceptance testing to identify and resolve bugs and performance issues. Finally, the application was deployed using cloud hosting platforms to ensure accessibility and scalability.

XIII. USER INTERFACE DESIGN

The user interface design of Blushee focuses on simplicity, visual appeal, and ease of navigation. Since the platform targets beauty and skincare consumers, special attention was given to creating an aesthetically pleasing design with modern pastel color schemes, smooth transitions, and responsive layouts.

The homepage of the platform includes featured products, promotional banners, trending categories, and search functionality. Navigation menus are organized to allow users to quickly access skincare, makeup, haircare, and beauty accessories. Product cards display images, prices, ratings, and quick action buttons for adding items to the cart or wishlist.

The product details page provides comprehensive information regarding ingredients, product descriptions, usage instructions, customer reviews, and ratings. The cart interface dynamically updates selected products and calculates pricing information in real time.

The admin dashboard is designed separately with analytical widgets, inventory statistics,

order management tables, and user management controls. Vendors also receive dedicated dashboards to upload products, monitor stock levels, and track sales performance.

The responsive design ensures consistent user experiences across all devices. Mobile responsiveness is achieved using flexible grids, media queries, and optimized component structures.

XIV. AUTHENTICATION AND SECURITY

Security is one of the most important aspects of any e-commerce platform. Blushee implements multiple security mechanisms to protect user information, payment details, and platform data from unauthorized access and cyber threats.

The platform uses JWT (JSON Web Token) authentication for secure user login and session management. Passwords are encrypted using hashing algorithms before being stored in the database. Authentication middleware validates tokens before allowing access to protected routes and functionalities.

Role-based authorization mechanisms are implemented to differentiate access permissions between users, vendors, and administrators. Sensitive operations such as product deletion, order modification, and administrative settings are restricted to authorized users only.

Input validation and sanitization techniques are applied to prevent malicious attacks such as SQL injection, cross-site scripting (XSS), and unauthorized data manipulation. HTTPS protocols and secure payment gateway integrations ensure encrypted communication during online transactions.

Cloud-based storage systems also provide backup and data recovery mechanisms to protect platform information from accidental data loss.



XV. PAYMENT GATEWAY INTEGRATION

Blushee integrates secure online payment gateways to provide seamless and reliable transaction processing for customers. Payment integration enables users to complete purchases using debit cards, credit cards, UPI systems, digital wallets, and internet banking services.

The payment system is connected with platforms such as Stripe or Razorpay to process transactions securely. During checkout, payment requests are transmitted through encrypted APIs to ensure data confidentiality and transaction integrity.

After successful payment verification, order confirmation details are stored in the database and displayed to users through the order management interface. Failed transactions are automatically detected and appropriate error messages are provided to users for retry operations.

The payment gateway system also supports refund handling, transaction history management, and invoice generation. Integration of secure payment services increases customer trust and enhances overall platform reliability.

XVI. TESTING AND VALIDATION

Testing and validation play an important role in ensuring the functionality, reliability, and security of the Blushee platform. Multiple testing methodologies were implemented throughout the development lifecycle to identify bugs and improve overall system performance.

Unit testing was performed to verify individual components such as login forms, product APIs, cart calculations, and database operations. Integration testing ensured smooth communication between frontend interfaces, backend APIs, and database services.

System testing was conducted to validate the complete workflow of the application, including user registration, product browsing, payment processing, and order tracking. Security testing was also performed to identify vulnerabilities related to authentication, session management, and unauthorized access.

Performance testing evaluated system responsiveness under varying levels of user traffic and database load. Cross-browser testing ensured compatibility with major web browsers such as Chrome, Firefox, Edge, and Safari.

User acceptance testing was carried out with sample users to gather feedback regarding usability, interface design, and overall shopping experiences. The testing process significantly improved system stability and user satisfaction.

XVII. DEPLOYMENT AND MAINTENANCE

The deployment of Blushee was carried out using cloud-based hosting platforms to ensure high availability, scalability, and accessibility. The frontend application developed using React.js was deployed using platforms such as Vercel or Netlify, while backend APIs were hosted using cloud server environments like Render or Railway.

MongoDB Atlas cloud services were utilized for database hosting and management. Environment variables and API keys were securely configured to protect sensitive information during deployment.

Continuous monitoring and maintenance activities are necessary to ensure smooth platform operation after deployment. Maintenance includes updating dependencies, fixing security vulnerabilities, optimizing database performance, and improving user experience based on customer feedback.

Future maintenance plans also involve introducing new features, enhancing UI designs,



integrating AI-based recommendation systems, and improving server-side optimization for handling increased traffic.

XVIII : PERFORMANCE OPTIMIZATION

The performance of the Blushee platform is optimized using efficient backend architecture and database management techniques. The system provides fast API responses by using optimized routing and minimal server load.

It ensures that database queries are structured efficiently using MongoDB indexing and proper schema design. The frontend is optimized using React.js component reusability to reduce unnecessary rendering.

This improves overall system speed and reduces loading time for users. It ensures smooth handling of multiple concurrent users without performance degradation. Which results in a fast, responsive, and scalable e-commerce platform.

XIX: PROPOSED SYSTEM

The proposed Blushee system is developed using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The system is designed to provide a complete e-commerce solution for beauty and skincare products with dedicated modules for users, vendors, and administrators.

The system provides user-friendly features such as product browsing, wishlist management, shopping cart functionality, and secure payment integration. It ensures secure authentication using modern security techniques such as encrypted login systems and token-based authentication.

This improves data security and prevents unauthorized access to sensitive user information. It ensures smooth communication between frontend and backend through RESTful APIs. Which results in a reliable and scalable e-

commerce platform capable of handling multiple users simultaneously.

XX: SYSTEM ARCHITECTURE

The architecture of Blushee follows a three-tier MERN Stack architecture consisting of the frontend layer, backend layer, and database layer. The system is designed to ensure scalability, maintainability, responsiveness, and secure communication between users and the server. The architecture enables efficient handling of user requests, product management, order processing, and authentication functionalities.

The frontend layer is developed using React.js, which provides a dynamic and responsive user interface for customers, vendors, and administrators. React Router is used for page navigation, while Tailwind CSS is utilized to create a modern pastel-themed user interface suitable for beauty and skincare applications. The frontend communicates with backend services using RESTful APIs.

The backend layer is implemented using Node.js and Express.js. Express.js handles API routing, middleware integration, authentication, and server-side logic. The backend processes user requests such as login, registration, product fetching, cart management, and payment processing. JWT authentication is implemented to secure user sessions and protect sensitive data from unauthorized access.

XXI. RESULTS AND DISCUSSION

The implementation of Blushee successfully achieved the primary objectives of developing a secure, responsive, and user-friendly e-commerce platform for makeup and skincare products. The platform demonstrated efficient product management, smooth user interactions, secure payment processing, and responsive design across multiple devices.



Testing results indicated that the system provides fast page rendering, accurate product filtering, and efficient database operations. Users were able to browse products, manage wishlists, add items to carts, and complete transactions without significant delays.

The admin and vendor dashboards simplified inventory management and order monitoring processes. JWT authentication mechanisms effectively protected user data and restricted unauthorized access to sensitive operations.

The use of MERN Stack technologies provided flexibility, scalability, and maintainability for future enhancements. Overall, the platform successfully fulfilled the requirements of a modern beauty-focused e-commerce system while maintaining high usability and performance standards.

XXII: AUTHENTICATION SYSTEM

The authentication system in Blushee is designed to ensure secure user login and access control. The system provides JWT-based authentication that generates secure tokens after successful login.

It ensures that only verified users can access protected routes such as cart, checkout, and order history. Passwords are encrypted before being stored in the database using hashing techniques.

This improves system security by preventing unauthorized access and protecting sensitive user data. It ensures that each session is validated before granting access to restricted features. Which results in a secure and trustworthy user authentication system.

XXIII. CONCLUSION

Blushee successfully demonstrates the implementation of a modern e-commerce platform dedicated to makeup and skincare products using MERN Stack technologies. The system provides a responsive and user-friendly environment for online shopping while ensuring secure authentication, efficient product management, and scalable architecture.

The platform addresses the limitations of traditional shopping methods by offering advanced features such as wishlist management, order tracking, responsive design, and centralized administration. The use of React.js, Node.js, Express.js, and MongoDB enables efficient communication between the frontend and backend while maintaining system performance and flexibility.

The project highlights the effectiveness of full-stack web technologies in developing scalable and secure e-commerce applications tailored specifically for beauty and skincare industries. Future enhancements such as AI recommendations and AR-based virtual try-on systems can further improve customer experiences and platform functionality.

REFERENCES

- [1] MongoDB, "MongoDB Documentation," 2025. <https://www.mongodb.com/docs/>
- [2] React, "React Official Documentation," 2025. <https://react.dev/>
- [3] Node.js, "Node.js Official Documentation," 2025. <https://nodejs.org/en/docs>
- [4] Express.js, "Express.js Guide," 2025. <https://expressjs.com/>
- [5] Tailwind CSS, "Tailwind CSS Documentation," 2025. <https://tailwindcss.com/docs>
- [6] JWT, "JSON Web Token Introduction," 2025. <https://jwt.io/introduction>



[7] MDN Web Docs, “JavaScript and Web Development Guide,” 2025. <https://developer.mozilla.org/>

[8] GeeksforGeeks, “MERN Stack Tutorial,” 2024. <https://www.geeksforgeeks.org/mern-stack/>

[9] W3Schools, “Web Development Tutorials,” 2025. <https://www.w3schools.com/>

[10] Amazon, “E-commerce Platform Overview,” 2025. <https://www.amazon.com/>

[11] Nykaa, “Beauty and Cosmetics E-commerce Platform,” 2025. <https://www.nykaa.com/>

[12] Sephora, “Global Beauty Retail System,” 2025. <https://www.sephora.com/>

[13] Shopify, “E-commerce Platform Architecture,” s

[14] REST API, “RESTful Web Services Concepts,” 2025. <https://restfulapi.net/>

[15] Google Cloud, “Scalable Web Application Architecture,” 2025. <https://cloud.google.com/architecture>

[16] Microsoft Learn, “Web Application Development Best Practices,” 2025. <https://learn.microsoft.com/>

[17] AWS, “Cloud Computing for Web Applications,” 2025. <https://aws.amazon.com/architecture/>

[18] Stack Overflow, “Web Development Community Insights,” 2025. <https://stackoverflow.com/>

[19] FreeCodeCamp, “Full Stack Development Learning Resources,” 2025. <https://www.freecodecamp.org/>

[20] OWASP, “Web Application Security Best Practices,” 2025. <https://owasp.org/>