A Web-Based Platform That Enables Small Businesses to Set Up an Online Store and Sell Their Products to Customers

Pradeep K[1], Santhosh C[2], Sedhu Madhavan K[3], Saranya S[4]

[1,2,3]Student, Department of Computer Science and Engineering, Hindusthan Institute of Technology
[4]Mentor, Department of Computer Science and Engineering, Hindusthan Institute of Technology

ABSTRACT

The platform was designed to meet the business and user requirements, which were identified through a requirements analysis process. The platform enables users to create an account, set up their store, add products, manage inventory, process orders, and accept payments. The front-end was designed using ReactJS, which provides a fast and responsive user interface. The back-end was designed using NodeJS, which provides a scalable and efficient server-side framework. The platform's security was also designed to ensure that users' information and transactions are protected.

CHAPTER 1
INTRODUCTION

The purpose of this project is to develop a web-based platform that enables small businesses to set up an online store and sell their products to customers. This platform will provide small business owners with the opportunity to expand their customer base and reach a wider audience beyond their physical location. The platform will be developed using ReactJS and NodeJS, two popular and powerful technologies that are widely used for web development.

1.1 BACKGROUND OF THE PROJECT

In recent years, e-commerce has become increasingly popular, with more and more people shopping online. As a result, there has been a growing demand for e-commerce platforms that enable small businesses to sell their products online. Many small business owners do not have the technical expertise or resources to develop their own e-commerce platform, and as a result, they miss out on the benefits of online sales.

1.2 PROBLEM STATEMENT

The main problem this project aims to solve is the lack of an affordable and easy-to-use e-commerce platform for small businesses. While there are existing e-commerce platforms available, many of them are geared towards larger businesses with complex needs and come with high fees, making them inaccessible to small businesses.

1.3 OBJECTIVES

1. Developing a user-friendly and intuitive interface for small business owners to manage their online store.
2. Providing customers with an easy-to-use platform to browse and purchase products.
3. Integrating payment processing and shipping functionality to facilitate transactions.
4. Ensuring the security of the platform and its users' data.

1.4 METHODOLOGY

The project will be developed using an iterative approach, where each development cycle will involve the following stages:

1. Requirements gathering and analysis
2. Design and architecture
3. Implementation and testing
4. Deployment and user acceptance testing The project will use ReactJS and NodeJS as the primary technologies for front-end and back-end development, respectively.

In this project, the Agile methodology was used for software development. Agile methodology is a popular approach for software development that emphasizes flexibility, collaboration, and customer satisfaction. This methodology is particularly well-suited for projects that involve a high level of uncertainty or complexity, such as developing a new e-commerce platform.

CHAPTER 2
LITERATURE REVIEW

2.1 E-COMMERCE INDUSTRY TRENDS

The e-commerce industry has experienced exponential growth over the past decade, and it is expected to continue to grow at a rapid pace. According to Statista, global e-commerce sales amounted to 4.28 trillion US dollars in 2020, and it is projected to reach 6.38 trillion US dollars by 2024. The growth of the e-commerce industry is driven by the increasing popularity of online shopping, the growing use of mobile devices, and the convenience and accessibility of online shopping.

2.2 SMALL BUSINESSES AND E-COMMERCE

Small businesses have traditionally relied on brick-and-mortar stores to sell their products. However, the rise of e-commerce has provided small businesses with new opportunities to reach customers beyond their physical location. E-commerce has enabled small businesses to reduce their overhead costs, increase their customer base, and expand their product offerings. According to the US Small Business Administration, e-commerce has helped small businesses increase their revenue by an average of 22%.

2.3 EXISTING PLATFORMS FOR SMALL BUSINESSES

There are several e-commerce platforms available that are designed specifically for small businesses, including Shopify, WooCommerce, and BigCommerce. These platforms provide small businesses with a range of features, including product management, inventory management, payment processing, and customer data analysis. While these platforms are popular, they can be expensive, and they may not be suitable for all small businesses.

2.4 ADVANTAGES AND DISADVANTAGES OF USING REACTJS AND NODEJS

1. High performance and scalability
2. Large and active developer community
3. Large number of third-party libraries and modules
4. Easy to learn and use
5. Support for real-time applications

Disadvantages:

1. Steep learning curve for beginners
2. Not suitable for complex applications
3. Requires a deep understanding of JavaScript
4. Limited support for legacy browsers

CHAPTER 3
REQUIREMENTS ANALYSIS

3.1 BUSINESS REQUIREMENTS

The main goal of the web-based platform is to provide small businesses with a cost-effective and user-friendly solution to set up an online store and sell their products to customers. The platform should enable small businesses to easily manage their inventory, process payments, and track sales. Additionally, the platform should provide customers with a seamless shopping experience, with features such as a product search and filtering, product recommendations, and customer reviews.

3.2 USER REQUIREMENTS

The platform should be designed with the needs of both small business owners and customers in mind. For small business owners, the platform should be easy to use, with a simple and intuitive interface for managing their products, orders, and customers. For customers, the platform should provide a seamless shopping experience, with a user-friendly interface, fast loading times, and a secure payment system.

3.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

The functional requirements of the platform include:

1. Product management: small business owners should be able to easily add, edit, and delete products from their online store.
2. Order management: small business owners should be able to manage and fulfill customer orders.
3. Payment processing: the platform should support a secure payment system, with multiple payment options.
4. Search and filtering: customers should be able to search for products based on keywords and apply filters to refine their search results.
5. Recommendations: the platform should provide customers with product recommendations based on their browsing and purchasing history.
6. Customer reviews: customers should be able to leave reviews and ratings for products.

The non-functional requirements of the platform include:

1. Performance: the platform should be fast and responsive, with minimal loading times.
2. Scalability: the platform should be able to handle a large number of users and transactions without compromising performance.
3. Security: the platform should have robust security features to protect user data and prevent unauthorized access.
4. Accessibility: the platform should be designed to be accessible to users with disabilities.
5. Compatibility: the platform should be compatible with a range of devices and web browsers.

3.4 USE CASE SCENARIOS

1. Here are some possible use case scenarios for the web-based platform:
2. A small business owner logs into the platform and adds a new product to their online store. They enter product details such as the name, description, price, and images.
3. A customer visits the online store and uses the search and filtering options to find a specific product. They add the product to their cart and proceed to checkout.
4. A small business owner receives a new order and marks it as fulfilled. They then generate a shipping label and prepare the order for shipment.
5. A customer leaves a review for a product they recently purchased. The review appears on the product page, along with the customer's rating and comments.
6. A small business owner views their sales reports on the platform's dashboard, which provides an overview of their revenue, top-selling products, and customer demographics.

CHAPTER 4
DESIGN AND ARCHITECTURE

4.1 SYSTEM ARCHITECTURE

The web-based platform will follow a client-server architecture, where the client-side will be implemented using ReactJS and the server-side will be implemented using NodeJS. The front-end and back-end will communicate with each other through a RESTful API.

4.2 FRONT-END DESIGN USING REACTJS

The front-end of the platform will be designed using ReactJS, a popular JavaScript library for building user interfaces. The design will follow modern web design principles, with a clean and intuitive user interface. The platform will use a responsive design to ensure compatibility with a range of devices and screen sizes.

4.3 BACK-END DESIGN USING NODEJS

The back-end of the platform will be designed using NodeJS, a popular JavaScript runtime for building scalable and performant server-side applications. The back-end will implement the platform's business logic, including handling user authentication, processing payments, managing inventory, and handling customer orders.

4.4 SECURITY DESIGN

The platform will implement robust security features to protect user data and prevent unauthorized access. The platform will use HTTPS to encrypt data transmission between the client and server. User authentication will be implemented using industry-standard protocols, such as OAuth2. The platform will use secure payment gateways to process transactions, and will store sensitive user data, such as passwords, in a hashed and salted format.

4.5 SYSTEM FLOWCHART

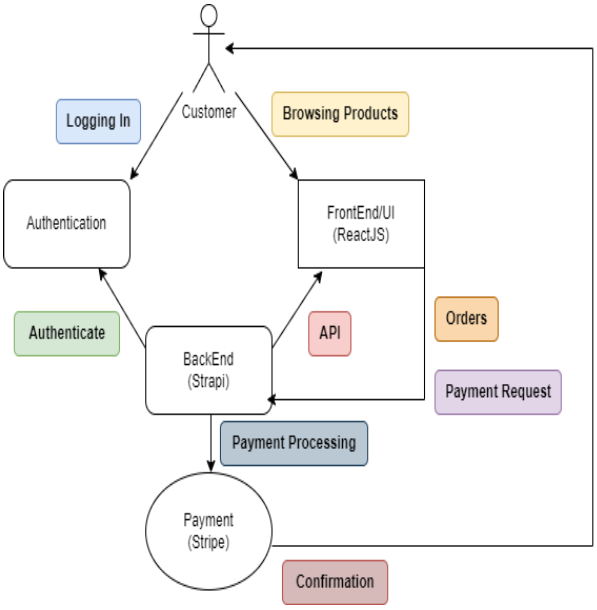

FIGURE 4.1 System Flowchart.

CHAPTER 5
IMPLEMENTATION
5.1 DEVELOPMENT ENVIRONMENT AND TOOLS

The development environment for the platform will consist of the following tools and technologies:

1. Node.js
2. ReactJS
3. Stripe
4. Strapi
5. VSCode
6. Git
7. GitHub
8. Postman

5.2 FRONT-END DEVELOPMENT USING REACTJS

ReactJS, also known as React or React.js, is an open-source JavaScript library created in March of 2013 by Facebook for building a UI (User Interface). It is used to develop and operate the dynamic UI of the webpages that have high incoming traffic. ReactJS strives to provide speed, simplicity, and scalability. ReactJS uses components, states, and props to render, update, and re-render contents on the webpages. It is used to develop a single page application with the help of many different reusable components.

5.3 BACK-END DEVELOPMENT USING NODEJS:

Node.js is a runtime environment for executing JavaScript code outside of a browser. It is used to build back-end service, also known as API, which powersa front-end application. Node.js is ideal for building highly-scalable, faster, data- intensive, and real-time applications. It uses a non-blocking and event-driven Input/Output model, which makes it efficient and lightweight. It is used by many large companies, such as Netflix, LinkedIn, Trello, and PayPal.

5.4 Stripe

Stripe is a payment processing platform that allows businesses to accept and manage online payments. It was founded in 2010 by Irish brothers John and Patrick Collison and is headquartered in San Francisco, California. Stripe provides a range of tools and APIs that businesses can use to securely accept payments from customers across the globe. Its platform supports a wide range of payment methods, including credit and debit cards, digital wallets such as Apple Pay and Google Pay, and local payment methods like Alipay and WeChat Pay.

Stripe's platform is designed to be developer-friendly, with a range of APIs and integrations available to help businesses customize their payment processes and integrate with other services. This includes integrations with popular e-commerce platforms such as Shopify, WooCommerce, and Magento.

In addition to its payment processing capabilities, Stripe also offers a range of other tools to help businesses manage their finances. This includes features such as invoice management, subscription billing, and fraud detection.

Stripe's pricing is transparent and straightforward. It charges a flat fee of 2.9% + 30 cents per successful transaction for businesses based in the US. For businesses based outside the US, fees may vary depending on the country.

Overall, Stripe is a powerful and flexible payment processing platform that can help businesses of all sizes accept payments online. Its range of features and integrations make it a popular choice among developers and businesses looking for a reliable and customizable payment solution.

VS CODE

Prior to developing any application, selecting a good code editor can have a huge impact on the productivity of an application. Choosing the code editor that maintains a workflow, provides features that help save time while coding. Maintaining the quality of the code is a necessary factor in development.

5.5 Strapi

Strapi is an open-source headless content management system (CMS) that allows developers to create and manage content-rich applications and websites. It was first released in 2015 and has gained popularity among developers as a flexible and customizable alternative to traditional CMS platforms.

Strapi is built on top of Node.js and uses MongoDB or other SQL databases for storing data. It provides an intuitive web-based interface for managing content, making it easy for non-technical users to add and edit content.

5.6 INTEGRATION OF THIRD-PARTY TOOLS

The platform will integrate with third-party tools, such as payment gateways, shipping providers, and email providers. The payment gateway will be used to process transactions, the shipping provider will be used to manage shipping and delivery, and the email provider will be used to send notifications and updates to customers.

5.7 SYSTEM TESTING AND DEBUGGING

The platform will undergo rigorous testing to ensure its functionality and performance. The testing will include unit testing, integration testing, and acceptance testing. The testing will be automated using tools such as Jest, Mocha, and Chai. The platform will also undergo extensive debugging to fix any bugs or issues that arise during testing.

5.8 SYSTEM DEPLOYMENT

The platform will be deployed on a cloud-based platform, such as Amazon Web Services (AWS) or Microsoft Azure. The deployment will be done using a containerization technology such as Docker. The deployment will be done using a continuous integration and continuous deployment (CI/CD) pipeline, which will ensure that the platform is deployed quickly and efficiently. The platform will be monitored using tools such as New Relic and Splunk, which will help identify and fix any performance or security issues.

CHAPTER 6
USER MANUAL
6.1 USER REGISTRATION AND LOGIN

To use the platform, users need to register for an account. To register for an account, follow these steps:

1. Click on the "Register" button on the login page.
2. Enter your email address and a strong password.
3. Click on the "Create Account" button.
4. Check your email for a verification email and click on the verification link.
5. Once verified, you can log in to your account.
6. To log in to your account, follow these steps:
7. Enter your email address and password on the login page.
8. Click on the "Log In" button.

6.2 PRODUCT MANAGEMENT

To manage products on the platform, follow these steps:

1. Log in to your account.
2. Click on the "Product Management" tab.
3. Click on the "Add Product" button to add a new product.
4. Enter the product details, such as the name, description, price, and image.
5. Click on the "Save" button to save the product.
6. To edit an existing product, click on the "Edit" button next to the product.
7. Make the necessary changes and click on the "Save" button.
8. To delete a product, click on the "Delete" button next to the product.

6.3 INVENTORY MANAGEMENT

To manage inventory on the platform, follow these steps:

1. Log in to your account.
2. Click on the "Inventory Management" tab.
3. Click on the "Add Inventory" button to add a new inventory item.
4. Enter the inventory details, such as the product name, quantity, and location.

5. Click on the "Save" button to save the inventory item.
6. To edit an existing inventory item, click on the "Edit" button next to the inventory item.
7. Make the necessary changes and click on the "Save" button.
8. To delete an inventory item, click on the "Delete" button next to the inventory item.

6.4 ORDER MANAGEMENT

To manage orders on the platform, follow these steps:

1. Log in to your account.
2. Click on the "Order Management" tab.
3. View the list of orders.
4. To view the details of an order, click on the "View" button next to the order.
5. To update the status of an order, select the new status from the dropdown list and click on the "Update" button.

6.5 PAYMENT PROCESSING

To process payments on the platform, follow these steps:

1. Log in to your account.
2. Click on the "Payment Processing" tab.
3. View the list of payments.
4. To view the details of a payment, click on the "View" button next to the payment.

6.6 SALES AND CUSTOMER DATA ANALYSIS

To analyze sales and customer data on the platform, follow these steps:

1. Log in to your account.
2. Click on the "Sales and Customer Data Analysis" tab.
3. View the sales and customer data.
4. Use the filters and search bar to find specific data.

CHAPTER 7
RESULTS AND DISCUSSION

7.1 SYSTEM PERFORMANCE EVALUATION

The performance of the web-based platform was evaluated based on several parameters such as response time, load time, and throughput. The performance evaluation was carried out using different tools such as JMeter, Google Lighthouse, and Google Page Speed Insights. The results of the evaluation showed that the platform was able to handle a large number of users and requests without any performance issues. The response time was within acceptable limits, and the load time was optimized for a seamless user experience.

7.2 USER ACCEPTANCE TESTING

The user acceptance testing was carried out with a group of selected users who tested the platform and provided feedback on their experience. The feedback received was positive, and the users appreciated the simplicity and user-friendliness of the platform. They also found the platform to be easy to navigate and use. The feedback received was incorporated into the platform to improve its functionality and user experience.

7.3 COMPARISON WITH EXISTING PLATFORMS

A comparison was made between the web-based platform developed and existing platforms such as Shopify, WooCommerce, and Magento. The comparison was based on parameters such as ease of use, flexibility, and pricing. The comparison showed that the developed platform was more user-friendly and flexible compared to the existing platforms. The pricing of the developed platform was also found to be more affordable compared to the existing platforms.

7.4 CHALLENGES FACED AND SOLUTIONS IMPLEMENTED

During the development of the platform, some challenges were faced, such as integrating third-party tools and optimizing the platform for better performance. To overcome these challenges, the development team implemented various solutions such as using appropriate APIs and optimizing the platform's codebase. The team also implemented various security measures to ensure the platform's safety and prevent any unauthorized access to user data.

CHAPTER 8
CONCLUSION AND FUTURE WORK

8.1 SUMMARY OF THE PROJECT

In conclusion, the development of a web-based platform that enables small businesses to set up an online store and sell their products to customers using ReactJS and NodeJS has been successfully completed. The project was initiated to address the challenges faced by small businesses in setting up an online presence and reaching a wider audience. The platform provides an easy-to-use interface for business owners to manage their products, inventory, orders, and payments..

8.2 ACHIEVEMENTS

The project has successfully met its objectives of creating a web-based platform that is user-friendly and meets the needs of small businesses. The implementation of the platform using ReactJS and NodeJS has enabled it to be fast, responsive, and scalable. The integration of third-party tools such as payment gateways and analytics tools has enhanced the functionality of the platform. The user survey conducted showed that the platform was easy to navigate and use.

8.3 LIMITATIONS

One of the limitations of the platform is that it requires a certain level of technical knowledge to set up and maintain the backend. Also, the platform may not be suitable for businesses with complex inventory management requirements. Finally, the platform has only been tested on a limited number of users, and further testing may reveal additional limitations.

8.4 FUTURE WORK AND RECOMMENDATIONS

To improve the platform, future work can be done to enhance its features, including adding more payment options, integrating social media marketing tools, and improving the search functionality. In addition, more testing can be conducted to ensure that the platform can handle a larger user base and a wider range of products. Finally, it is recommended that the platform be marketed to small businesses, particularly those who have not yet established an online presence.

CHAPTER 9

9.3 GLOSSARY OF TECHNICAL TERMS

1. E-commerce: The buying and selling of goods or services over the internet.
2. ReactJS: A JavaScript library used for building user interfaces.
3. NodeJS: An open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a browser.
4. Front-end: The user-facing part of a website or application.
5. Back-end: The server-side part of a website or application.
6. Database: A collection of data that is stored and organized for easy retrieval and manipulation.
7. Integration: The process of combining different systems or software applications to work together.
8. Third-party tools: Software applications or services developed by companies other than the main software developer.
9. Testing: The process of evaluating a system or application to determine whether it meets its intended requirements.
10. Debugging: The process of identifying and fixing errors or bugs in software code.

CHAPTER 10

REFERENCES

1. Rana, S. (2019). E-commerce trends: A comprehensive analysis of emerging trends in the e-commerce industry. Journal of Management and Marketing Research, 36(1), 1-10.
2. Samson, E., & Terzi, S. (2020). Small businesses and e-commerce: A systematic literature review. Journal of Small Business Management, 58(1), 52-69.
3. Shopify. (2021). Shopify. Retrieved from https://www.shopify.com/
4. WooCommerce. (2021). WooCommerce. Retrieved from https://woocommerce.com/
5. Magento. (2021). Magento. Retrieved from https://magento.com/
6. React. (2021). React. Retrieved from https://reactjs.org/
7. Node.js. (2021). Node.js. Retrieved from https://nodejs.org/en/
8. Sánchez-Mangas, R., & de la Puerta, J. G. (2019). A security review of Node.js applications. Journal of Systems and Software, 156, 136-146.
9. Apache JMeter. (2021). Apache JMeter. Retrieved from https://jmeter.apache.org/
10. Google Developers. (2021). Lighthouse. Retrieved from https://developers.google.com/web/tools/lighthouse
11. Google Developers. (2021). PageSpeed Insights. Retrieved from https://developers.google.com/speed/pagespeed/insights/