International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Full Stack E-Commerce Application Using Strapi.io and Stripe Technology

Mr. S. S. Nandha Kumar[1], *Mr. P. Anbumani*[2]

[1](M.C.A), [2]M.C.A, M. Phil, NET., Associate Professor,
Department of MCA, Krishnasamy College of Engineering and Technology.

ABSTRACT

In the present age, the vast majority use innovation to carry on with their existences and satisfy their day to day needs. In this age, the vast majority of us use Web based business sites for looking for garments, food, and electronics[1]. The developer has created one Online business web application by utilizing Strapi.io and Stripe innovation as it contains PostgreSQL, Cloudinary cloud, React.JS library, and Node.JS stage[2]. This application is completely practical with various perspectives for client and administrator and likewise has coordinated with installment passage for checkout. Utilizing this site, clients can purchase various sorts of items and pick various styles of items in light of client interests. In this undertaking, the Administrator can add and erase various items[3]. The software engineers have created regulatory capabilities for the site, for example, making an item, making classes, Administrator dashboard, Overseeing items, Oversee classifications. For clients, they can rapidly add their things to the truck. In view of the things in the truck then the bill gets created and the client can pay by utilizing Stripe.

I. INTRODUCTION

Nowadays innovation has turned into a fundamental web based showcasing instrument. Many people are showing interest in purchasing things on the web. In any case, people can see that many little shops and supermarkets are selling their things disconnected. With this sort of selling a large portion of us will confront terrible encounters. For example, in certain shop dealer has the item to sell in the proposition however the purchaser may not be aware of it, or the client might require the item directly then will go to the shop, yet the item is unavailable, all things considered, people will confront a terrible encounter[4]. Besides, in web based shopping clients can choose many items in light of their inclinations and their cost additionally, one can contrast costs likewise from one store with one more by utilizing web based shopping [5]. By experiencing every one of the issues and shortcomings of the disconnected shopping framework, making a Web based business web application is fundamental for looking and shopping in each shop. Nowadays people have seen so many internet business sites made like Flipkart, Amazon, and Myntra, and one can undoubtedly purchase their fundamental items by utilizing these sites. By utilizing these kinds of sites one can purchase their items by remaining in their home. Ultimately, people can see the contrast between the costs of items, and assuming that people see the expense of the item will be marginally higher in disconnected shopping when contrasted with web based shopping. For making these sorts of Online business web applications respond js, Strapi, and Stripe will be the most ideal choice to assist us with making the best and strong web applications[6].

II. PROBLEM STATEMENT

This venture intends to make a web application that will make it simpler to track down fascinating items. This Web based business web application administrator can add classifications like iron box, water purifier, and so on which will draw in clients. Clients additionally can undoubtedly look for their #1 item. The users can likewise get them effectively by adding to the truck and expanding or diminishing by tapping on the "+" sign and "-" sign. Subsequent to adding users can check the aggregate sum of what has been added to the truck. A fruitful installment entryway is empowered to do installments by check card, Visa, and net banking[7].

III. PROPOSED METHODOLOGY

User-centric design: The user interface should be designed with the customer in mind, with intuitive navigation, easy-to-use search functionality, and a streamlined checkout process that minimizes barriers to purchase.

Mobile optimization: The app should be optimized for mobile devices, which are increasingly becoming the primary way that customers access e-commerce apps.

Integrated payment solutions: The app should integrate with a range of payment options, including credit cards and debit cards, to make it easy for customers to make purchases.

By incorporating these features into an e-commerce app, businesses can create a compelling user experience that drives sales, enhances customer loyalty, and differentiates their brand in a crowded market.

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

- Administrator
- Users

1. ADMINISTRATOR MODULE

An administrative module in an e-commerce application typically refers to the back-end or admin panel of the system. The ecommerce application is a web-based interface that allows administrators or store owners to manage various aspects of the e-commerce platform. The administrative module provides tools and functionalities to handle the backend operations and configuration of the online store.

1.1 Authentication and Authorization

Authentication and authorization are fundamental components of the Admin Dashboard Module in a React.js e-commerce application. Here's how authentication and authorization can be implemented:

1.1.1 Authentication

- Implement a user login page where administrators can enter their credentials, such as username/email and password.
- Validate user credentials against a user database or backend server to ensure the entered information is correct.
- Issue a session token or JSON Web Token (JWT) upon successful authentication, which will be used to authenticate subsequent requests.

1.1.2 Authorization

- Define roles and permissions for administrators to determine what actions they can perform within the admin dashboard. For example, roles may include "super-admin," "product manager," "order manager," etc.
- Store role information for each administrator in the user database or backend server.
- When a user logs in, retrieve their role and associated permissions.
- Implement authorization checks on the server side or within the React.js application to verify if a user has the necessary permissions to access or perform specific administrative tasks.

1.2 Admin Dashboard

The dashboard overview functionality in a React.js e-commerce application provides administrators with a consolidated view of key metrics and statistics related to the e-commerce platform's performance. The admin dashboard offers a quick snapshot of the business's health and allows administrators to monitor important data at a glance. Here are the key components and features typically included in the Dashboard Overview.

- **Charts and Graphs:** Utilize visual representations such as line charts, bar graphs, or pie charts to present data in an easily digestible format. Use charts to display trends, comparisons, or distribution of data. For example, the admin can show sales trends over time, revenue distribution by product category, or customer acquisition by marketing channel.
- **Date Range Selection:** Include a date range selector that allows administrators to specify the time period for which they want to view the dashboard data. This enables them to analyze data over specific time intervals and compare performance between different periods.
- **Export and Report Generation:** Allow administrators to export dashboard data or generate reports in various formats, such as PDF or CSV. This facilitates further analysis, data sharing, or presentation to stakeholders.

By providing a comprehensive and visually appealing Dashboard Overview, administrators can quickly assess the e-commerce platform's performance and make informed decisions. Customize the metrics and visualization based on user specific business needs and goals. Continuously evaluate and refine the dashboard based on feedback and evolving requirements.

1.3 Product Management

Product Management is a crucial component of the Admin Dashboard Module in a React.js e-commerce application. Product management module allows administrators to efficiently manage the products available for sale on the platform.

- **Product Listing:** Display a list of all products available on the e-commerce platform. Show essential information for each product, such as the product name, image, price, and stock availability.
- **Product Creation:** Provide a user-friendly interface for administrators to add new products to the platform. Include fields for product details, such as name, description, price, images, variations (e.g., sizes, colors), categories, and other relevant attributes.
- **Product Editing:** Allow administrators to modify existing product details, including the ability to update the product name, description, price, images, variations, categories, and other attributes.
- **Product Search and Filtering:** Implement search and filtering options to help administrators quickly find specific products based on attributes such as name, category, price range, or stock availability.

By implementing a robust Product Management functionality, administrators can effectively manage the products available on the e-commerce platform, ensuring accurate information, streamlined updates, and efficient inventory management. Customize and enhance these features based on customer specific business requirements.

1.4 Product Categories Management

Product Categories play a significant role in organizing and categorizing products in an e-commerce application. Product categories management module helps users navigate the product catalog, improve searchability, and enhance the overall user experience.

- **Category Creation:** Provide an interface for administrators to create new product categories. Include fields for the category name, description, and optional attributes like an image or icon.
- **Category Editing:** Allow administrators to modify existing category details, including the category name, description, and associated attributes. Provide options to update the category hierarchy or move categories within the hierarchy.
- **Product Assigning:** Enable administrators to assign products to one or multiple categories during the product creation or editing process. Provide a user-friendly interface or dropdown menus for category selection.
- **Category Pages:** Generate dynamic category pages that display products belonging to a specific category. These pages can show category-specific banners, descriptions, filters, sorting options, and pagination for a seamless browsing experience

By implementing a robust Product Categories functionality, the admin can organize the product catalog effectively, improve user navigation, and enhance the overall user experience of e-commerce applications.

1.5 Order Management

Order Management is a crucial component of an ecommerce application's admin dashboard. Order management module allows administrators to efficiently manage and process customer orders.

- **Order Listing:** Display a list of all customer orders placed on the e-commerce platform. Show essential information for each order, such as order number, customer name, order status, total amount, and order date.
- **Order Details:** Provide a detailed view of each order, including the purchased products list, quantities, prices, shipping address, billing information, and any additional order-specific details.
- **Order Analytics:** Integrate order analytics to track metrics like order count, revenue, average order value, or order conversion rate. Display this data in the admin dashboard to help administrators analyze order performance and trends.
- **Order Export and Reports:** Allow administrators to export order data or generate reports in various formats, such as PDF or CSV. This enables further analysis, financial reporting, or integration with other systems.

By incorporating a comprehensive Order Management functionality, administrators can efficiently process and fulfill customer orders, ensuring a smooth and satisfactory experience for buyers. Customize and enhance these features based on users specific business requirements, shipping processes, and customer support workflows.

The administrative module plays a crucial role in managing the day-to-day operations and maintaining the backend functionality of an e-commerce application. Order management module empowers administrators to efficiently handle various aspects of the online store, ensuring smooth operation and effective management of the e-commerce business.

2. USERS MODULE

The user module in an ecommerce application refers to the front-end or user-facing part of the system. User module provides a user interface that allows customers to browse products, make purchases, manage their accounts, and perform other actions related to their shopping experience.

2.1 Product search

The Product Search functionality in a React.js e-commerce application allows users to search for specific products based on keywords or criteria. Product search module enables users to find products quickly and efficiently.

- **Search Bar:** Include a search bar component at a prominent location in this application, such as the header or navigation area. The search bar should allow users to enter keywords or search terms.
- **Search Results:** Display search results based on the entered keywords. Show a list of products that match the search query, including product names, images, prices, and other relevant information.
- **Pagination:** If the search results contain a large number of products, implement pagination to display a limited number of products per page. Allow users to navigate through different pages of search results.

2.2 Product details

The Product Details functionality in a React.js e-commerce application provides users with comprehensive information about a specific product. Product details module allows users to view detailed product descriptions, images, specifications, pricing, and other relevant details.

- **Product Description:** Show a detailed description of the product, including its features, benefits, and any additional information that helps users understand the product's value proposition.
- **Add to Cart:** Include an "Add to Cart" button or icon that allows users to add the product to their shopping cart directly from the product details page.
- **Related Products:** Show a section featuring related or recommended products that complement or are similar to the current product. This can help users discover additional products they might be interested in.

2.3 Add to cart

The "Add to Cart" functionality in a React.js e-commerce application allows users to add products to their shopping cart or bag. Add to cart module enables users to collect and store items for purchase before proceeding to the checkout process.

- **Cart State:** Maintain a cart state within the application, either as a local state or using a state management library like Redux or React Context. The cart state should store the selected products, their quantities, and any other relevant information.
- **Product Quantity:** Allow users to specify the quantity of the product they want to add to the cart. This can be done through increment/decrement buttons.
- **Cart UI:** Display a visual representation of the cart, typically as a mini cart or a dedicated cart page. The UI of the cart shows the added products, their quantities, prices, and a subtotal. Update the cart UI dynamically as users add or remove products.
- **Cart Summary:** Provide a summary of the cart's contents, including the total number of items in the cart and the total price. Update the summary whenever the cart is modified.
- **Remove from Cart:** Include a "Remove" or "Delete" option for each product in the cart UI, allowing users to remove unwanted items from their cart.
- **Quantity Update:** Enable users to update the quantity of products in the cart. Update can be done through input fields, increment/decrement buttons, or other intuitive controls.
- **Cart Badge:** Display a visual indicator, such as a cart icon with a badge, to show the number of items currently in the cart. Update the badge dynamically as items are added or removed.
- **Cart Checkout:** Provide a prominent "Checkout" button or link that redirects users to the checkout process when they are ready to complete their purchase.

By implementing the "Add to Cart" functionality, the programmer enables users to collect and manage products for purchase in React.js e-commerce application. This functionality is a crucial step towards creating a seamless shopping experience for the users.

2.4 Cart checkout

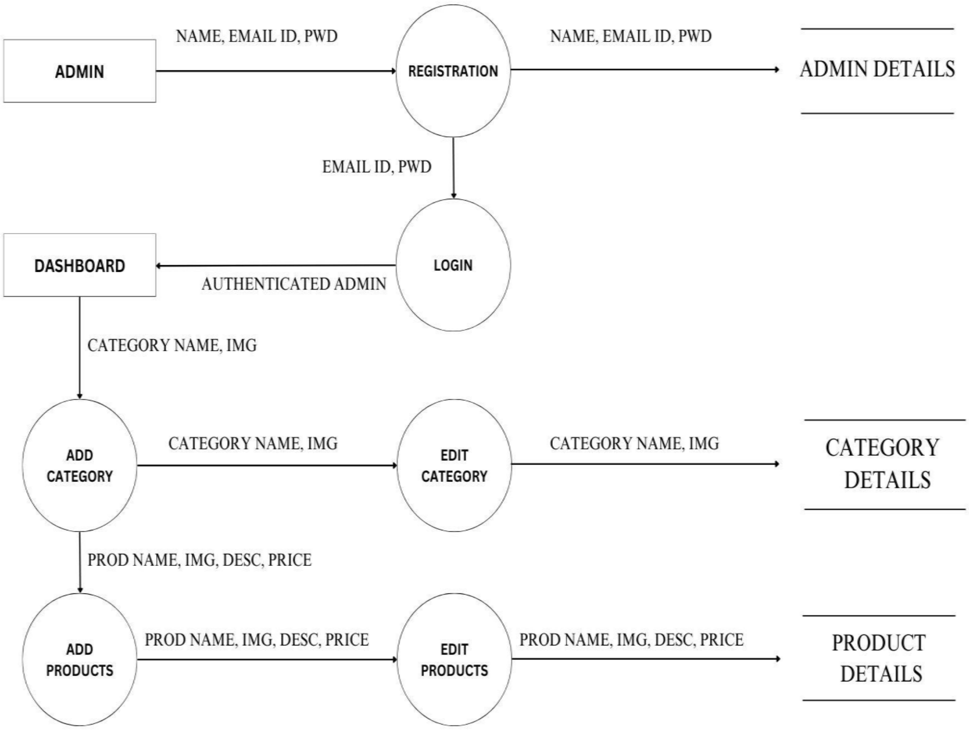
The Cart Checkout functionality in a React.js e-commerce application allows users to review their selected items, provide shipping and billing information, and proceed with the purchase. Cart checkout module facilitates the final steps of the shopping process.

- **Checkout Page:** Create a dedicated Checkout page where users can review and finalize their purchase. Checkout page typically displays the selected items, their quantities, prices, and a subtotal.
- **Shipping Information:** Collect shipping information from the user, including the shipping address, contact details, and preferred shipping method. Provide input fields or form components to gather this information.
- **Billing Information:** Gather billing information if it differs from the shipping information. Include fields for the billing address, payment method selection, and any other relevant details.
- **Order Summary:** Display a summary of the order, including the selected items, their quantities, prices, and the total order amount. Include any applicable taxes, discounts, or shipping costs.
- **Payment Integration:** Integrate with a payment gateway or payment service provider to securely process the user's payment. Payments include credit card processing and the debit card payment methods. Implement the necessary form components or APIs to handle payment submission.
- **Order Confirmation:** Display an order confirmation message or page after the payment is successfully processed.
- **Guest Checkout:** Offer the option for users to complete their purchase as a guest, without requiring them to create an account. However, consider providing incentives for users to create an account, such as order history, faster checkout for future purchases, or loyalty program benefits.
- **Error Handling:** Implement error handling for scenarios such as payment failures, incomplete form submissions, or invalid input. Display appropriate error messages to guide users in resolving any issues.

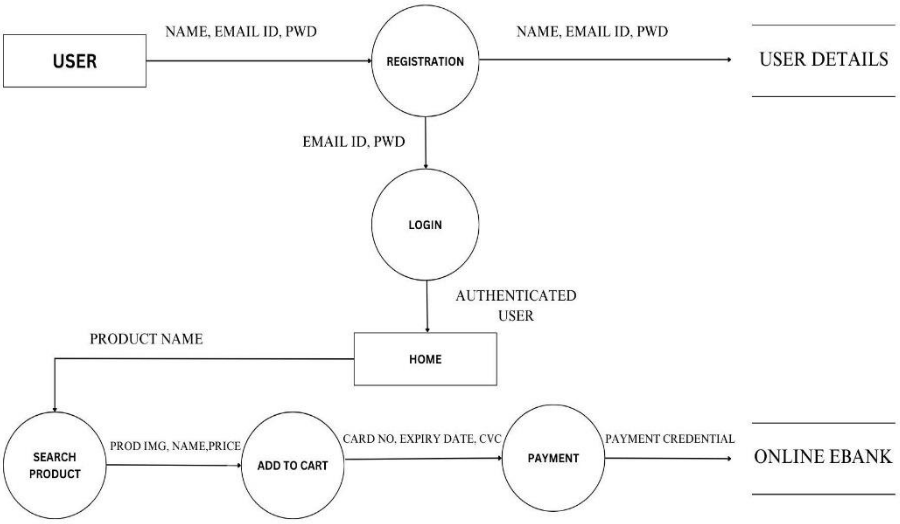
By implementing a well-designed and intuitive Cart Checkout functionality, programmer provide users with a seamless and secure way to complete their purchase in e-commerce application.

IV. DATA FLOW DIAGRAM

1. ADMIN DATA FLOW DIAGRAM



2. USER DATA FLOW DIAGRAM



V. CONCLUSION

The framework has been created with much consideration and liberated from mistakes and simultaneously it is proficient and less tedious. The motivation behind this undertaking was to foster a web application for buying things from a shop. This task helped us in acquiring important data and commonsense information on a few subjects like planning site pages utilizing html , css and reactJS, utilization of responsive formats, planning of web applications, and the board of strapi application. The whole framework is gotten. Likewise this undertaking assists the developer with understanding about the improvement periods of a venture and programming improvement life cycle. The programmer figured out how to test various elements of a task. This undertaking has given us extraordinary fulfillment in having planned an application which can be executed to any nearby shops or marked shops selling different sorts of items by straightforward alterations.

VI. FUTURE ENHANCEMENT

Further various highlights can be added to this framework in future like giving clients more command over orders so every client can keep up with their own dashboard. One more element the developer wished to carry out was giving classes to clients so various offers can be given to each class. Framework might monitor the historical backdrop of acquisition of every client and give ideas in view of their set of experiences and furthermore this application would be created in portable variant moreover. These highlights might have been executed except if time didn't restrict us.

VII. REFERENCES

[1] Chanana, N., & Goele, S. (2012). Future of e-commerce in India. International Journal of Computing & Business Research, 8.

[2] Mai, N. (2020). E-commerce Application using MERN stack.

[3] Ullah, S. E., Alauddin, T., & Zaman, H. U. (2016, January). Developing an E-commerce website. In 2016 International Conference on Microelectronics, Computing and Communications (MicroCom) (pp. 1-4). IEEE.

[4] Nemat, R. (2011). Taking a look at different types of e-commerce. World Applied Programming, 1(2), 100-104.

[5] King, D. N., & King, D. N. (2004). Introduction to e-commerce. Prentice Hall.

[6] Niranjanamurthy, M., Kavyashree, N., Jagannath, S., & Chahar, D. (2013). Analysis of e-commerce and m-commerce: advantages, limitations and security issues. International Journal of Advanced Research in Computer and Communication Engineering, 2(6), 2360-2370.

[7] Nagothu Diwakar Naidu, Pentapati Adarsh, 2021. E-Commerce web Application by using MERN Technology

[8] Hoque, S. (2020). Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node. js. Packt Publishing Ltd.

[9] D. Laksono, ”Testing Spatial Data Deliverance in SQL and NoSQL Database Using NodeJS Full Stack Web App,” 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, 2018, pp. 1- 5.

[10] Bo Zhang, Ruihan Yong, Meizi Li, Jianguo Pan, Jifeng Huanglaa, "A Hybrid Trust Evaluation Framework for Ecommerce in Online Social Network" 2169-3536 (c) 2016 IEEE.

[11] IEEE.Dyl, T. and Przeorski, K., 2017. Mastering Full-Stack React Web Development. Packt Publishing.

[12] Ambler, T. and Cloud, N., 2015. Javascript Frameworks For Modern Web Dev. Apress.

[13] Bin, B.S. and Park, J.K. (2002), "An empirical study on the success factors of a small business startingup", The Asia Pacific journal of Small Business, Vol. 24 No. 3, pp. 135-158

[14] Nodejs.org. 2020. Nodejs. Accessed on 28 April 2020 [online] Available at: https://nodejs.org/

[15] Jyoti Shetty, Deepika Dash, Akshaya Kumar Joish, Guruprasad C "Review Paper on Web Frameworks, Databases and Web Stacks" 2020 IRJET-V7141078

[16] Gu, J., & Lee, Y. (2018). Development of a Secure E-commerce Platform using MERN Stack. International Journal of Security and Its Applications, 12(4), 153-166.

[17] Kumar, S., & Bhushan, B. (2019). A Study of E-commerce and Its Security Issues. International Journal of Advanced Research in Computer Science, 10(3), 149-152.

[18] Lee, J., Lee, M., & Kim, Y. (2020). User Experience of E-commerce Auction Systems: An Empirical Study of Seller Perspectives. International Journal of Human-Computer Interaction, 36(5), 414-4