# E-COMMERCE WEBSITE FOR UNIVERSITY STUDENTS

# (*A case study of the University of Ilorin*)

# SUBMITTED BY:

# GROUP 5 UNIVERSITY OF ILORIN, ILORIN, KWARA COMPUTER SCIENCE

# CSC 420 – SOFTWARE ENGINEERING JUNE, 2024

# SUBMITTED TO:

# DR. S.A. SALIHU DEPARTMENT OF COMUTER SCIENCE, FACULTY OF COMMUNICATION AND INFORMATION SCIENCES

## GROUP 5 MEMBERS

1. Jimoh Azeez Kayode 19/52HA069

2. Jide-Kazeem Abdulmalik Oladimeji 19/52HA068

3. Ezirim Kizito Chinemerem - 19/52HA057

4. ⁠Gidado ABDULLATEEF adebola 19/52HA059

5. ⁠Ilori Peter Oluwasegun 19/52HA066

6. ⁠Ibrahim Muhammed Babatunde 19/52HA062

7. ⁠Ibitomihi Eniola Banji 19/52HA061

8. Hassan Abubakar 19/52HA060

9. Ikirodah Michael O. 19/52HA065

10. Elodimuor Favour 19/52HA056

11. Evopa Victor Nero 19/52HA058

12. Komolafe Victor Johnbosco 19/52HA0

13. Issa Mariam 19/52HA067

## ABSTRACT

This project involves the development of an e-commerce website exclusively for university students, aiming to provide a user-friendly platform for purchasing various products online. The website features user authentication, product categorization, and a secure checkout process, ensuring a seamless and secure shopping experience. The project employed modern web technologies and followed best practices in web development and security.

TABLE OF CONTENTS

[TITLE PAGE 1](#_Toc169742715)

[GROUP 5 MEMBERS 2](#_Toc169742716)

[ABSTRACT 3](#_Toc169742717)

[TABLE OF CONTENTS 4](#_Toc169742718)

[I. INTRODUCTION 5](#_Toc169742719)

[II. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) 9](#_Toc169742720)

[III. FEASIBILITY STUDY 10](#_Toc169742721)

[IV. REQUIREMENTS ANALYSIS AND SPECIFICATION 13](#_Toc169742722)

[V. DESIGN 16](#_Toc169742723)

[VI. CODING 25](#_Toc169742724)

[VII. TESTING 30](#_Toc169742725)

[VIII. MAINTENANCE 36](#_Toc169742726)

[IX. CONCLUSION 39](#_Toc169742727)

## ****I. INTRODUCTION****

Since the Stone Age, technological advancements have consistently aimed to simplify human life, particularly within the business sector. From the creation of numerical systems and the abacus to textile machines and the advent of computers, technology has been pivotal in transforming business operations. The invention of the internet and the World Wide Web has further revolutionized communication and collaboration, transcending geographical barriers. This connectivity gave rise to E-commerce, which involves buying and selling goods and services online.

E-commerce software is now being developed globally to meet the diverse needs of various organizations. This report details Group 5's process in creating an E-commerce website designed for the University of Ilorin's staff and students, aiming to simplify their daily activities through technology. The software allows users to log in with their school ID, browse a catalog of goods, add items to their cart, make payments, and await delivery.

**Objective**

This E-commerce website aims to bridge the gap between users and the products they wish to purchase, providing a user-friendly platform with a wide range of relevant goods. It incorporates a seamless payment system, ensuring smooth transactions. By utilizing efficient algorithms and low-storage databases, the software enhances user experience through improved speed and efficiency. The primary objective is to facilitate online purchases, eliminating the need for physical visits to the marketplace.

**Simplicity and Ease Of Use**

The website emphasizes simplicity to help users easily navigate a large database of goods, ensuring they can quickly find what they need.

**Scalability**

While fully functional, the website's design and architecture are flexible and scalable, allowing for future enhancements and new features to further improve user experience.

**Payment Platform**

An integrated payment platform streamlines transactions, reducing user stress and enabling swift completion of purchases within a single website.

**Development Approach**

The development of the e-commerce website for university students followed a structured and systematic approach to ensure the project was completed efficiently and effectively. The approach can be divided into several key phases: Planning, Design, Development, Testing, and Deployment. The Agile methodology was chosen to manage the project, allowing for iterative development and continuous feedback.

**Planning Phase**

The planning phase involved understanding the requirements and objectives of the project. This phase included:

1. Requirement Gathering: Conducting meetings with group members to gather and document the requirements. This included understanding the needs of university students and the desired features of the e-commerce website.
2. Project Scope Definition: Defining the scope of the project to ensure that the development efforts are aligned with the objectives. This included listing the features to be implemented, such as user authentication, product categories, and the checkout process.
3. Feasibility Study: Assessing the technical feasibility of the project, including the selection of technologies, tools, and frameworks to be used.

**Design Phase**

The design phase focused on creating a blueprint for the website. This phase included:

1. System Architecture Design: Designing the overall architecture of the system, including the frontend, backend, and database components. This involved creating diagrams to represent the flow of data and the interactions between different components.
2. User Interface (UI) Design: Creating wireframes and prototypes for the website's user interface. Tools like Adobe XD and Figma were used to design the UI, ensuring it was user-friendly and visually appealing.
3. Database Design: Designing the database schema to store user data, product information, and order details. PostgreSQL was chosen for its flexibility and scalability.

**Development Phase**

The development phase was iterative, with the project divided into several sprints. Each sprint focused on developing specific features and functionalities. This phase included:

1. Frontend Development: Using HTML, CSS, and JavaScript to create the user interface.
2. Backend Development: Developing the server-side logic using Django. This included implementing user authentication, product management, and order processing.
3. Database Integration: Integrating PostgreSQL to handle data storage and retrieval. Django model was used as an ORM to interact with the database.
4. Feature Implementation: Developing the core features of the website, such as user login, product browsing and search, shopping cart management, and the checkout process.

**Testing Phase**

The testing phase involved validating the functionality, performance, and security of the website. This phase included:

1. Unit Testing: Writing and executing tests for individual components to ensure they function correctly in isolation. Jest was used as the testing framework.
2. Integration Testing: Testing the interactions between different components to ensure they work together seamlessly. This included testing the endpoints and database interactions.
3. User Acceptance Testing (UAT): Conducting testing sessions with a group of university students to gather feedback and identify any usability issues. This feedback was used to make necessary adjustments and improvements.
4. Security Testing: Performing security assessments to identify and mitigate potential vulnerabilities. This included testing for SQL injection, cross-site scripting (XSS), and other common security threats.

**Deployment Phase**

The final phase involved deploying the website to a production environment. This phase included:

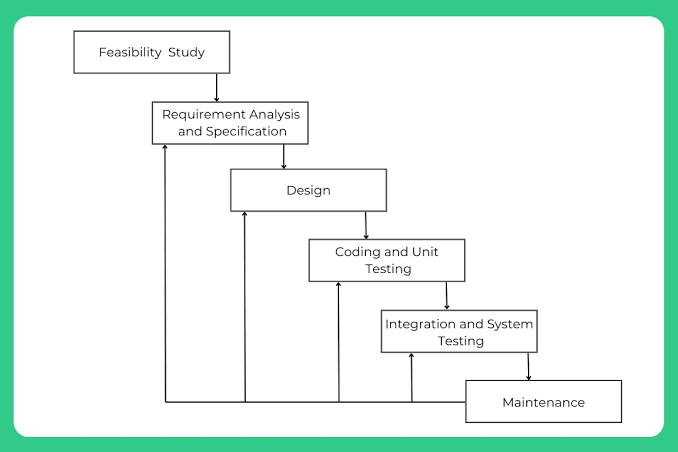
1. Hosting Setup: Choosing a hosting provider and setting up the server environment. Render was selected for its ease of use and integration with Django applications.
2. Continuous Integration/Continuous Deployment (CI/CD): Setting up CI/CD pipelines to automate the deployment process. This ensured that any new changes were automatically tested and deployed to the production server.
3. Monitoring and Maintenance: Implementing monitoring tools to track the performance and availability of the website. Regular maintenance tasks were scheduled to ensure the website remained up-to-date and secure.

By following this structured development approach, the e-commerce website was developed efficiently, meeting the project requirements and ensuring a high level of quality and user satisfaction.

## II. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

The Software Development Life Cycle (SDLC) is a systematic approach to developing software, encompassing design, creation, testing, deployment, and maintenance. It ensures software quality, meets user needs, and is delivered on time and within budget.

The SDLC phases include feasibility study, requirements analysis, design, coding, testing, and maintenance. In this project, we used the Iterative Waterfall/Incremental Model, which allows for adaptability and flexibility in response to changing requirements. We employed this iterative model to develop an E-commerce Website and its application is described in detail below.



**Figure 1: Iterative waterfall model**

## **III. FEASIBILITY STUDY**

The website aims to provide a platform for students and staff to buy goods and services within the school community. This study evaluates the technical, economic, operational, and schedule feasibility of the project.

**Technical Feasibility**

The technical feasibility study assessed the feasibility of implementing this project using existing technologies and resources. The team evaluated the current technical capabilities and the resources at our disposal comprehensively.

1. Technical Requirements: The technical requirements for building the e-commerce website encompass a variety of essential technologies. For the front-end development, we use HTML, CSS, and JavaScript to create a user-friendly and visually appealing interface. On the backend, we make use of Django, which handles server-side logic and operations. To manage data storage efficiently, we can choose the database, PostgreSQL. Additionally, security is paramount, necessitating the implementation of SSL certificates to secure data transmission, robust user authentication mechanisms to protect user accounts, and data encryption to safeguard sensitive information.
2. Resources: The resources required for building the e-commerce website include both development tools and human resources. Essential development tools comprise integrated development environments (IDEs) such as PyCharm, which facilitates efficient coding and debugging, and version control systems like Git, which enable collaborative work and track changes across the development team. On the human resources side, the project will rely on a team of skilled web developers who write and maintain the code, and designers who create the visual aspects and user experience of the website.
3. Evaluation: The evaluation of the project's feasibility indicates that the required technology and tools, such as IDEs, version control systems, and programming languages, are readily available to the team. Additionally, the team has access to skilled personnel, including web developers and designers, who possess the necessary expertise to carry out the project.

In conclusion, the project is technically feasible given the available resources and expertise. The team is well-equipped with the necessary technology and tools, including modern integrated development environments (IDEs), version control systems, and the programming languages and frameworks for both frontend and backend development. Additionally, the team comprises experienced web developers and creative designers, all of whom bring valuable expertise to the project. Moreover, the institution is filled with a pool of skilled personnel and vast majority of the school populace is familiar with the use of E-commerce web application. This combination of technological resources, expert knowledge, and enthusiastic manpower ensures that the technical challenges of building an e-commerce website can be effectively met, paving the way for a successful project execution.

**Economic Feasibility**

The Economic Feasibility determines the cost-effectiveness of building the e-commerce website by evaluating both the initial and ongoing costs associated with development, hosting, and maintenance, and comparing these expenses. This aim is to determine if the financial investment required is justified by the potential benefits and returns.

1. Development Cost: The cost assessment of the project centered on utilizing free and low-cost tools. The initial development primarily rely on open-source software and platforms, such as free integrated development environments (IDEs) like PyCharm, and version control systems like Git, which do not incur any costs. Hosting expenses is kept minimal by opting for budget-friendly options, ranging from ₦15,000 to ₦25,000 per month. By focusing on these cost-saving measures, the project aims to remain economically viable while providing a valuable E-commerce store.
2. Revenue Potential: The revenue potential for the e-commerce website includes generating income through transaction fees and advertising. By charging a small percentage of each sale, the website can create a steady stream of revenue that contributes to its operational costs and sustainability. Additionally, the website can offer advertising space for local businesses or school events, providing another source of income.
3. Evaluation: The evaluation of the project's economic feasibility reveals that it has low initial and operational costs, making it an attractive venture for a school project. By leveraging free and low-cost tools for development and maintenance, such as open-source software, budget-friendly hosting services, and minimal domain registration fees, the project minimizes its financial burden. Additionally, the potential to generate moderate revenue through small transaction fees on each sale and advertising space for local businesses or school events adds to the project's financial viability.

In conclusion, the project is economically feasible with a low financial barrier and potential for self-sustainability. The combination of low costs and potential revenue streams indicates that the project is economically sound and capable of sustaining itself over time.

## IV. REQUIREMENTS ANALYSIS AND SPECIFICATION

The requirements analysis and specification phase is critical to the success of any software development project. It involves understanding the needs and constraints of the stakeholders and documenting the functional and non-functional requirements of the system. For the e-commerce website tailored for university students, this phase ensured that the developed solution met the users' needs and adhered to the project's objectives.

**Stakeholder Analysis**

The primary stakeholders for this project included:

1. University Students: The end-users who would use the website to browse, select, and purchase products.
2. University Administration: Ensuring that only verified students have access to the platform.
3. Project Supervisors and Instructors: Overseeing the project development and ensuring it meets academic standards.
4. Development Team: Responsible for designing, developing, and maintaining the website.

**Functional Requirements**

Functional requirements define the specific behavior or functions of the system. For this project, the key functional requirements included:

1. User Registration and Authentication:
   1. Only university students should be able to register using their university-provided email addresses.
   2. Users should be able to log in securely with a username and password.
   3. Passwords must be hashed and stored securely in the database.
2. Product Catalog:
   1. The website should display a catalog of products organized into categories.
   2. Users should be able to search for products by name, category, and other attributes.
   3. Each product listing should include details such as price, description, and availability.
3. Shopping Cart:
   1. Users should be able to add products to a shopping cart.
   2. Users should be able to view and manage the contents of their cart, including updating quantities or removing items.
4. Checkout Process:
   1. Users should be able to proceed to checkout, review their order, and provide shipping and payment information.
   2. The system should support secure payment processing via third-party payment gateways.
   3. Users should receive a confirmation email upon successful purchase.
5. Order Management:
   1. Users should be able to view their order history and track the status of their orders.
   2. Administrators should be able to manage and update the status of orders.

**Non-Functional Requirements**

Non-functional requirements define the system's operational attributes, such as performance, usability, and security. For this project, the key non-functional requirements included:

1. Performance:
   1. The website should load within 3 seconds on average for a standard user with a stable internet connection.
   2. The system should be able to handle up to 100 concurrent users without significant performance degradation.
2. Usability:
   1. The user interface should be intuitive and easy to navigate, catering to users with varying levels of technical expertise.
   2. The website should be fully responsive, providing a seamless experience across different devices and screen sizes.
3. Security:
   1. All user data, especially sensitive information such as passwords and payment details, must be securely encrypted.
   2. The system should be protected against common web vulnerabilities, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
4. Scalability:
   1. The system architecture should support future growth, allowing for the addition of new features and an increasing number of users without requiring significant rework.
5. Reliability:
   1. The website should have an uptime of at least 99.5%, ensuring it is available to users most of the time.
   2. Regular backups should be implemented to prevent data loss in case of system failures.

**Requirement Specifications**

The detailed specifications derived from the analysis included:

1. User Stories: Detailed descriptions of user interactions with the system, such as "As a student, I want to be able to search for products by category so that I can find what I need quickly."
2. Use Case Diagrams: Visual representations of the interactions between users and the system, highlighting the main functionalities.
3. Data Flow Diagrams (DFDs): Illustrations of how data moves through the system, showing inputs, outputs, and storage points.
4. Entity-Relationship Diagrams (ERDs): Schematics showing the relationships between different data entities in the database.

## V. DESIGN

The design process of an e-commerce website tailored for students and their lecturers involves multiple phases to ensure the final product is intuitive, user-friendly, and aligns with the specific needs of the target audience.

Iterative prototyping is crucial for refining the website's design. This approach involves continuous feedback and improvements to ensure the website meets user needs and expectations.

1. Initial Prototyping: The design process begins with creating initial prototypes, including wireframes and low-fidelity mock-ups. These prototypes help visualize the layout, basic functionality, and user flow of the e-commerce website. Key features for students and lecturers, such as course material purchase, discounted student deals, and lecturer-specific resources, are highlighted.
2. Internal Feedback Sessions: Regular internal feedback sessions are conducted, where team members review the prototypes and provide insights. These sessions foster open discussions and brainstorming, enabling the identification of areas for improvement and potential enhancements specific to the needs of students and lecturers.
3. Usability Testing: Usability testing is conducted with team members simulating real-life scenarios. This testing assesses how effectively the website serves its intended users, highlighting any usability issues. Scenarios include students purchasing textbooks and lecturers accessing teaching resources.
4. Gathering User Feedback: Feedback from potential end-users, including students and lecturers, is collected through informal discussions and surveys. This feedback provides valuable perspectives on the website’s functionality, usability, and overall user experience.
5. Incorporating Feedback: Based on the feedback from internal team members and potential end-users, iterative improvements are made to the website’s user interface. Changes range from minor adjustments, like button placement, to major redesigns, such as overhauling the navigation structure to better suit academic workflows.
6. Validation and Testing: After each round of improvements, the changes are validated through further usability testing and feedback sessions. This ensures that the refinements effectively enhance user experience and align with user expectations, particularly focusing on the ease of use for students and lecturers.
7. Continuous Refinement: The iterative prototyping process is repeated multiple times, with each iteration bringing the website’s design closer to its final form. Continuous refinement incorporates insights from feedback sessions and usability testing, progressively improving the design.
8. Final Design Selection: By the end of the iterative prototyping process, the project team collectively selects the final user interface design. This design encompasses the most effective and user-friendly elements, ensuring usability, visual appeal, and alignment with project requirements tailored for students and lecturers.

**Key Considerations**

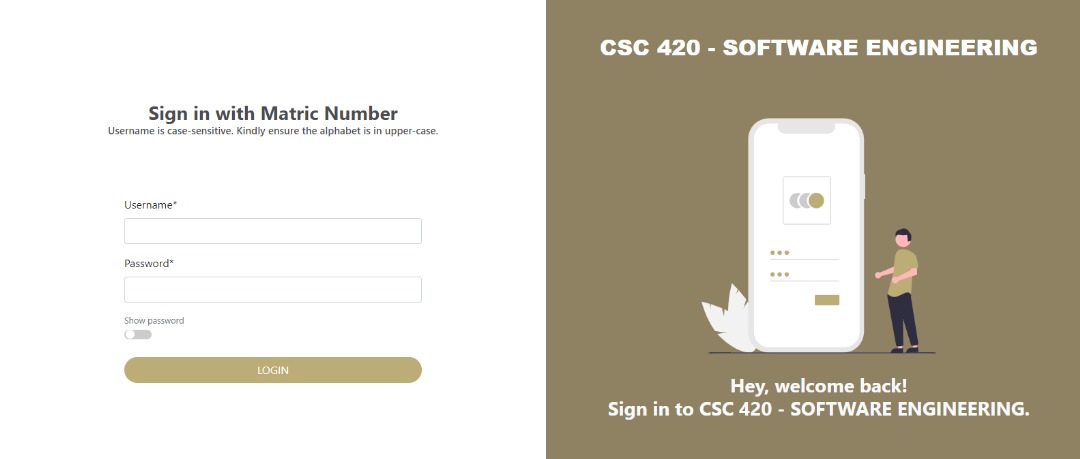
Throughout the design process, the following considerations are crucial:

1. User-Centric Design: Ensuring the website is intuitive and meets the specific needs of students and lecturers.
2. Accessibility: Incorporating features that make the website accessible to users with disabilities.
3. Mobile Responsiveness: Ensuring the website functions well on mobile devices, as students often access resources on-the-go.
4. Security and Privacy: Implementing robust security measures to protect user data, particularly sensitive academic and financial information.

The design process of the e-commerce website focuses on creating a user-centered and feature-rich platform tailored for students and lecturers. The design team prioritized simplicity and clarity to ensure users can easily navigate and understand the website. The following key features were integral to the design:

**Auth & Login: Wireframes and Mockups**

1. Initial sketches and wireframes were created to visualize the layout and flow of the authentication and login process. The design emphasized ease of use, allowing students and lecturers to quickly access their accounts with minimal friction.
2. The login page includes options for account creation, password recovery, and social media logins to cater to different user preferences.



**Figure 2: The login page**

**Cart: Interactive Cart Interface**

1. The cart interface was designed to be easily accessible and user-friendly. It allows users to view, edit, and manage their selected items before proceeding to checkout.
2. Features such as item quantity adjustment, product removal, and price updates were incorporated to provide a seamless shopping experience.

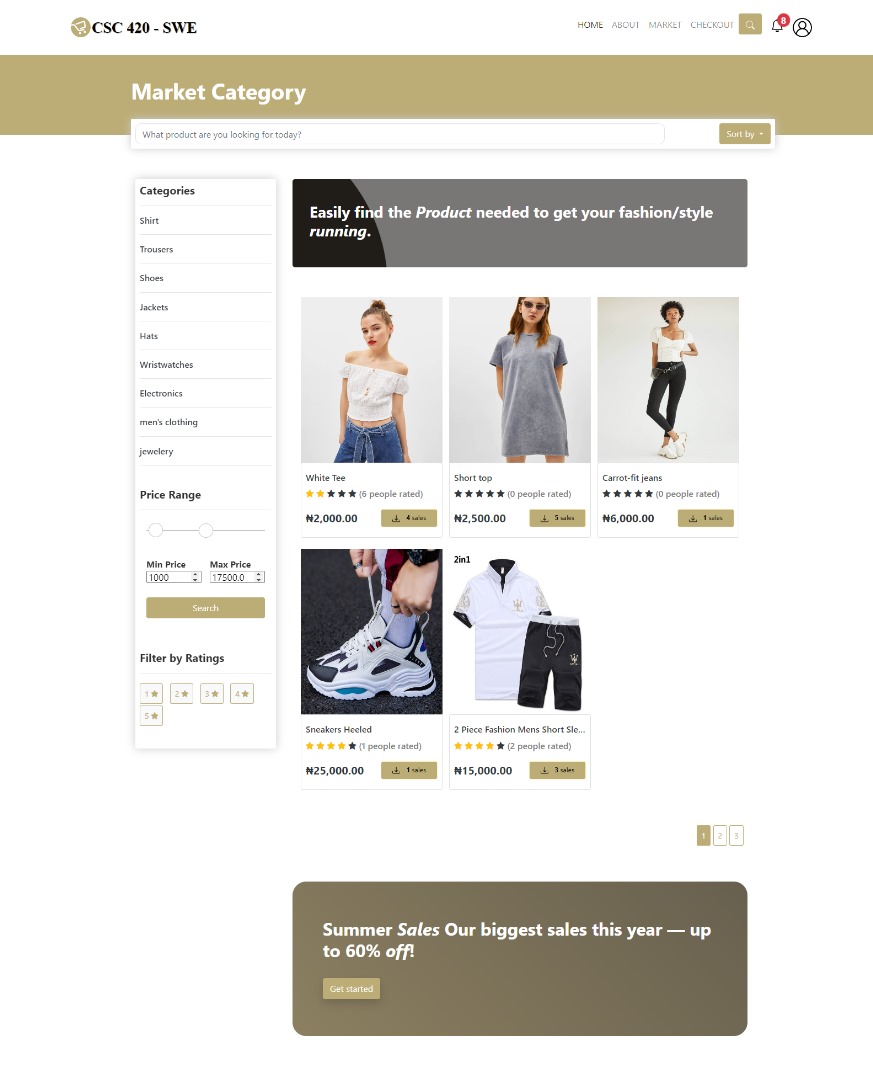
A screenshot of a checkout

Description automatically generated

**Figure 3: Cart and Checkout page**

**Shop Category Page: Category Navigation**

1. The shop category page was designed to organize products into intuitive categories, making it easier for users to find specific items. Categories include textbooks, digital resources, and classroom supplies.
2. Filters and sorting options were added to help users quickly locate products based on criteria like price, popularity, and relevance.



**Figure 4: Shop Category Page**

**Order Complete: Order Confirmation Page**

1. After completing a purchase, users are directed to an order confirmation page. This page provides details of the order, including a summary of purchased items, delivery information, and payment confirmation.
2. Clear next steps, such as tracking the order or continuing shopping, were included to enhance the user experience.

A screenshot of a website

Description automatically generated

**Figure 5: Order Confirmation Page**

**Offline Page: Offline Handling**

1. A custom offline page was designed to handle instances where users aren’t connected to the internet. The design maintains the website’s visual identity and provides helpful links on what to do next.

A screenshot of a computer

Description automatically generated

**Figure 6: Offline Handling Page**

**Payment: Secure Payment Gateway**

1. The payment page was designed with security and user convenience in mind. Multiple payment options, including credit cards, digital wallets, and student discounts, were integrated to accommodate various user preferences.
2. Clear instructions and a progress indicator were included to guide users through the payment process.

A screenshot of a computer

Description automatically generated

**Figure 7: Payment Page**

**Popup Cart: Quick Access Cart**

1. A popup cart feature allows users to quickly view and manage their selected items without navigating away from the current page. This feature enhances the shopping experience by providing immediate access to the cart summary.

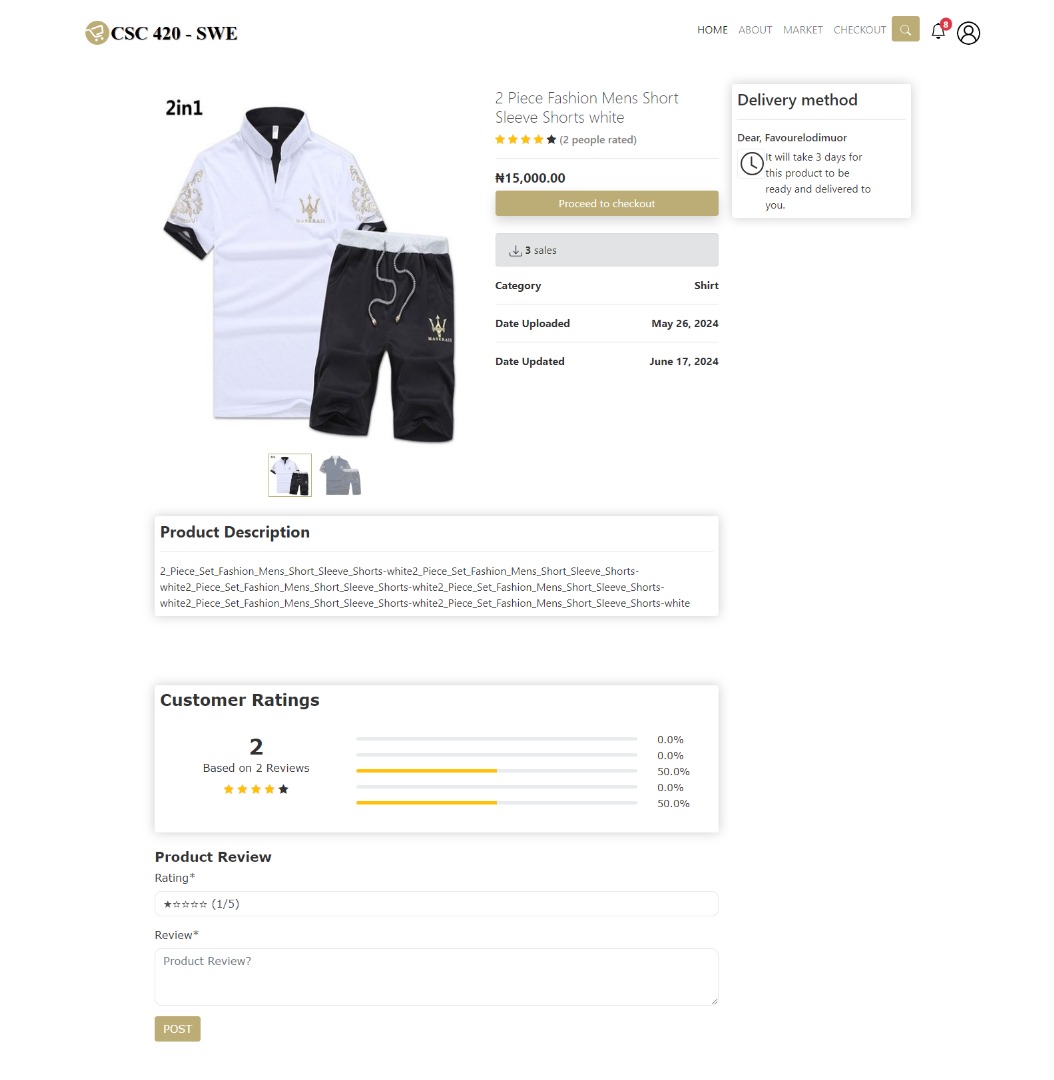
A screenshot of a computer

Description automatically generated

**Figure 8: Quick Access Cart Section**

**Product Page: Detailed Product Information**

1. Product pages were designed to provide comprehensive information about each item. Key elements include high-quality images, detailed descriptions, specifications, reviews, and related products.
2. The design ensures that users have all the necessary information to make informed purchasing decisions.



**Figure 9: Detailed Product Information Page**

## VI. CODING

The e-commerce app's coding stage signifies the change from design concepts to functional reality. During this pivotal phase, the carefully constructed designs and user experience were implemented using programming code. This stage was the central aspect of the software development process, during which the conceptualization of an online commerce platform materialized into a concrete digital solution.

Transforming Design into code: The design phase explained in the previous stage served as the blueprint for the coding process. The selected software developer took those designs and converted them into code, utilizing programming languages, libraries, and frameworks that best suited the app's requirements. This phase involved several key steps

Front-End Development: Our frontend developers played a crucial role in bringing the visual aspects of the e-commerce app to life. Their work ensured that the user interface was not only aesthetically pleasing but also functional and intuitive. Here are the key contributions and tasks undertaken by the frontend developers:

1. Translating Designs into Code: Using HTML, CSS, and JavaScript, the frontend developers translated the UI/UX designs into dynamic and responsive web pages. They followed the design criteria meticulously to maintain consistency between the design mockups and the real application.
2. Ensuring Cross-Browser Compatibility: The frontend team worked carefully to make sure the app's UI operated flawlessly across multiple browsers and devices. This needed significant testing and changes to accommodate browser-specific idiosyncrasies and maintain a consistent user experience.

Some key technologies employed by the front-end developers include:

1. HTML and CSS: These foundational languages are used to structure the content and style the visual elements of the application's user interface.
2. JavaScript: JavaScript is employed to build dynamic and interactive front-end components, enhancing user engagement and responsiveness.
3. PyCharm: This integrated development environment (IDE) provides a robust platform for front-end developers to write, edit, and debug code efficiently.

Backend Development: Our backend developers were essential in building the underlying structure of the e-commerce app, ensuring robust functionality, security, and performance. Here's an overview of the key contributions and tasks undertaken by the backend team:

1. Server-Side Logic Implementation: The backend developers implemented the server-side logic that powers the app using programming languages like Python and frameworks such as Django. This included handling user authentication, managing data, and processing transactions.
2. Database Management: They designed and managed the app's database schema, ensuring efficient data storage and retrieval. Using PostgreSQL, the backend team handled data related to products, users, orders, and reviews.
3. API Development and Integration: The developers not only created RESTful APIs to facilitate communication between the front-end and back-end components but also integrated existing APIs. This included connecting to third-party services for authentication.
4. Performance Optimization: They focused on optimizing the performance of the server-side operations to ensure fast and efficient handling of user requests. This included optimizing database queries and load balancing.
5. Integration with Third-Party Services: The backend developers integrated various third-party services and APIs to enhance the app's functionality. This included payment gateways, delivery services, and analytics tools.
6. User Authentication and Authorization: Implementing secure user authentication and role-based access control was crucial for protecting user data and ensuring only authorized users could perform certain actions.
7. Collaboration with Frontend Developers: The backend team worked closely with frontend developers to ensure seamless integration and functionality across the entire application. This collaboration ensured that the APIs met the frontend requirements and provided the necessary data and services..

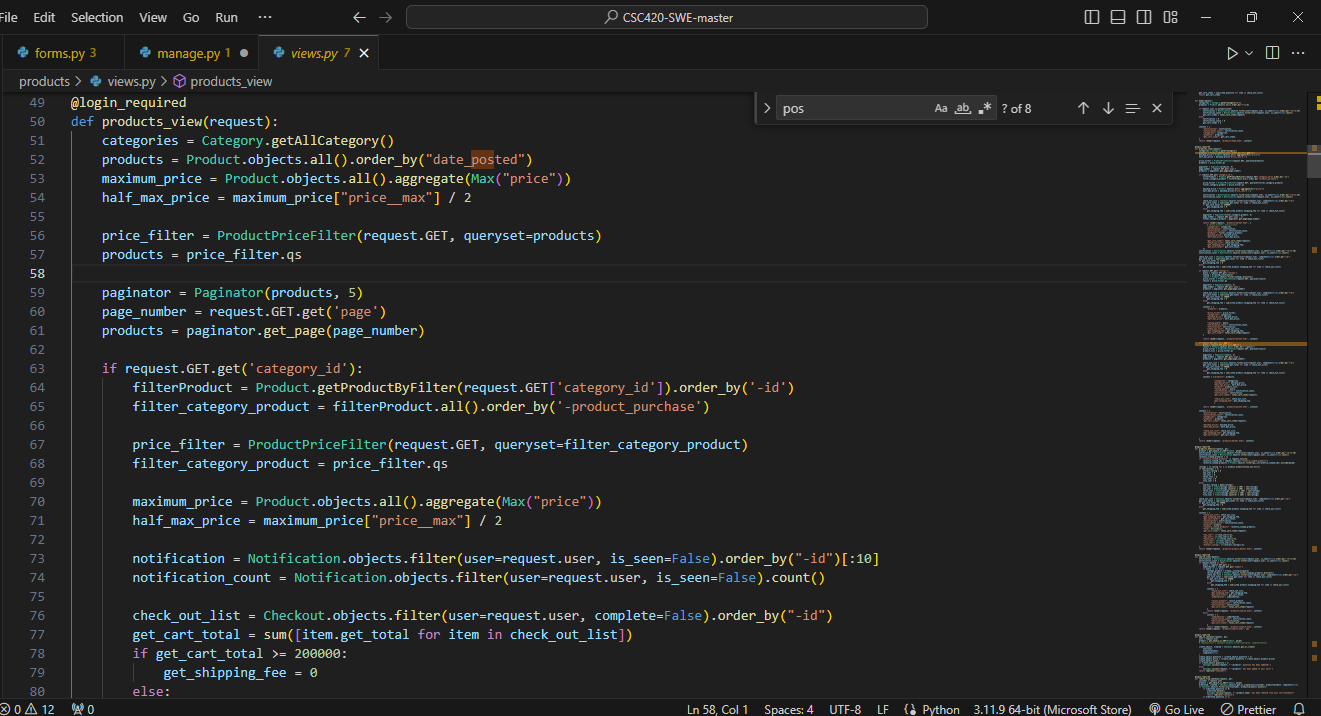
Some key technologies employed by the back-end developers include:

1. Django: A high-level Python web framework that enables rapid development and clean, pragmatic design. Django's batteries-included philosophy makes it straightforward to build web applications.
2. PostgreSQL: A powerful open-source relational database management system (RDBMS) known for its reliability and robust feature set. PostgreSQL efficiently stores and manages expense records while providing strong data integrity and scalability.
3. Dbeaver: A comprehensive PostgreSQL administration tool, Dbeaver offers a graphical interface for database management and visualization, aiding developers in monitoring and maintaining the database.
4. PyCharm: A versatile integrated development environment (IDE) that provides Django developers with tools for writing, editing, and debugging code efficiently, enhancing productivity and code quality.
5. Django-cors-headers: A Django middleware that allows secure cross-origin resource sharing (CORS), ensuring controlled data exchange between different domains in a Django application.

Through these efforts, our backend developers ensured that the e-commerce app was secure, efficient, and capable of delivering a smooth and reliable user experience.

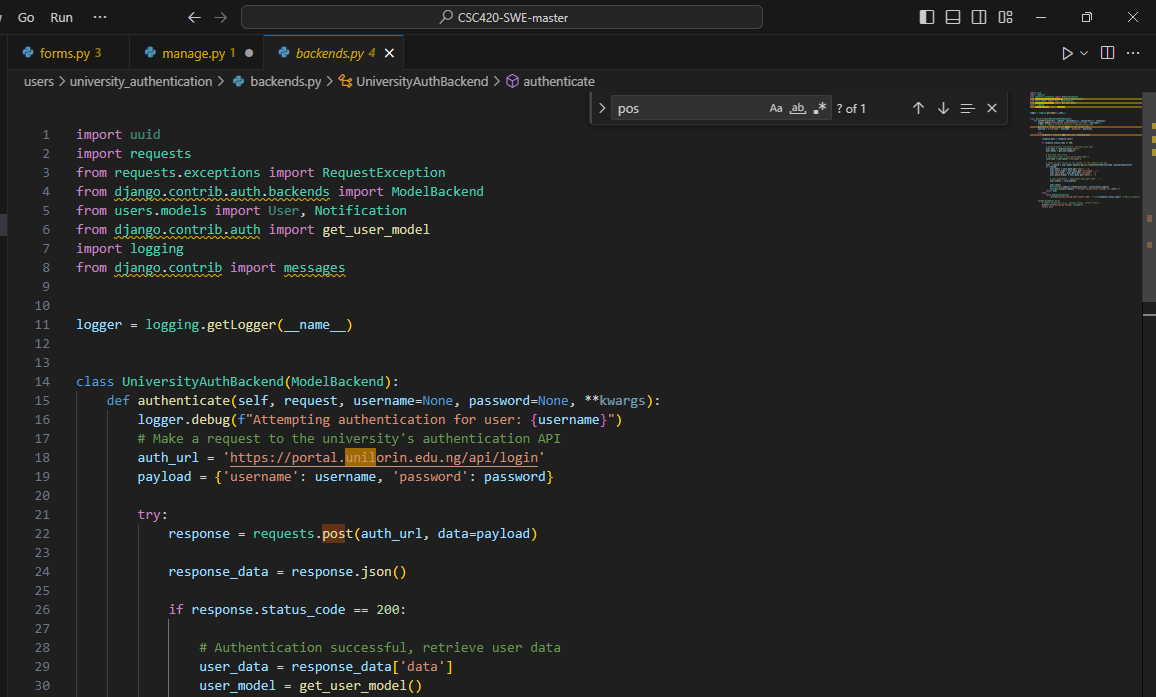
Implementing Functionality: The E-commerce app functionality were brought to life during the coding phase. Some key functionality include:

1. Category and Price Filtering: Products can be filtered by category and price range. The shipping fee is calculated based on the total cart value.



**Figure 10. Code snippet for Category and Price Filtering and shipping fee**

1. User Authentication: The developers develop a custom authentication backend for the e-commerce app, integrating with the university's authentication API to verify user credentials. The developers ensured that only users authenticated by the university's system can access the e-commerce platform, enhancing security and leveraging existing user data.

****

**Figure 11. Code snippet for User Authentication**

**Ensuring Data Security and Privacy**

Developers prioritize the security of users' data. They implement the bcrypt package for hashing passwords and utilize the university's API to access the database, protecting sensitive information and ensuring that user data remains private and confidential.

**Iterative Development**

The process of coding is continuous. Developers typically engage with designers, testers, and other team members to ensure that the code corresponds with the planned functionality and user experience. Regular code reviews and testing help detect and correct errors early in the development process.

## VII. TESTING

The testing phase for the eCommerce application developed for students and university staff at the University of Ilorin was crucial in ensuring the app's reliability, functionality, security, and overall user experience. The application, built with Django for the backend, SQLite for development, PostgreSQL for production database, and HTML, CSS, and JavaScript for the frontend, features product listings, shopping carts, and payment processing.

This section details the various testing methodologies and activities carried out.

**Testing Strategies**

The testing team employed a comprehensive testing strategy that included unit testing, integration testing, system testing, and acceptance testing to ensure the application met all functional and non-functional requirements.

**Unit Testing**

Unit testing focused on individual components to ensure they function correctly in isolation. Some test cases used include:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Test Steps | Test Data | Expected Results | Actual Results | Pass/Fail |
| TU01 | Check Customer Login with Valid Data | 1. Go to site  2. Enter Email Address  3. Enter Password  4. Click Login | Email: testuser@unilorin.edu.ng  Password: correctpassword | User should log in successfully | As expected | Pass |
| TU02 | Check Customer Login with Invalid Data | 1. Go to site  2. Enter Email Address  3. Enter Password  4. Click Login | Email: testuser@unilorin.edu.ng  Password: wrongpassword | User should not be able to log in | As expected | Pass |
| TU03 | Adding Product to Cart | 1. Click on a product  2. Click 'Add to Cart' | Product: Laptop | Product should be added to the cart | As expected | Pass |
| TU04 | Removing Product from Cart | 1. Go to cart  2. Click 'Remove' on a product | Product: Laptop | Product should be removed from the cart | As expected | Pass |
| TU05 | Checkout Process | 1. Go to cart  2. Click 'Checkout'  3. Enter shipping information  4. Complete payment | Shipping Info: Valid address  Payment Info: Valid credit card | Order should be placed successfully | As expected | Pass |

**Integration Testing**

Integration testing was performed to ensure that different modules and components of the application work together as expected. Key integration points tested include:

1. User Authentication and Authorization: Ensuring that the authentication system interacts correctly with the user database and session management.
2. Shopping Cart and Product Catalog: Verifying that products can be added to the cart, quantities updated, and products removed, with the changes accurately reflected in the user interface and database.
3. Order Management and Payment Gateway: Ensuring that orders are created correctly, payment processes are initiated, and responses from the payment gateway are handled appropriately.

Test cases for integration testing include:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Description | Test Steps | Test Data | Expected Results | Actual Results | Pass/Fail |
| TI01 | User Registration and Login | 1. Register new user  2. Log in with new credentials | Email: newuser@unilorin.edu.ng  Password: newpassword | User should register and log in successfully | As expected | Pass |
| TI02 | Add Product to Cart | 1. Log in  2. Browse products  3. Add product to cart | Product: Book | Product should appear in the cart | As expected | Pass |
| TI03 | Complete Order with Payment | 1. Add product to cart  2. Proceed to checkout  3. Enter payment details  4. Confirm payment | Payment Info: Valid credit card | Order should be completed, and payment processed | As expected | Pass |

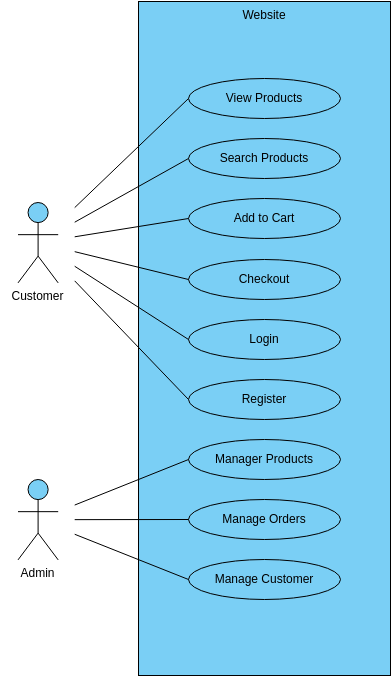
**System Testing**

System testing ensured that the entire application functions correctly as a whole. This included:

1. Alpha Testing: Conducted by the development team to identify and fix initial bugs.
2. Beta Testing: Conducted by a select group of end-users (students and staff) to gather feedback and identify any usability issues.
3. Acceptance Testing: Conducted to verify that the application meets the requirements and is ready for deployment.

**Use-Case Diagram**

A use-case diagram illustrates how the system interacts dynamically, helping compile the system's requirements. It helps identify actors, create use cases, and understand system functionalities.



**Figure 12: Use-case Diagram**

**Summary of Findings**

During the testing phase, the eCommerce application was rigorously tested to ensure compliance with all specified requirements. Key findings and actions included:

1. Bug Fixes: Identified bugs were fixed promptly.
2. Performance Improvements: Optimized code to enhance performance.
3. User Feedback: Incorporated feedback from beta testers to improve usability.

By employing these testing strategies, the team ensured the eCommerce application for the University of Ilorin was reliable, secure, and provided a seamless user experience, ready for deployment to students and staff.

## VIII. MAINTENANCE

In the ever-evolving world of e-commerce, a well-maintained website is am important factor for success. This chapter outlines a comprehensive maintenance plan to ensure our website remains secure, optimized and delivers a seamless user experience for the users. Our maintenance plan will involve the following:

Performance Optimization: A fast-loading website is crucial for optimal user experience and search engine ranking. Our strategies for performance optimization include:

1. Image optimization: Reducing the file size without compromising quality.
2. Caching mechanisms to store frequently accessed data for faster loading.
3. Code optimization to minimize unnecessary code and improve loading speed.
4. Regular monitoring of website performance using analytics tools.

Content Management: Fresh and accurate content is key to a thriving online store. Our content management plans include:

1. Regularly update product descriptions and promotions to keep customers engaged and informed.
2. Ensure product descriptions are clear, informative and up-to-date.
3. Clearly communicate promotions and sales to avoid confusion. Update expiration dates and product availability to maintain customer trust.

Monitoring and Reporting: Just like any other business, our online store thrives on insights. Here is how we intend to leverage website analytics:

1. Website Traffic Watch: We will monitor website traffic using analytics tools to understand visitor numbers, sources, and peak browsing times.
2. User Behavior: Track user behavior to see which pages are popular, where drop-offs occur, and how customers navigate our store.
3. Analyze sales performance to identify top-selling products, conversion rates and any areas needing improvement.
4. Data-driven decisions: Generate regular reports to translate website activity into actionable insights. Use this data to optimize our website, marketing strategies, and product offerings.

Security Measures: Customer trust is paramount in keeping a business running. Here’s how we intend to fortify our website:

1. We will prioritize our website’s security to safeguard customer data and prevent security breaches.
2. We will enforce strong password policies for both customer and admin accounts.
3. We will religiously update security software and patch vulnerabilities to stay ahead of evolving threats.

Maintenance Schedule

A well-oiled website requires consistent care. Here is a schedule we plan to adopt to keep our online store functioning at it’s best:

Daily Checkup

1. We will schedule Schedule daily backups to ensure we have a recent copy of our website data in case of emergencies.
2. We will run daily security scans to identify and address any malware or vulnerabilities before they cause problems.

Weekly Website Checkup

1. Devote time each week to review website analytics. Gain insights into customer behavior and identify areas for improvement.
2. Regularly update product inventory and pricing to maintain accuracy and avoid customer frustration.
3. Conduct weekly tests of core functionalities like search bars and the checkout process to ensure a seamless user experience.

Monthly Content Refresh

1. Update product descriptions, blog posts, and other website content on a monthly basis to keep our store fresh and informative.
2. Regularly review user reviews and ratings to address customer concerns and build trust.
3. Apply monthly updates for plugins, themes, and other software to benefit from bug fixes and security enhancements.

Quarterly Performance Review

Conduct quarterly reviews of website performance metrics like loading speed and mobile responsiveness to identify areas for optimization.

1. Schedule regular user testing sessions to gather feedback on the website’s usability and make improvements based on user experience.
2. Analyze the effectiveness of our marketing campaigns on a quarterly basis to see what’s working and adjust strategies as needed.

## IX. CONCLUSION

The development of the e-commerce website for university students was a comprehensive and rewarding project that achieved its primary objectives of creating a secure, user-friendly platform tailored specifically for the university community. This project provided invaluable insights into the full lifecycle of web development, from initial planning and design through to implementation, testing, and deployment.

Key Achievements

1. User Authentication: We successfully implemented a robust authentication system that ensures only university students can register and log in. This involved integrating secure login protocols and using encryption methods to protect user data.
2. Product Categorization: The website features an intuitive categorization system that makes it easy for users to browse and find products. This categorization not only improves the user experience but also helps in organizing the inventory effectively.
3. Seamless Checkout Process: One of the key features of the website is its streamlined checkout process. We integrated secure payment gateways and ensured that the process is straightforward and efficient, enhancing the overall user satisfaction.
4. Responsive Design: The website was designed to be fully responsive, ensuring a consistent and user-friendly experience across various devices and screen sizes. This was achieved through careful planning and the use of modern front-end frameworks.
5. Security Measures: Significant attention was given to the security aspects of the website. From data encryption and secure authentication to regular security testing, we implemented multiple layers of security to safeguard user information and transactions.

Lessons Learned

Throughout the development process, several valuable lessons were learned:

1. Importance of Planning: Detailed planning and requirement gathering at the beginning of the project were crucial for its success. This helped in setting clear objectives and managing time effectively.
2. Iterative Development: Adopting the Agile methodology allowed for continuous improvement and adaptation to feedback. Regular sprints and reviews helped in identifying issues early and making necessary adjustments.
3. User-Centered Design: Conducting user testing and gathering feedback was essential in creating a user-friendly interface. This ensured that the website met the actual needs and preferences of the target users.
4. Security Best Practices: Implementing security best practices was a critical aspect of the project. This included understanding and mitigating potential vulnerabilities and ensuring compliance with data protection standards.

Future Enhancements

While the project successfully met its initial objectives, there are several areas for potential future enhancements:

1. Enhanced Search Functionality: Improving the search functionality to include advanced filtering options and personalized recommendations based on user behavior.
2. Mobile App Development: Developing a mobile application to complement the website, providing users with more flexibility and a better shopping experience on the go.
3. Expanded Payment Options: Integrating additional payment methods to cater to a broader range of user preferences.
4. Marketing and Promotion: Implementing marketing strategies to increase user engagement and attract more university students to the platform.

In conclusion, the e-commerce website for university students not only serves its intended purpose but also lays a strong foundation for future improvements and expansions. The project has provided a practical experience in web development and has equipped the team with skills and knowledge that will be valuable in future endeavors.