



(RESEARCH ARTICLE)



A study on the design and development of a full-stack E-commerce web application with integrated payment processing via stripe

Nwankwo Chuka Chukwuemeka *

Faculty of Computing, Engineering and Science, University of South Wales.

World Journal of Advanced Research and Reviews, 2026, 30(03), 955-962

Publication history: Received on 01 May 2026; revised on 10 June 2026; accepted on 12 June 2026

Article DOI: <https://doi.org/10.30574/wjarr.2026.30.3.1650>

Abstract

Background: The e-commerce industry has witnessed appreciable growth driven by evolving consumer behaviours and technological advancements. Despite the availability of off-the-shelf solutions, many businesses require tailor-made platforms that better serve specific operational needs.

Objective: This paper presents the design and development of a functional, responsive, and secure e-commerce website built from scratch using HTML, CSS, JavaScript, Node.js, Express.js, MongoDB, and the Stripe payment gateway.

Methods: An iterative, agile-informed methodology was adopted, encompassing requirements analysis, UI/UX design, frontend and backend development, third-party payment integration, and multi-phase testing including unit, integration, and user acceptance testing.

Results: The completed platform successfully delivers user registration and authentication, product browsing with filtering, dynamic cart management, and a seamless Stripe-powered checkout. Responsive design was achieved via CSS media queries and Bootstrap, while MongoDB underpinned robust data management.

Conclusion: The study demonstrates that a fully functional, secure, and scalable e-commerce website can be built using open-source web technologies. Future work includes hybrid mobile app development, AI-powered chatbots, advanced analytics, and multi-currency support.

Keywords: e-Commerce; Web Development; HTML; CSS; Javascript; Node. Js; Mongoddb; Stripe; Responsive Design; Payment Gateway

1. Introduction

The ever-changing and insatiable consumer wants have forced businesses to continue to evolve in order to satisfy customer yearnings for more comfortable shopping experiences. This has expanded e-commerce and further transformed how consumers and firms transact business. Customers can now search, compare, make informed decisions, and buy products from the comfort of their homes, with purchased products shipped directly to their provided shipping addresses through online shopping platforms (Chaffey, 2020).

For businesses to thrive and stay competitive in the digital marketplace, they must offer a dynamic, safe, and user-friendly online platform as consumer expectations change. The construction of an e-commerce website entails careful planning, design, execution, and implementation, with the target area being user experience, data security, and transaction security. HTML, CSS, and JavaScript are the core technologies responsible for developing responsive, user-

* Corresponding author: Nwankwo Chuka Chukwuemeka Email: chuka.nwankwo2@gmail.com

friendly, and dynamic websites capable of providing an endearing environment for users (Duckett, 2011). In addition, a secured payment gateway is required to handle transactions and ensure data protection.

There are many off-the-shelf solutions for e-commerce, but they often lack the flexibility and customisation needed to meet specific business requirements. This paper addresses the problem of developing a tailor-made e-commerce website from scratch, with emphasis on user-friendliness, data protection, secured transactions, scalability, and maintainability. The aim is to develop a website that not only grants users access to search, select, and buy products easily, but also ensures secure payment transactions and data protection.

1.1. Objectives

The principal objectives of this study are: (i) to analyse market and consumer needs through comprehensive research identifying target audiences, behaviours, and expectations (Chaffey & Ellis-Chadwick, 2019); (ii) to define the scope and features of the website including user registration, authentication, product selection, and secure payment processing; (iii) to design a user-centric interface that improves the shopping experience and conforms to UX/UI best practices to maximise customer engagement (Krug, 2014); (iv) to implement secure payment and data protection by integrating a payment platform that complies with relevant regulations and implements data encryption to build customer trust (Pavlou, 2011); (v) to develop a robust backend using MongoDB for managing inventory and processing orders; and (vi) to test and optimise for performance and security through unit testing, user testing, load testing, and security testing (Huang, Huang & Liu, 2013).

1.2. Scope

The scope encompasses design and development of an e-commerce website enabling users to register, authenticate, login, search and compare products, add items to a shopping cart, make payments, and provide shipping information. The study focuses on frontend development using HTML, CSS, and JavaScript, with backend functionality using Node.js and Express.js to handle payments, database operations, and user authentication.

2. Literature review

2.1. Evolution of E-Commerce

Electronic Commerce (E-Commerce) has changed how businesses and customers connect. When a CD was sold over the American retail platform NetMarket in 1994, it became the first recorded e-commerce transaction (Khurana, 2020). Amazon launched as a small online bookstore in 1994, and by its first month had distributed books to 45 countries and all 50 states (Stone, 2013). Amazon's customer-centric innovations including one-click shopping, personalised recommendations, and user evaluations set new standards for e-commerce (Brandt, 2011). Similarly, eBay was created by Pierre Omidyar in September 1995 as a peer-to-peer marketplace and grew rapidly to hold over two million auctions by 1997 (Kaplan, 2002).

The dot-com boom of the late 1990s saw venture capital in internet firms rise from \$3 billion in 1995 to over \$40 billion in 1999, with NASDAQ rising from under 1,000 to 5,000 points (Cassidy, 2002). The subsequent bust, beginning in March 2000, saw NASDAQ fall to approximately 1,100 points by October 2002 (Cassidy, 2002), but the companies that survived, such as Amazon and eBay, went on to become giants of the sector (Morris & Alam, 2001).

2.2. Web Technologies in E-Commerce

The evolution of e-commerce is closely tied to advancements in HTML and CSS. Tim Berners-Lee developed the first HTML in 1991 as a basic markup language for organising content on the internet (Berners-Lee, 1999). The introduction of CSS in 1996 enabled developers to separate content from presentation, producing more complex and visually appealing websites (Meyer, 2007). The release of HTML5 in 2008 and CSS3 in the late 2000s brought native multimedia support, animations, transitions, and flexbox for responsive design (Hickson & Hyatt, 2014; Bos et al., 2011), allowing e-commerce websites to deliver increasingly interactive and device-adaptive experiences (Chaffey, 2015).

JavaScript, introduced by Netscape in 1995, has grown into an essential tool in e-commerce development. It enables dynamic user interfaces capable of refreshing content without page reloads, which significantly increases conversion rates (Flanagan, 2011). Asynchronous data loading via AJAX allows real-time updates without disrupting user interaction (McFarland, 2008). JavaScript frameworks such as React, Angular, and jQuery further enhance functionality and maintainability (Facebook, 2021; Google, 2021; Resig & Bibeault, 2013). The introduction of Node.js in 2009

extended JavaScript to the server side, providing an event-driven, non-blocking I/O architecture well-suited to scalable network applications (Tilkov & Vinoski, 2010).

2.3. Payment Systems

Payment systems play a major role in the convenience and security of online transactions. Early e-commerce required merchants to directly handle sensitive card information, presenting significant security concerns (Schneider, 2017). Secure Socket Layer (SSL) encryption was developed to mitigate these vulnerabilities. PayPal, developed in 1998, transformed online payments by serving as a middleman between customers and sellers (Kaplan, 2002). Mobile payments using NFC technology, exemplified by Apple Pay and Google Wallet, have further improved transaction speed (Ngai & Gunasekaran, 2007). Blockchain-based cryptocurrencies offer alternatives that reduce transaction costs and speed cross-border payments (Catalini & Gans, 2016). Stripe, the payment gateway adopted in this study, provides security, ease of integration, global reach, and compliance with PCI-DSS industry standards (Sterling, 2017).

2.4. Security and Privacy

Security concerns in e-commerce include cyberattacks such as malware, phishing, and hacking that can compromise sensitive consumer data (Schneider, 2017). E-commerce websites typically deploy TLS/SSL protocols to encrypt data transferred between users and the server. Intrusion detection systems and firewalls also help protect against unauthorised access (Grossman, 2011). Privacy legislation such as the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act (CCPA) in the United States impose stringent data protection standards on organisations (Zhu & Jiang, 2021). Biometric authentication and two-factor authentication (2FA) further reduce the risk of unauthorised account access (Chukwudi, 2026).

2.5. Advanced Technologies and Future Directions

Artificial intelligence and machine learning are now integral to e-commerce, enabling personalised recommendation engines and AI-powered chatbots that handle consumer inquiries efficiently (Kumar et al., 2019; Zhao et al., 2019). Augmented reality allows customers to visualise products in their environments before purchase, as demonstrated by IKEA's AR application (Hilken et al., 2017). Blockchain technology improves transaction transparency and trust by providing a decentralised, immutable ledger (Tapscott & Tapscott, 2016). Progressive Web Apps (PWAs) combine the best features of web and mobile applications, delivering fast loading, offline functionality, and cross-platform compatibility (Russell, 2015), with companies like Alibaba and Flipkart reporting notable increases in conversion rates following PWA adoption (Wadhwa, 2018).

3. Methodology

3.1. Research and Development Approach

The study adopted an iterative, agile-informed methodology to guarantee continuous improvement throughout the development lifecycle (Schwaber & Sutherland, 2020). Requirements were gathered through literature review, analysis of existing e-commerce platforms, and consumer opinion sampling. The development followed five principal phases: (i) planning, in which study requirements, design layouts, and system architecture were defined; (ii) design and development, in which HTML, CSS, and JavaScript were used to build the frontend; (iii) payment integration, in which Stripe was incorporated for secure payment processing; (iv) testing and debugging; and (v) deployment and monitoring.

3.2. System Architecture

The website was constructed on a three-tier architecture. The frontend tier was implemented using HTML5, CSS3, Bootstrap, and JavaScript (EJS templating). The backend tier was built using Node.js and Express.js to handle business logic and API integration. The data tier used MongoDB as the NoSQL database for managing products, orders, and user data. Stripe served as the third-party payment processing service. This architecture was chosen because Node.js and MongoDB form a cohesive JavaScript-based stack that reduces context switching for developers and supports high concurrency (Tilkov & Vinoski, 2010; Chodorow, 2013).

3.3. Frontend Design

Semantic HTML5 elements were employed to improve Search Engine Optimisation (SEO) and accessibility (Duckett, 2011). The homepage utilised header, nav, main, footer, section, and article elements to logically organise content. Product pages employed div and figure elements to display product images, descriptions, prices, and reviews. User authentication forms used form, input, and button elements to capture registration and login data.

CSS was applied to guarantee responsive design and visual appeal. Grid and flexbox layouts were used for responsive structuring, while media queries ensured automatic layout adjustment for desktops, tablets, and mobile devices (Cederholm, 2012). A SCSS-based custom stylesheet was created importing Bootstrap functions, with a defined colour palette including deep purple as the primary colour, azure blue as secondary, and tangerine orange as accent. Bootstrap provided pre-designed components that accelerated development and ensured design consistency (Spurlock, 2013).

3.4. Backend Development

The server-side application was implemented using Node.js and Express.js. Express.js provided a minimal web framework for handling routing, middleware, and HTTP requests (Brown, 2019). EJS (Embedded JavaScript) templates were used to render dynamic HTML from the server, allowing seamless integration of frontend and backend logic. MongoDB, accessed via the Mongoose ODM, stored user schemas with fields for name, email, password, and role (admin or customer). RESTful API endpoints were developed for user authentication, product management, and order processing. Passport.js with bcrypt was used for secure local authentication, implementing password hashing and session management.

3.5. Payment Integration

Stripe was selected as the payment gateway due to its security, ease of integration, comprehensive feature set, global reach, and PCI-DSS compliance. The integration process involved creating a Stripe account, obtaining API keys, adding the Stripe JavaScript library, and operating in test mode for the purposes of this study. The Stripe checkout page collected billing, shipping, and payment card information directly, reducing the website's exposure to sensitive financial data and conforming to industry best practices (Sterling, 2017).

3.6. Testing

Three levels of testing were conducted. Unit testing verified individual components and functions in isolation. Integration testing assessed the complete Stripe payment flow, confirming end-to-end transactions functioned correctly. User acceptance testing was performed with multiple registered users to confirm that the interface was responsive and that core workflows including registration, browsing, cart management, and checkout operated as expected. Debugging was performed iteratively throughout development, with Stripe's official documentation consulted to resolve integration issues.

4. Results

4.1. Website Functionality

The implementation successfully delivered a fully functional e-commerce platform. Users are required to register and accept Terms and Conditions before making purchases, after which they can login, browse products, and manage their accounts. The product browsing interface allows users to search for items, view product details including images, descriptions, and prices, and filter products by category. The dynamic shopping cart enables users to add products, adjust quantities using Add and Subtract buttons, remove items entirely, view a running total, and proceed to checkout. The About Us and Contact Us pages provide mission and vision information and a message form respectively.

4.2. User Authentication

User registration and login were implemented using Passport.js with a local strategy. Bcrypt was used to hash passwords before storage in MongoDB. The user schema defined fields for name (String, required), email (String, required, unique), password (String, required), and role (enum: admin/customer, default: customer). Testing with three distinct user profiles confirmed that personalised welcome messages were displayed correctly, role-based access functioned as expected, and session persistence operated reliably.

4.3. Payment Processing

Integration with Stripe for payment processing was completed successfully. Payments were processed securely through Stripe's platform in test mode, with test card number 4242 4242 4242 4242 used to simulate transactions. The checkout interface displayed itemised order totals, collected full name, shipping address, and email, and presented payment details fields consistent with Stripe's hosted elements. A successful test transaction of £535.00 was completed, confirming that the end-to-end payment flow operated correctly. The Stripe platform also collected shipping information including address, city, postcode, and country.

4.4. Database Management

MongoDB Compass was used to inspect the database state. The chuka-ecom database contained three collections: orders (storage size 20.48 kB, 4 documents), products (storage size 20.48 kB, 13 documents), and users (storage size 20.48 kB, 6 documents). This confirmed that the database correctly persisted product catalogue entries, user registrations, and completed orders, providing a functional foundation for inventory and order management.

4.5. Performance and Usability

The website performed optimally with responsive design and fast load times across tested devices. User feedback collected during user acceptance testing indicated that the platform was easy to navigate and that the checkout process was seamless and straightforward. Responsive CSS media queries ensured the interface adapted correctly to desktop, tablet, and mobile screen sizes.

5. Discussion

5.1. Technical Architecture

The adoption of a unified JavaScript stack across frontend (HTML/CSS/JS/EJS) and backend (Node.js/Express.js) reduced the cognitive overhead associated with context-switching between languages and facilitated faster iteration (Holzner, 2012). MongoDB's document-oriented schema provided the flexibility to accommodate the varied data structures inherent in product catalogues and user records without the rigidity of relational schemas (Chodorow, 2013). The non-blocking, event-driven nature of Node.js ensured that the server could handle concurrent requests efficiently, a characteristic particularly valuable in e-commerce contexts where traffic spikes are common (Elliott, 2016).

5.2. Payment Security

Delegating payment processing to Stripe rather than building a custom payment system was a pragmatic decision that offered several advantages. Stripe's PCI-DSS compliance and mature security infrastructure eliminated the need to handle raw card data directly, reducing the platform's risk surface considerably (Schneider, 2017). The test mode implementation allowed thorough validation of the checkout flow without exposing live financial data. Future deployment would require switching to live Stripe API keys and ensuring HTTPS is enforced across all pages.

5.3. Responsive Design Effectiveness

The use of Bootstrap alongside custom SCSS variables provided a consistent and visually coherent interface that adapted effectively across device types. CSS media queries allowed breakpoints to be defined for mobile, tablet, and desktop viewports, addressing the growing proportion of e-commerce transactions conducted on mobile devices (Kourouthanassis & Giaglis, 2012). While desktop e-commerce platforms offer advantages including larger screen real estate, easier navigation, better processing power, and enhanced content management tools (Chaffey, Hemphill & Edmundson-Bird, 2019), the responsive approach ensured that mobile users were not disadvantaged.

5.4. Challenges Encountered

Three principal challenges were encountered during development. First, Stripe integration presented configuration difficulties that were resolved through systematic consultation of Stripe's official documentation. Second, achieving consistent responsive behaviour across all device types required extensive testing with CSS media queries. Third, selecting an appropriate database platform was initially challenging; ultimately, MongoDB's flexibility and scalability made it the preferred choice for managing diverse product catalogues and user data (Chodorow, 2013). These challenges reflect the typical complexity of full-stack web application development and underscore the value of agile iterative approaches that allow problems to be identified and addressed progressively.

5.5. Comparison with Existing Literature

The findings of this study align with the broader trajectory identified in the literature, wherein modern e-commerce platforms increasingly rely on JavaScript-based full stacks, cloud databases, and third-party payment APIs to deliver scalable and secure shopping experiences (Chaffey, 2020). The successful integration of MongoDB as a NoSQL backend corroborates Chodorow's (2013) assessment of its suitability for e-commerce applications. The adoption of Stripe as a payment gateway reflects the wider trend toward delegating sensitive financial processing to specialised, PCI-DSS-compliant third parties rather than managing it internally (Sterling, 2017).

6. Conclusion

This paper has presented the design and development of a fully functional full-stack e-commerce web application built using HTML, CSS, JavaScript, Node.js, Express.js, MongoDB, and Stripe. The platform successfully delivers the core requirements of a modern e-commerce system, including secure user registration and authentication, a comprehensive product browsing and filtering interface, dynamic shopping cart management, and a Stripe-powered checkout process with real-time billing and shipping collection. Responsive design was achieved through the combined use of Bootstrap and CSS media queries, ensuring a consistent user experience across desktop, tablet, and mobile devices.

The iterative, agile-informed development approach enabled challenges related to payment integration, responsive design, and database management to be identified and resolved progressively. The unified JavaScript stack across frontend and backend reduced development complexity while MongoDB's flexible document model provided an effective data management foundation. The study demonstrates that a competitive, secure, and scalable e-commerce platform can be constructed from scratch using open-source web technologies without the constraints inherent in off-the-shelf solutions.

Recommendations

Based on the outcomes and limitations of this study, the following recommendations are made for future development:

- **Hybrid Mobile Application:** A hybrid web application using Ionic or Capacitor should be developed to wrap the existing codebase into a native container for iOS and Android, thereby attracting the growing segment of mobile commerce users.
- **Two-Factor and Biometric Authentication:** Two-factor authentication (2FA) and biometric login options should be added to enhance account security and reduce the risk of unauthorised access, particularly for the admin profile (Chui, Manyika & Miremadi, 2017).
- **AI-Powered Chatbot:** An AI-powered chatbot should be integrated for round-the-clock customer support, handling common queries, order tracking, and product inquiries, thereby reducing the burden on human agents (Zhao et al., 2019).
- **Advanced Analytics:** Sophisticated analytics tools should be implemented to monitor website traffic, sales performance, and customer behaviour, providing data to inform marketing strategies and inventory control (Kumar et al., 2019).
- **Multi-Language and Multi-Currency Support:** Expanding the platform to support multiple languages and currencies would enable entry into international markets and accommodate a broader customer base.
- **Social Media Integration:** Integrating social media sharing and advertising features for platforms such as Instagram and Facebook would increase product visibility and drive additional traffic to the website.
- **GDPR and CCPA Compliance:** A formal compliance review should be conducted before public deployment to ensure full adherence to the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), including explicit consent mechanisms and data deletion capabilities (Zhu & Jiang, 2021).

References

- [1] Atzori, L., Iera, A. and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), pp. 2787-2805.
- [2] Berners-Lee, T. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. New York: HarperCollins.
- [3] Bos, B., Celik, T., Hickson, I. and Lie, H.W. (2011). *Cascading Style Sheets Level 3 (CSS3) Specification*. Available at: <https://www.w3.org/TR/css-2011> (Accessed 20 July 2024).
- [4] Bos, B., Lie, H.W., Lilley, C. and Jacobs, I. (1998). *Cascading Style Sheets, Level 2 (CSS2) Specification*. Available at: <https://www.w3.org/TR/1998/REC-CSS2-19980512> (Accessed 23 August 2024).
- [5] Brandt, R. (2011). *One Click: Jeff Bezos and the Rise of Amazon.com*. New York: Penguin Group.
- [6] Brown, E. (2019). *Web Development with Node and Express: Leveraging the JavaScript Stack*. 2nd ed. Sebastopol: O'Reilly Media.
- [7] Byron, A. and Sanderson, B. (2017). *Learning GraphQL: Declarative Data Fetching for Modern Web Apps*. Sebastopol: O'Reilly Media.
- [8] Casciaro, M. and Mammino, L. (2016). *Node.js Design Patterns*. 2nd ed. Birmingham: Packt Publishing.

- [9] Casino, F., Dasaklis, T.K. and Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36, pp. 55-81.
- [10] Cassidy, J. (2002). *Dot.com: The Greatest Story Ever Sold*. New York: HarperCollins.
- [11] Catalini, C. and Gans, J.S. (2016). Some Simple Economics of the Blockchain. MIT Sloan Research Paper No. 5191-16. Available at SSRN: <https://ssrn.com/abstract=2874598> (Accessed 20 August 2024).
- [12] Cederholm, D. (2012). *CSS3 For Web Designers*. New York: A Book Apart.
- [13] Chaffey, D. (2007). *E-Business and E-Commerce Management: Strategy, Implementation and Practice*. 3rd ed. Harlow: Pearson Education.
- [14] Chaffey, D. (2015). *Digital Business and E-commerce Management*. 6th ed. Harlow: Pearson Education.
- [15] Chaffey, D. (2020). *Digital Business and E-commerce Management*. 7th ed. London: Pearson.
- [16] Chaffey, D., Hemphill, T. and Edmundson-Bird, D. (2019). *Digital Business and E-commerce Management*. 7th ed. Harlow: Pearson.
- [17] Chaffey, D. and Ellis-Chadwick, F. (2019). *Digital Marketing: Strategy, Implementation and Practice*. 7th ed. Harlow: Pearson.
- [18] Chodorow, K. (2013). *MongoDB: The Definitive Guide*. 2nd ed. Sebastopol: O'Reilly Media.
- [19] Chukwudi George-Linus Onyekwere, Otuu Obinna Ogbonnia and Stanley Muturi Githinji. Beyond benchmark accuracy: Evaluating deepfake detection tools for digital forensic admissibility through a systematic review. *World Journal of Advanced Research and Reviews*, 2026, 30(02), 1466-1477. Article DOI: <https://doi.org/10.30574/wjarr.2026.30.2.1387>
- [20] Cohen, A. (2002). *The Perfect Store: Inside eBay*. New York: Little, Brown and Company.
- [21] Coursaris, C. and Hassanein, K. (2002). Understanding m-commerce: A consumer-centric model. *Quarterly Journal of Electronic Commerce*, 3(3), pp. 247-271.
- [22] Duckett, J. (2011). *HTML & CSS: Design and Build Websites*. New York: Wiley.
- [23] Ellison, G. and Ellison, S.F. (2005). Lessons about Markets from the Internet. *Journal of Economic Perspectives*, 19(2), pp. 139-158.
- [24] Elliott, I. (2016). *Just Express*. 3rd ed. I/O Press.
- [25] Facebook (2021). React - A JavaScript library for building user interfaces. Available at: <https://reactjs.org> (Accessed 10 July 2024).
- [26] Flanagan, D. (2011). *JavaScript: The Definitive Guide*. 6th ed. Sebastopol: O'Reilly Media.
- [27] Fling, B. (2009). *Mobile Design and Development*. Sebastopol: O'Reilly Media.
- [28] Google (2021). Angular - One framework. Mobile & desktop. Available at: <https://angular.io> (Accessed 25 August 2024).
- [29] Grossman, J. (2011). *The Tangled Web: A Guide to Securing Modern Web Applications*. Indianapolis: Wiley.
- [30] Hickson, I. and Hyatt, D. (2014). HTML5 Specification. Available at: <https://www.w3.org/TR/html5> (Accessed 10 August 2024).
- [31] Hilken, T., de Ruyter, K., Chylinski, M., Mahr, D. and Keeling, D.I. (2017). Augmenting the eye of the beholder: exploring the strategic potential of augmented reality to enhance online service experiences. *Journal of the Academy of Marketing Science*, 45(6), pp. 884-905.
- [32] Holzner, S. (2012). *Node.js: The Right Way: Practical, Server-Side JavaScript That Scales*. Dallas: The Pragmatic Programmers.
- [33] Huang, Y., Huang, S. and Liu, Y. (2013). Web Performance Testing and Analysis for E-Commerce. *International Journal of Software Engineering and Knowledge Engineering*, 23(1), pp. 15-28.
- [34] Kantor, J. and Streitfeld, D. (2015). Inside Amazon: Wrestling Big Ideas in a Bruising Workplace. *New York Times*. Available at: <https://www.nytimes.com/2015/08/16/technology/inside-amazon> (Accessed 3 August 2024).
- [35] Kaplan, D. (2002). *eBay: The World's Online Marketplace*. London: Reuters.

- [36] Khurana, A. (2020). The History and Evolution of Ecommerce. Available at: <https://www.thebalancesmb.com/the-history-and-evolution-of-ecommerce-1141609> (Accessed 20 July 2024).
- [37] Kim, J. and Forsythe, S. (2008). Adoption of Virtual Try-on technology for online apparel shopping. *Journal of Interactive Marketing*, 22(2), pp. 45-59.
- [38] Kotha, S. (1998). Competing on the Internet: The Case of Amazon.com. *European Management Journal*, 16(2), pp. 212-222.
- [39] Kourouthanassis, P. and Giaglis, G. (2012). Introduction to the special issue mobile commerce: the past, present, and future of mobile commerce research. *International Journal of Electronic Commerce*, 16(4), pp. 5-17.
- [40] Krug, S. (2014). *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. 3rd ed. Berkeley: New Riders.
- [41] Kumar, V., Rajan, B., Venkatesan, R. and Lecinski, J. (2019). Understanding the Role of Artificial Intelligence in Personalized Engagement Marketing. *California Management Review*, 61(4), pp. 135-155.
- [42] Laudon, K. and Traver, C. (2018). *E-commerce 2018: Business, Technology, Society*. 14th ed. Harlow: Pearson.
- [43] McFarland, D. (2008). *JavaScript & jQuery: The Missing Manual*. Sebastopol: O'Reilly Media.
- [44] Meyer, E. (2007). *Cascading Style Sheets: The Definitive Guide*. 3rd ed. Sebastopol: O'Reilly Media.
- [45] Morris, C.R. and Alam, P. (2001). *Money, Greed, and Risk: Why Financial Crises and Crashes Happen*. New York: Times Books.
- [46] Ngai, E.W.T. and Gunasekaran, A. (2007). A review for mobile commerce research and applications. *Decision Support Systems*, 43(1), pp. 3-15.
- [47] Pavlou, P.A. (2011). State of the information privacy literature: Where are we now and where should we go? *MIS Quarterly*, 35(4), pp. 977-988.
- [48] Resig, J. and Bibeault, B. (2013). *jQuery in Action*. 3rd ed. Shelter Island: Manning Publications.
- [49] Russell, A. (2015). *Progressive Web Apps: Building Modern Web Apps*. Available at: <https://developers.google.com/web/progressive-web-apps> (Accessed 29 July 2024).
- [50] Schneider, G.P. (2017). *Electronic Commerce*. 12th ed. Boston: Cengage Learning.
- [51] Schwaber, K. and Sutherland, J. (2020). *The Scrum Guide*. Available at: <https://scrumguides.org> (Accessed 15 August 2024).
- [52] Smith, B. and Linden, G. (2017). Two decades of recommender systems at Amazon.com. *IEEE Internet Computing*, 21(3), pp. 12-18.
- [53] Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. Sebastopol: O'Reilly Media.
- [54] Sterling, J. (2017). *JavaScript and Payment Gateways: A Practical Approach*. Available at: <https://www.paypal-engineer.com/articles/2017/javascript-and-payment-gateways> (Accessed 2 September 2024).
- [55] Stone, B. (2013). *The Everything Store: Jeff Bezos and the Age of Amazon*. New York: Little, Brown and Company.
- [56] Tapscott, D. and Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. New York: Portfolio/Penguin.
- [57] Tilkov, S. and Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 14(6), pp. 80-83.
- [58] Turban, E., King, D., Lee, J., Liang, T. and Turban, D. (2015). *Electronic Commerce: A Managerial and Social Networks Perspective*. 8th ed. Cham: Springer.
- [59] Wadhwa, V. (2018). *Progressive Web Apps: Unlocking the Next Wave of Growth for Flipkart*. Available at: <https://developers.google.com/web/showcase/2016/flipkart> (Accessed 25 July 2024).
- [60] World Bank (2021). *World Development Report 2021: Data for Better Lives*. Washington DC: The World Bank.
- [61] Zhao, R., Liu, J., Zhang, X., Lyu, M.R. and King, I. (2019). Improving customer service in e-commerce with AI chatbots. *IEEE Transactions on Knowledge and Data Engineering*, 31(5), pp. 876-889.
- [62] Zhu, X. and Jiang, H. (2021). Data Privacy and Security Issues in E-commerce. *Journal of Global Information Technology Management*, 24(1), pp. 1-9.