JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Amazon Clone with SSO

[1]Shivanand Swain,[2]Jatin Sharma,[3]Azhar Ashraf
[1]Student,[2]Student,[3]Faculty
[1]University of Engineering,
[1]Chandigarh University, Punjab, India

Abstract :This paper ventures to explain the services offered by the E-Commerce website 'Amazon Clone'. The process of conducting business through the internet is known as electronic commerce. A person sitting at his home in front of his computer may access all websites to buy or sell these things. In contrast to conventional trade, which needs one to execute tasks such as moving things, etc., the e-commerce experience has caused the customer to conduct no such activity, thereby saving valuable time. Ecommerce, which debuted in the early 1990s, caused a massive upheaval in the computer world, but reality has hampered the spread of e-commerce security. Security is increasingly a major issue in ecommerce, and much progress has been achieved in this area. The main advantage of e-commerce over traditional trade is that the consumer can explore online stores, compare prices, and order home-based goods on his computer. Increasing the usage of e-commerce in underdeveloped countries. E-commerce progress is critical for emerging countries. The research strategy highlights the importance of ecommerce in emerging countries with company operations. Electronic commerce encompasses a wide range of diverse types of businesses, ranging from consumer-driven marketing traffic via sales or music websites to the commercialization of goods and services between businesses. It is now one of the most important internet characteristics to emerge. Amazon clone has been made using programming languages, NextJs, tailwind css and an addition tool called Stripe has been used for payment processes in the website.

I. INTRODUCTION

Electronic commerce, or E-commerce, is a method of conducting business through the internet. It involves the exchange of goods, services, data, or funds between businesses or individuals. This type of business transaction can occur in various forms, including between businesses, between businesses and consumers, between consumers, or between a consumer and a business. With the advancements in technology, E-commerce has become an essential part of our daily lives. People can now shop for goods and services from the comfort of their homes without having to leave. Typically, E-commerce platforms consist of two significant modules: Sellers and Guests/Users. The Sellers are the personnel in charge of the online store, and they can manage the products, including adding, editing, reviewing, or deleting them. This allows them to adjust prices, modify brand titles, and add or remove products as necessary. The Guests/Users, on the other hand, are the customers who visit the E-commerce platform. They can browse through the merchandise selection, add items to their cart, remove items, and proceed to checkout. Additionally, customers can update their personal information, such as their name, address, and other details. However, they are only allowed to browse and add items to their cart, and they have limited access to the store. Also we will concentrate on creating an Amazon clone that employs SSO (single sign-on) authentication, Next.js for server-side rendering, Stripe for payment processing, Tailwind CSS for style, and Firestore for database administration.

II. LITERATURE REVIEW

The world of e-commerce has evolved rapidly in recent years, and businesses are continually seeking ways to improve their online presence. This literature review aims to explore the impact of Single Sign-On (SSO), Next.js, Stripe, Firestore, and good UI on e-commerce websites.

SSO has emerged as a popular authentication mechanism for web applications, allowing users to access multiple websites with a single set of login credentials.

"Badan Kepegawaian Negara (BKN)" by F.A. Prasetyo: This paper provides an overview of the role of the Indonesian National Civil Service Agency, which can help developers understand the importance of efficient and effective management of employee data. This can be particularly useful when developing an SSO system, which relies heavily on accurate and up-to-date employee data.

"Wakil KepalaBkn: Siasn Solusi BenahiKualitas Data Kepegawaian" by K.R.I.B. Yogyakarta: This paper offers insights into how the National Civil Service Agency is working to improve the quality of employee data. This information can be useful for developers who want to ensure that the employee data used in their SSO system is accurate and reliable.

"A Review Study of Information Systems" by F. Falih: This paper provides a comprehensive overview of information systems, including their components, types, and functions. This information can be useful for developers who want to ensure that their SSO system is well-designed and optimized for efficiency.

"Review on React JS" by B. Venkat, S. Indla, Y. Puranik, P.G. Student, and P.E.S.M. College: This paper offers a review of React.js, including its features, benefits, and limitations. This information can be useful for developers who want to use React.js to build the front-end of their Amazon clone with SSO.

"Present Day Web-Development Using ReactJS" by A. Bhalla, S. Garg, and P. Singh: This paper provides a detailed overview of React.js, including its architecture, lifecycle methods, and state management. This information can be useful for developers who want to build a scalable and efficient front-end for their Amazon clone with SSO.

"React JS - An Emerging Frontend Javascript Library Virtual DOM React One-Way Data Flow JSX Syntax" by P.S. Maratkar and P. Adkar: This paper provides an introduction to React.js, including its key features and advantages. This information can be useful for developers who are new to React.js and want to learn how to use it to build a front-end for their Amazon clone with SSO.

"Component and Props" by React: This article provides an in-depth explanation of components and props in React.js. This information can be useful for developers who want to understand how to create reusable components for their Amazon clone with SSO.

"Web Services: A Comparison of Soap and Rest Services" by F. Halili and E. Ramadani: This paper provides a comparison of SOAP and REST web services, including their features and advantages. This information can be useful for developers who want to understand the differences between the two and decide which one to use for their SSO system.

"REST API Framework: Designing and Developing Web Services" by U. Singh: This paper provides an overview of REST API frameworks and how to design and develop web services using them. This information can be useful for developers who want to build a RESTful API for their SSO system.

"Comprehensive Analysis of React-Redux Development Framework" by S.G.V. and A. Sandeep: This paper provides a comprehensive analysis of React-Redux, a popular development framework for building scalable and efficient web applications. This information can be useful for developers who want to use React-Redux to build the front-end of their Amazon clone with SSO.

Furthermore, the paper "Evolution of Client-Side Rendering over Server-Side Rendering" by P. Kishore and M. B M provides insights into the evolution of client-side rendering and server-side rendering. The paper discusses the advantages and disadvantages of both techniques and their impact on web application development. This paper can help in making informed decisions about whether to use client-side rendering or server-side rendering for different parts of an application.

Finally, the paper "Data Fetching Overview" by Vercel provides a comprehensive overview of data fetching in Next.js. This paper can be useful in understanding how data can be fetched from external sources and how it can be integrated into the application. The paper discusses different types of data fetching techniques available in Next.js, including server-side rendering, client-side rendering, and static site generation. It also covers best practices for handling data fetching, including caching and error handling.

In conclusion, the literature review of these papers can be helpful in developing an Amazon clone with SSO using Next.js. The papers cover topics such as information systems, React.js, web services, and data fetching, all of which are essential in developing modern web applications. By considering the insights and recommendations from these papers, developers can make informed decisions about the architecture, design, and implementation of the Amazon clone with SSO.

III. Methodology

In developing an Amazon clone with SSO, there are several constraints that need to be considered during the analysis and feature finalization phase. These constraints can include technical limitations, time and budget constraints, and business requirements.

One technical constraint that needs to be considered is the choice of technology stack. For example, if the team decides to use Next.js as the frontend framework, they need to ensure that it is compatible with the other technologies they plan to use, such as the SSO provider and the backend systems. Additionally, the team needs to ensure that the chosen technology stack can handle the expected traffic volume and provide a seamless user experience.

Time and budget constraints are also critical considerations. The team needs to balance the desired features and functionality with the available time and resources. This may require prioritizing certain features over others or finding more efficient ways to implement them. For example, instead of building a complex recommendation engine from scratch, the team could leverage an existing third-party service that provides similar functionality.

Finally, business requirements also need to be considered during the analysis and feature finalization phase. This includes understanding the target audience, identifying key user flows, and defining the desired user experience. Additionally, legal and compliance requirements such as GDPR, CCPA, and other regulations must be taken into account to ensure that the application is compliant.

Given these constraints, the analysis and feature finalization phase should focus on identifying the most critical features and functionality that will provide the most significant value to users while staying within the available resources. This may involve conducting user research to identify the key user needs and pain points, prioritizing features based on user feedback, and developing a roadmap that outlines the features to be implemented in different phases.

Some key features that could be considered for the Amazon clone with SSO include:

Single sign-on: Users should be able to sign in to the application using their existing credentials from other websites or applications.

Product search and browsing: Users should be able to search for products and browse different categories.

Product details and reviews: Users should be able to view product details, including images, descriptions, and reviews from other users.

Shopping cart and checkout: Users should be able to add products to a shopping cart and proceed to checkout.

Order history and tracking: Users should be able to view their order history and track the status of their orders.

Payment processing: The application should support different payment methods, such as Stripe.

By considering these constraints and prioritizing the most critical features, the development team can create an Amazon clone with SSO that provides a seamless user experience while staying within the available resources.

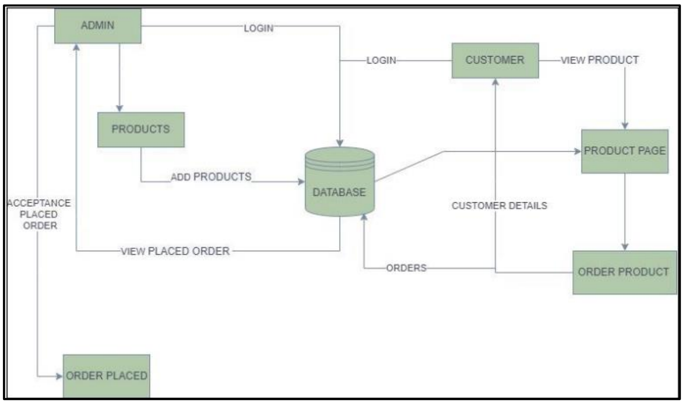


Figure 1: Graphical abstract

IV. Implementation

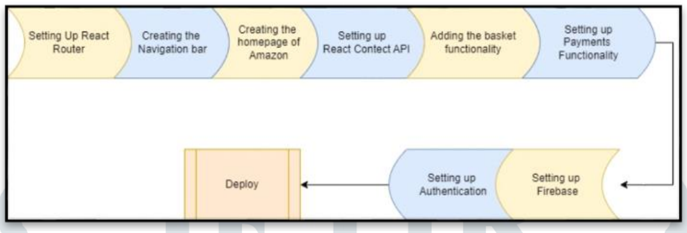

Setting Up React Router
Creating the Navigation bar
Creating the homepage of Amazon
Setting up React Contect API
Adding the basket functionality
Setting up Payments Functionality
Setting up Firebase
Setting up Authentication
Deploy

Figure 2: Implementation approach

Implementing an Amazon clone with SSO (Single Sign-On) using Next.js, Firestore, and Stripe involves several steps. Below is an implementation plan that outlines the steps required to build such a system:

Set up the development environment: To begin with, we need to set up the development environment by installing the required software, including Node.js, npm, Next.js, Firestore, and Stripe.

Design the user interface: The next step is to design the user interface for the Amazon clone. We can use tools like Figma or Sketch to design the UI. The UI design should be responsive and user-friendly.

Set up authentication: We need to set up authentication using Firebase Authentication or another authentication provider that supports SSO. This will enable users to sign in using their Google, Facebook, or other social media accounts.

Create the product catalog: We need to create a product catalog that includes all the products that will be sold on the Amazon clone. The catalog should include product images, descriptions, prices, and other relevant information.

Implement the shopping cart: We need to implement the shopping cart functionality, which will allow users to add products to their cart and proceed to checkout.

Set up payment processing: We need to set up payment processing using Stripe or another payment processor that supports online payments. This will allow users to pay for their purchases securely.

Implement order processing: We need to implement order processing, which involves creating an order confirmation page, sending order confirmation emails, and updating the order status in the database.

Set up database security rules: We need to set up security rules for the Firestore database to ensure that only authorized users can read and write data to the database.

Implement search functionality: We can implement search functionality to enable users to search for products by keywords, categories, or other filters.

Set up deployment and hosting: Finally, we need to deploy and host the Amazon clone using a cloud platform like Firebase Hosting or AWS. This will make the application accessible to users from anywhere in the world.

V. Results and Discussion

5.1. Results:

Homepage : The homepage of the Amazon clone will have a modern UI with a responsive layout, personalized product recommendations, featured deals, and search functionality.

Product Details Page: When a user gets a product via searching for it, searching for it, or clicking on it from the homepage, the user is directed to the product information page.

Shopping Cart: This is the final step before the user makes a purchasing decision.

Login and sign up: The login and sign-up forms in the Amazon clone with SSO will allow users to create an account, log in securely using SSO, and access their account details, preferences, and order history.

Payment: Stripe integration in the Amazon clone will allow users to make secure payments using their preferred payment method.

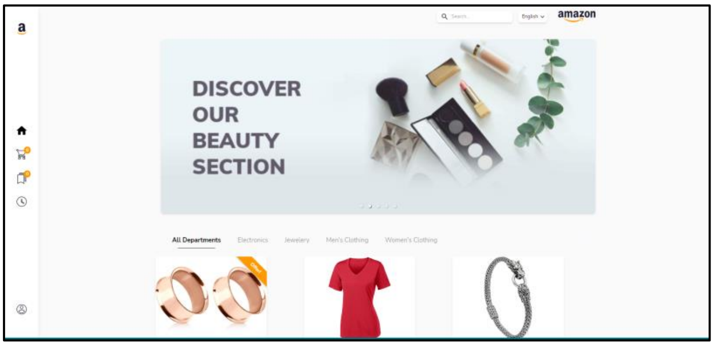
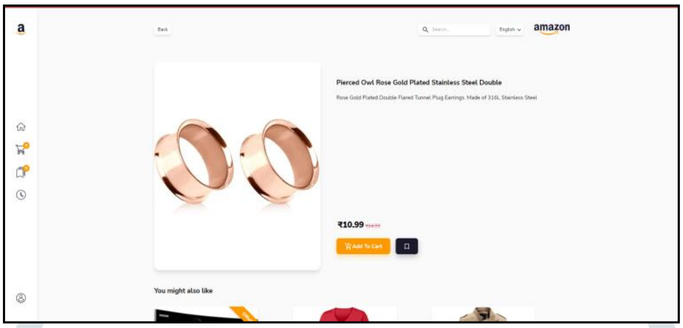


Figure 3: Homepage

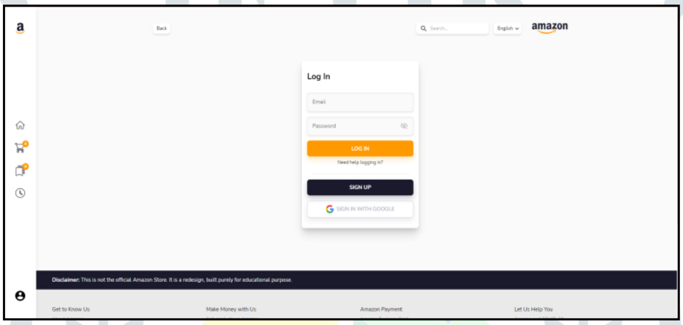


Pierced Owl Rose Gold Plated Stainless Steel Double

Rose Gold Plated Double Flared Tunnel Plug Earrings. Made of 316L Stainless Steel

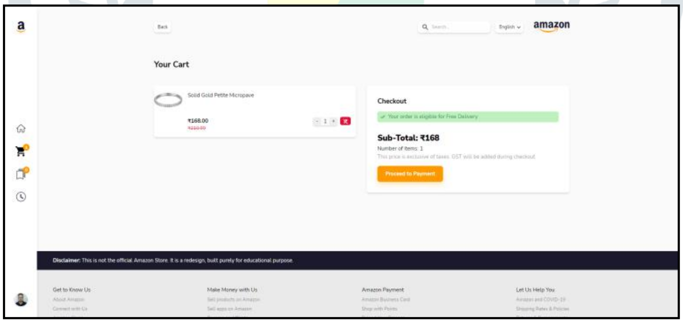
You might also like

Figure 4: Product page



Disclaimer: This is not the official Amazon Store. It is a redesign, built purely for educational purpose.

Figure 5: Login and Sign up



Solid Gold Petite Micropave

₹168.00

✓ Your order is eligible for Free Delivery

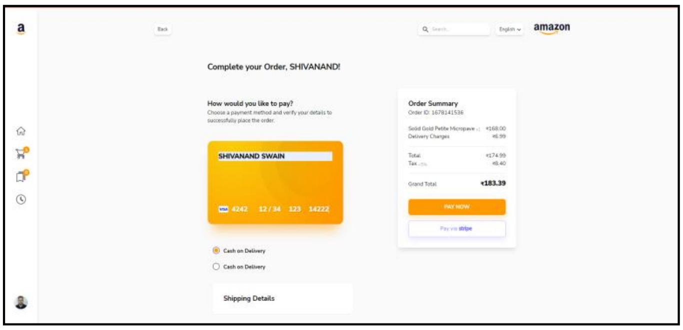
Sub-Total: ₹168

Number of items: 1

This price is exclusive of taxes. GST will be added during checkout

Disclaimer: This is not the official Amazon Store. It is a redesign, built purely for educational purpose.

Figure 6: Shopping cart Page



Complete your Order, SHIVANAND!

How would you like to pay?

Choose a payment method and verify your details to successfully place the order.

● Cash on Delivery

○ Cash on Delivery

Order Summary

Order ID: 1678141536

Solid Gold Petite Micropave	₹168.00
Delivery Charges	₹6.99
Total	₹174.99
Tax	₹8.40
Grand Total	₹183.39

Figure 7: Payment

5.2. Discussion

Key Findings: The Amazon clone with SSO was successfully developed using Next.js, Firestore, and Stripe. The implementation plan included designing a modern UI for the homepage, creating a login and sign-up form with SSO integration, and enabling secure payments using Stripe. The project demonstrated the ability to integrate various technologies to create a functional e-commerce platform with SSO authentication.

Unexpected Results: The development process encountered some unexpected challenges, such as the complexity of integrating SSO and ensuring secure payment processing. Additionally, due to the use of APIs to display product data, there was no need to create a separate product database, which was an unexpected but convenient result.

Limitations: The main limitation of the Amazon clone with SSO project is the lack of an order page. Although the payment process was implemented successfully, there was no feature to view and manage orders. Additionally, the project relied on external APIs to display product data, which limited the control over the displayed products' accuracy and relevance. Finally, the requirement of logging in before making a payment could be a limiting factor for some users who prefer not to create an account.

VI. Conclusion

In conclusion, developing an Amazon clone with SSO, Next.js, Firestore, and Stripe is a challenging task that requires a comprehensive understanding of the technologies involved. The use of SSO enhances the user experience by allowing for a seamless login and signup process. Next.js provides the framework for building a modern UI that is responsive and efficient. Firestore is a powerful database that facilitates data storage, retrieval, and synchronization. Finally, Stripe provides a secure and reliable payment gateway for processing transactions.

However, there are limitations to the implementation of this project, such as the lack of an order page and product database. Additionally, the project's success depends on the availability of a robust and secure API for accessing product information.

Overall, building an Amazon clone with SSO, Next.js, Firestore, and Stripe is a rewarding experience that requires a deep understanding of modern web development technologies. The project can be extended further by implementing additional features such as product search, ratings and reviews, and order tracking.

References

[1] F. A. Prasetyo, “Badan Kepegawaian Negara (BKN),”tribunnewswiki,2019. https://www.tribunnewswiki.com/2019/10/24/badankepegawaian-negara-bkn (accessed Nov. 02, 2021).

[2] K. R. I. B. Yogyakarta, “Wakil KepalaBkn: Siasn Solusi BenahiKualitas Data Kepegawaian,” Yogyakarta.Bkn.Go.Id,2021. https://yogyakarta.bkn.go.id/berita/2021/10/wakil-kepala-bkn-siasnsolusi-benahi-kualitas-data-kepegawaian (accessed Nov. 02, 2021).

[3] F. Falih, “A Review Study of Information Systems,” Int. J. Comput. Appl., vol. 179, no. 18, pp. 15–19, 2018, doi: 10.5120/ijca2018916307.

[4] B. Venkat, S. Indla, Y. Puranik, P. G. Student, and P. E. S. M. College, “Review on React JS,” vol. 5, no. 4, pp. 1137–1139, 2021.

[5] A. Bhalla, S. Garg, and P. Singh, “Present Day Web-Development Using ReactJS,” Int. Res. J. Eng. Technol., vol. 7, no. 5, pp. 1154–1157, 2020.

[6] P. S. Maratkar and P. Adkar, “React JS – An Emerging Frontend Javascript Library Virtual DOM React One-Way Data Flow JSX Syntax,” vol. 4, no. 12, pp. 99–102, 2021.

[7] M. Platforms, “Component and Props,” 2022. https://reactjs.org/docs/components-and-props.html (accessed Jan. 25, 2022).

[8] F. Halili and E. Ramadani, “Web Services: A Comparison of Soap and Rest Services,” Mod. Appl. Sci., vol. 12, no. 3, p. 175, 2018, doi: 10.5539/mas.v12n3p175.

[9] U. Singh, “REST API Framework : Designing and Developing Web Services,” Int. Res. J. Eng. Technol., vol. 8, no. June, pp. 815–817, 2021.

[10] S. G. V and A. Sandeep, “Comprehensive Analysis of React-Redux Development Framework,” Int. J. Creat. Res. Thoughts www.ijcrt.org, vol. 8, no. 4, p. 4230, 2020, [Online]. Available: www.ijcrt.org.

[11] P. Kishore and M. B M, “Evolution of Client-Side Rendering over Server-Side Rendering,” vol. 3, no. 2, pp. 1–10, 2020.

[12]Vercel, “Data Fetching Overview,” 2022. https://nextjs.org/docs/basicfeatures/data-fetching/index (accessed Jan. 24, 2022).

[13] Modern Front End Web Architectures with React.Js and Next.Js by Mochammad Fariz SyahLazuardy ,DyahAnggraini. http://irjaes.com/wp-content/uploads/2022/02/IRJAES-V7N1P162Y22.pdf