My Store: MERN Stack eCommerce App with Order Summary and Invoice

Pallavi Govind Rao[1], Sheetal Singh[1], Neha Singh[1], Anshika Pandey[1] and Mr. Mudit Dubey[2]
([1]Student, Dept. of BCA) & ([2]Assistant Professor, Dept. of BCA)
(Digvijai Nath Post Graduate College, Gorakhpur)

ABSTRACT
In general, eCommerce sites create an invoice and email it to the user after order completion or delivery, which is confusing and a poor post-sales experience. In this paper, we present "My Store", a full-stack e-commerce website built with the MERN stack (MongoDB, Express.js, React.js, and Node.js), which eliminates this issue by offering an Order Summary at the time of order placement. The platform provides product search and category filtering, user authentication (signup and login), cart management, address-based order checkout with Cash-on-Delivery (COD) support, and an admin dashboard with complete product management (add/edit/update/delete), all with secure login. Once the order has been placed, the user can view and download the Order Summary in a neatly-formatted PDF document or continue browsing products. The application uses the MongoDB database to store user, product, cart, address and order collections. The product provides transparency in real-time, reduces order related support calls and enhances user trust and confidence.

KEYWORDS: MERN Stack, eCommerce, Order Summary, Invoice Download, React.js, Node.js, MongoDB, Admin Panel, User Authentication, PDF Generation.

I. INTRODUCTION

The rise of e-commerce has resulted in changes in the way consumers discover, evaluate and purchase products. With millions of orders being processed daily for businesses of various sizes, communication after an order is placed is a key factor in creating and retaining trust. However, most eCommerce applications do not offer good order confirmation until after an order has been processed or dispatched, leaving their customers hanging.

In this paper we present "My Store" eCommerce web application, built on MERN stack, to address this issue. It allows browsing of products, user registration and login, shopping cart, checkout, managing addresses and secure order placement. The key feature is the immediate generation of the Order Summary when placing an order, and its download as a PDF to give the user instant feedback about the order.

It also has a secure admin login page (with a hidden URL) where the store owner can add, edit, update and delete products. It is built using Node.js/Express.js for the back end, React.js for the front end and a database - MongoDB with four major collections: users, products, carts, addresses and orders.

II. LITERATURE REVIEW

Nielsen [1] identified ten heuristics of user interface design for interactive systems, including "visibility of system status", which can be applied to e-commerce checkout. This means that the system should always keep users informed about what is going on, through appropriate feedback within reasonable time. Immediate post-purchase notifications are an example.

Alrawabdeh [2] investigated the factors that influence the acceptance of eCommerce websites and concluded that trust in eCommerce is especially low in purchase and post-order stage. This study identifies the need for instant post-purchase confirmation to alleviate purchase-related anxiety and facilitate repeat transactions.

Gangeshwer [3] analysed the e-commerce buying process and identified that the post-checkout stage is not well addressed. The post-checkout page is usually a generic "Thank You" page with little or no information, which does not align with the users' expectations for post-order confirmation.

Kumar and Singh [4] studied client-side document generation libraries. They concluded that the jsPDF library is effective for generating and downloading invoices in the browser, avoiding the time-consuming server-side rendering of PDFs.

Turban et al. [5] examined best practices for designing e-commerce systems. They proposed that real-time, automatic invoice generation improves the user experience and is essential to minimise disputes and customer support calls.

Flanagan [6] studied the architecture of modern single-page applications (SPAs) based on React.js and noted that component-based construction enables highly modular post-transaction views, such as the order summary panel display, to be generated and updated with the fewest page reloads.

III. SYSTEM DESIGN AND METHODOLOGY

3.1 Technology Stack

My Store is built with the MERN stack, a common framework for creating full-stack JavaScript applications. That stack is composed of MongoDB (NoSQL database), Express.js (web framework), React.js (user interface library) and Node.js (JavaScript runtime). This enables a unified development environment for work on the client and server sides, easy data management, and a short learning curve for developers.

3.2 System Architecture

This application has three layers. This layer is built with React.js and serves as the presentation and interaction layer, managing components. The Application Logic Layer is a RESTful API built with Express.js on Node.js and is responsible for business rules, authentication, and order management. The Data Layer is a MongoDB database with collections users, products, addresses, carts and orders. The client and the server communicate via a RESTful API implemented with Express.js on Node.js.

3.3 Key Features and Modules

The application has the following modules:

- Product Listings Module: Home page shows a list of products. The user can search for products in the search box (by keywords) or view them by category. The user can navigate to the product details page by clicking on a product.
- User Authentication Module: Users can sign up on the Signup page. Users can log in on the Login page. Users can log in via sessions, using JSON Web Tokens (JWT). The cart is only accessible to logged-in users.
- Cart: The user can add items to the cart. All products in the cart will be listed on the cart page, with product images, names, prices, quantities (plus and minus buttons), and remove buttons. The cart will also display the total, and users can click on Proceed to Checkout to place an order.
- Add/Select Address and Checkout Module: The user will be prompted to add/select the address. This will appear on the checkout page, along with the subtotal and the Place Order (COD) button.
- Order Summary and PDF Download Module: After placing an order, the user will be taken to the Order Summary page. This will display the Order ID, status, total, address and products. By clicking on the Download Order PDF button, you can download a PDF version (using the jsPDF library). You can click on Continue Shopping
- Admin Panel Module: There's a hidden URL to access the admin panel that is only accessible by admin users. The admin panel has a Product List that can be added, edited, or deleted. This is a user hidden panel.

3.4 Database Design

Data is stored in five collections in MongoDB. The users' collection has name, email, bcrypt-encrypted passwords and dates. The product collection consists of a name, description, price, quantity, category, and image. The addresses collection has the user ID and the shipping address. The carts collection holds the user ID and an array of products and quantities. The orders collection includes an order ID, user ID, an array of items, the order subtotal, user address, and order status. MongoDB's flexible schema makes it easy to loop through and map JavaScript objects.

IV. RESULTS AND DISCUSSION

We implemented and tested the My Store app across all scenarios. We have included screenshots of the major screens.

4.1 Cart Page

My Store's cart page is shown in Figure 1 below. This displays the individual items the customer has selected, their images, names, prices, an update to change the quantity, and a remove button. At the bottom, the total is shown. By clicking on the Proceed to Checkout button, the user should go to the next step.

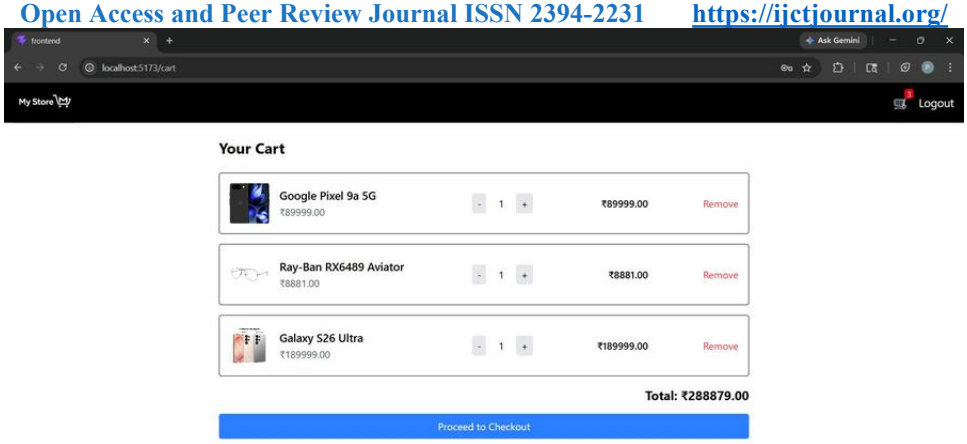

Fig. 1 – Cart Page of My Store

4.2 Checkout Page

The Checkout page is shown in Figure 2. It shows the current delivery address and a radio button. The Order Summary displays the total order amount, and the Place Order (COD) button makes an API call to place the order.

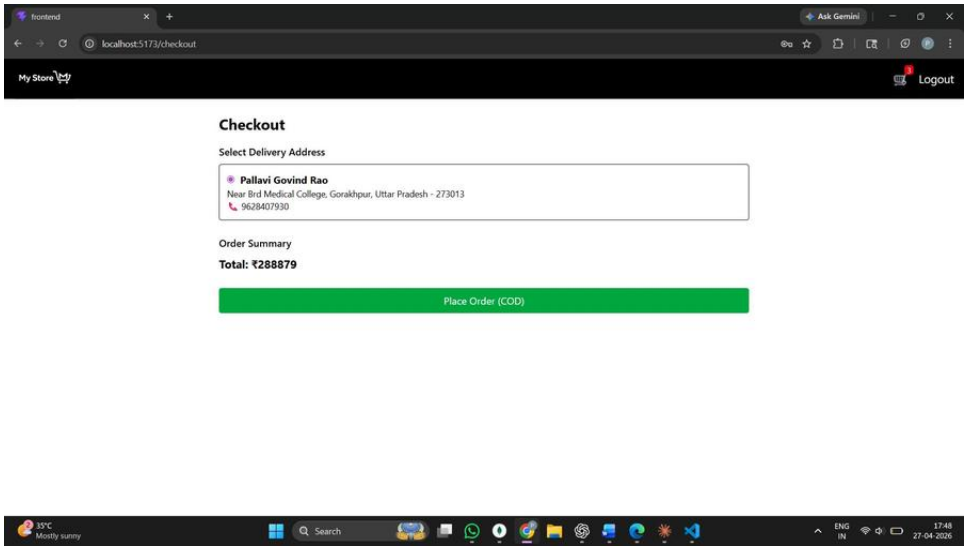

Fig. 2 - Checkout with Address and Order Total

4.3 Order Summary Page (Instant Confirmation)

Fig. 3 shows the research contribution: an instant Order Summary page displayed after a successful order. It displays the Order ID, Order Status (Placed), Grand Total, Shipping Address and the products that have been ordered. There's also a green button labelled "Download Order in PDF".

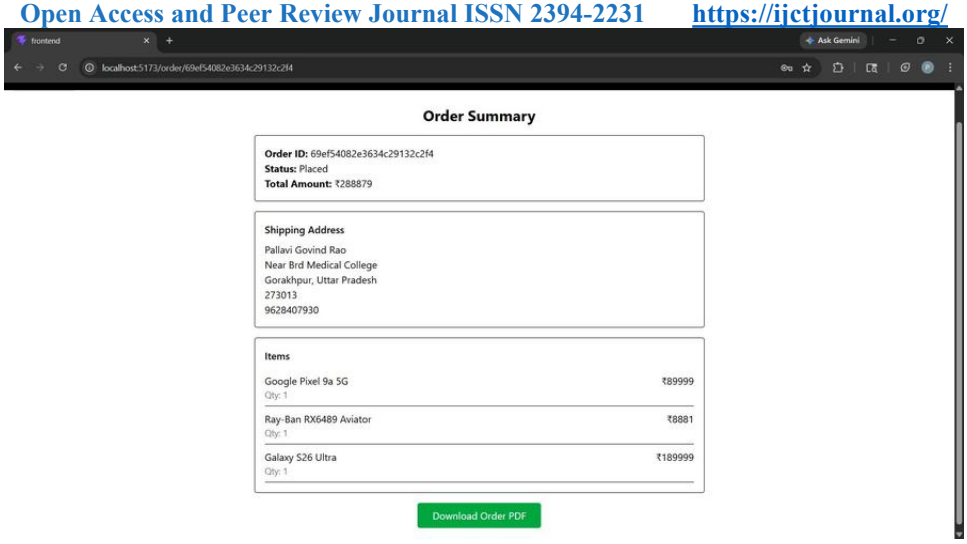

Fig. 3- Order Summary (Instant) with PDF download

4.4 Downloaded PDF Invoice

Fig. 4 - Downloaded Order PDF (Button Download Order PDF). The PDF file is generated on the client side (no round-trip to the server) using the jsPDF library. It contains the My Store logo, Order ID, date, a table of product items, quantities, and prices, subtotal, shipping, total, and address. It is nicely laid out and can be saved/printed.

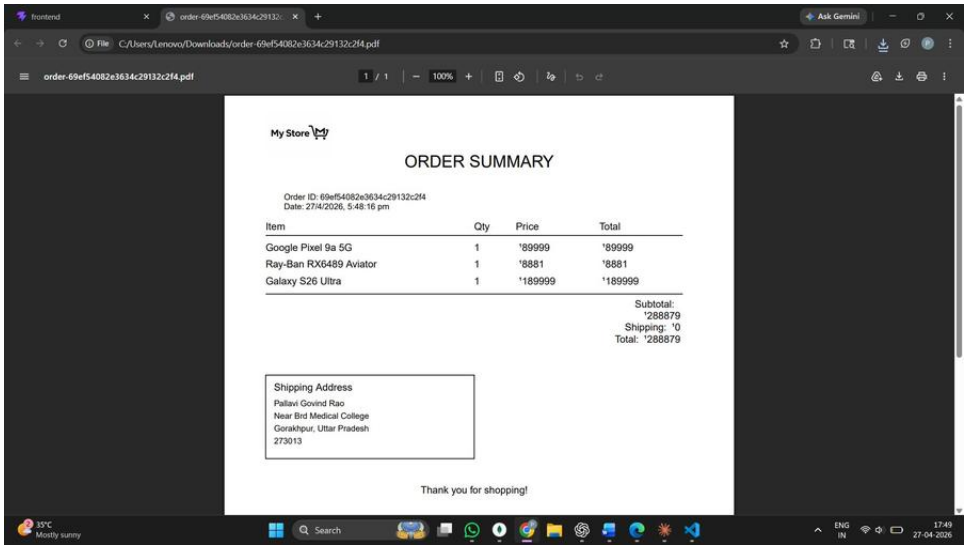

Fig. 4 – PDF Order Summary

4.5 Admin Panel – Add a Product

The Admin Panel can be accessed by a secret URL, and is illustrated in Figure 5. This page displays a Product List table with product names, prices, and stock levels. The Add New Product button is used to create a new product; the Edit button is used to edit a product; and the Delete button is used to delete a product.

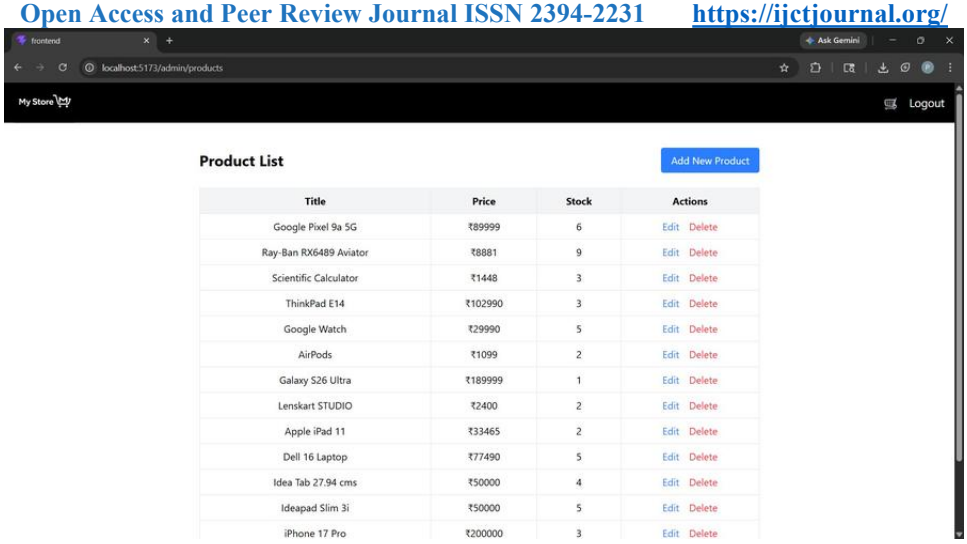

Fig. 5 – Product List

4.6 MongoDB Database – Users

In the figure below, we show the user collection from the MERN-ecommerce database in MongoDB Compass. We have user documents in our collection with an auto-generated ObjectId, name, email, bcrypt-hashed password, created date and last updated date. We also have other collections for products, orders, carts and addresses.

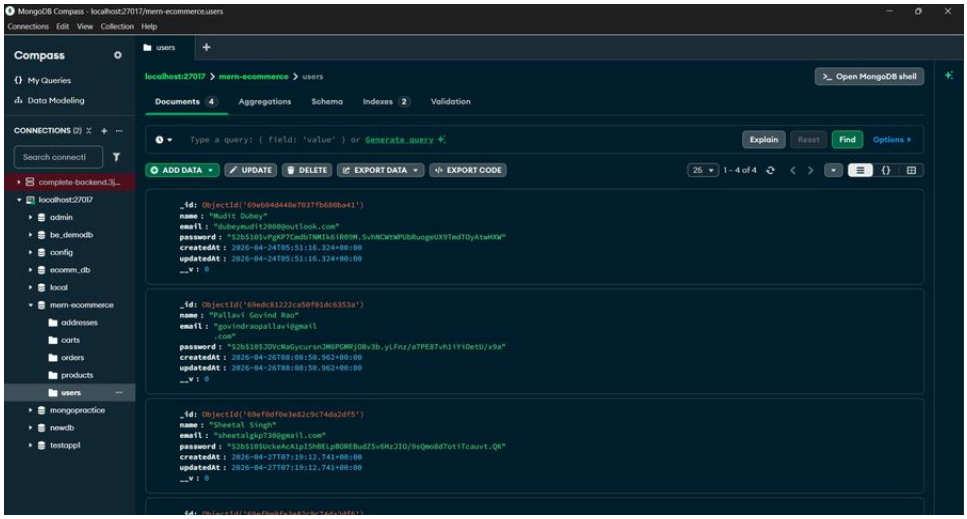

Fig. 6 – MongoDB Compass: mern-ecommerce → users Collection

4.7 Discussion

The system fulfils all intended objectives. The Order Summary page renders within milliseconds of order placement, as the data is returned directly. The system meets all the objectives. The Order Summary page is rendered in milliseconds when placing an order, as the data is sent in the API response and rendered by the React component, without additional queries. The PDF is generated in less than 200 milliseconds on the client side because jsPDF is used. The admin panel is secure due to the route protection, and only admin users can update products. MongoDB's object storage was suitable for the document objects in the orders and cart collections.

V. CONCLUSIONS

My Store, an e-commerce web app developed using the MERN stack, has been proposed to address a common user-interaction problem in e-commerce. The Order Summary, automatically generated when an order is placed and the option to download as PDF, gives immediate order confirmation, rather than the user having to wait for order processing or delivery, as is the case with current solutions.

It has been possible to implement user login and registration successfully, product listing (with search and filters), cart, checkout (based on address), cash-on-delivery (COD) orders, instant order confirmation, and a secure admin panel to manage products. The data is managed in MongoDB, which has five organised collections.

The next step is to enable online payments (Razorpay/Stripe), support sending an email to the user with the invoice (PDF) after placing the order, support order tracking, add a dashboard for the admin with analytics, and make the app a a progressive web app (PWA) for mobile.

REFERENCES

[1]. Nielsen, J. (1994). Usability Engineering. Morgan Kaufmann Publishers, San Francisco, CA.

[2]. Alrawabdeh, W. (2014). The Internet and e-Commerce Use and Adoption in Jordan. The Electronic Journal of Information Systems in Developing Countries, 61(1), 1-18.

[3]. Gangeshwer, D. K. (2013). E-commerce or Internet Marketing: Business View from India. International Journal of u- and e-Service, Science and Technology, 6(6), 187-194.

[4]. Kumar, A. and Singh, P. (2020). Generating Documents in Web Applications: jsPDF and PDFMake. International Journal of Web Engineering and Technology, 15(2), 112-128.

[5]. Turban, E., King, D., Lee, J., Liang, T., and Turban, D. (2015). Electronic Commerce: A Managerial and Social Networks Perspective (8th ed.). Springer.

[6]. Flanagan, D. (2020). The Definitive Guide to JavaScript (7th ed.) O'Reilly Media.

[7]. MongoDB Inc. (2022). MongoDB Documentation. https://www.mongodb.com/docs/

[8]. React Documentation. (2022). React - JavaScript library for building user interfaces.

[9]. Node.js Foundation. (2022). Node.js Documentation. https://nodejs.org/en/docs/

[10]. parallax/jsPDF. (2022). Create PDFs in JavaScript.