Project Report

1. INTRODUCTION

ShopEZ is a full-stack e-commerce application built using the MERN stack (MongoDB, Express, React, Node.js) as part of the Smart Bridge project. This application provides a seamless shopping experience for users, allowing them to browse, filter, and purchase products, as well as manage orders. It includes both user and admin functionalities.

1.1 Project Overview:

ShopEZ is a full-featured e-commerce web application developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The platform is designed to deliver a seamless and efficient online shopping experience for customers while offering robust tools for sellers and administrators. ShopEZ enables users to browse, search, and purchase products with ease, while sellers can manage their inventory and track orders through an intuitive dashboard. Administrators are provided with powerful controls to oversee platform activity, ensuring a secure and well-regulated marketplace.

1.2 Purpose

The primary purpose of ShopEZ is to create a scalable, responsive, and user-centric e-commerce solution that caters to the needs of all stakeholders—customers, sellers, and administrators. By leveraging modern web technologies, ShopEZ aims to bridge the gap between functionality and user experience, ensuring a smooth shopping process, efficient seller operations, and simplified administrative management. The project also serves as a practical implementation of modular development and real-time interactivity within a modern online marketplace.

2. IDEATION PHASE

2.1 Problem Statement

In today’s fast-paced digital world, many e-commerce platforms struggle with either limited functionality for sellers or poor user experience for customers. Small to mid-level sellers often lack access to streamlined tools for managing inventory and tracking sales. Meanwhile, customers face issues such as cluttered interfaces, confusing checkout processes, and lack of trust in platform security. Additionally, administrators find it difficult to maintain order and oversee platform integrity effectively. There is a need for a comprehensive, scalable, and user-friendly platform that caters equally to buyers, sellers, and administrators.

2.2 Empathy Map Canvas

To better understand the needs and behaviors of our platform's users, an Empathy Map Canvas was created for three primary personas: **Customer**, **Seller**, and **Admin**. This tool helps identify user goals, frustrations, and motivations, providing valuable insights to improve the platform’s usability and functionality.

1. Customer

Says	Thinks

Says	Thinks
“I want to find what I need quickly.”	“Are these products reliable and affordable?”
“This checkout process should be easy.”	“I hope my order is secure and arrives on time.”

Does	Feels
Browses product catalog and adds items to cart	Frustrated by complex or slow checkout
Reads reviews and compares prices	Confident when product info and pricing are clear

Needs:

- Easy navigation and product discovery
- Secure and fast checkout
- Order tracking and confirmation

Pain Points:

- Lack of product information or reviews
- Complicated checkout processes
- Concerns over payment security and delivery times

2. Seller

Says	Thinks
“I want my products to be seen.”	“How can I track my sales easily?”
“Managing listings should be simple.”	“Is the platform helping me reach customers?”

Does	Feels
Uploads products and monitors orders	Overwhelmed if platform management is unclear
Manages inventory and promotions	Empowered when tools are intuitive and helpful

Needs:

- Easy-to-use product management tools
- Sales and performance analytics
- Reliable customer reach and visibility

Pain Points:

- Limited promotional tools
- Poor backend usability
- Lack of real-time feedback on sales

3. Admin

Says	Thinks
“I need to ensure everything runs smoothly.”	“Are users getting a secure, seamless experience?”
“I want to manage products and users efficiently.”	“How can I improve platform stability?”

Does	**Feels**
Manages categories, banners, and users	Responsible for uptime and platform security
Reviews user activity and oversees orders	Pressured to handle issues quickly and securely

Needs:

- Full access to product and user management
- Tools to monitor and analyze platform activity
- Secure, stable backend operations

Pain Points:

- Complex or outdated admin interface
- Limited visibility into user behaviors
- High responsibility for system uptime and data security

2.2 Brainstorming

In the brainstorming phase, various ideas were generated to shape ShopEZ into a user-friendly and efficient e-commerce platform. We explored the core needs of buyers, such as smooth product browsing, secure checkout, order tracking, and intuitive user flow. At the same time, we identified administrative requirements like product management, user monitoring, and overall control of the platform.

We mapped out potential features including a dynamic product catalog, cart system, simplified order placement, and a secure authentication system. Emphasis was placed on creating a responsive UI, ensuring data consistency through MongoDB, and implementing seamless communication between frontend and backend using RESTful APIs. This stage helped align the team's vision and establish a clear development roadmap for ShopEZ.

Grouped Features:

- User Module: Registration, login, cart, checkout, search, order tracking
- Seller Module: Product listing, inventory dashboard, order fulfillment
- Admin Module: User/seller control, analytics, content moderation
- External: Paypal, MongoDB Atlas.

3. EQUIREMENT ANALYSIS

3.1 Customer Journey map

- **Awareness**: User lands on ShopEZ via search, referral, or direct visit.
- **Browsing**: Explores products using filters and sorting options.
- **Selection**: Adds product to cart and reviews selected items.
- **Cart Management**: Manages cart items before finalizing purchase.
- **Checkout**: Enters details and completes a secure transaction.
- **Order Confirmation**: Receives confirmation and order summary.
- **Post-Purchase**: Tracks delivery and order status updates.
- **Feedback & Support**: Leaves feedback or seeks support if needed.

3.2 Solution Requirement

- Simplify homepage layout with clear categories and navigation.
- Improve product filters for speed and usability.
- Display detailed product info with images and reviews.
- Enable easy cart item editing with clear pricing.
- Optimize checkout with auto-fill and multiple payment modes.
- Send real-time order confirmations via screen, email, and SMS.
- Integrate live order tracking and status updates.
- Provide fast customer support with chatbot and quick help options.

3.3 Data Flow Diagram

- User logs in and sends credentials to the authentication system.
- System verifies login and redirects to the homepage.
- Homepage fetches product data from the product database.
- User views product listings retrieved from the server.
- User applies filters; system queries products by selected categories.
- Filtered product list is displayed to the user.
- User selects product and adds it to the cart.
- Cart stores product details linked to the user session.
- User enters address; system saves it to the order database.
- On checkout, payment details are sent to PayPal API.
- PayPal processes payment and sends confirmation.
- System creates an order record and stores transaction info.
- User receives confirmation of successful order placement.

3.4 Technology Stack

- **Frontend:** React.js, Tailwind CSS, Redux Toolkit
- **Backend:** Node.js, Express.js

- **Database:** MongoDB Atlas (cloud-hosted NoSQL)
- payments, Cloudinary for file storage.

4. PROJECT DESIGN

4.1 Problem Solution Fit

Our solution addresses the identified problem by enabling users to quickly browse, select, and purchase items with real-time inventory and fast order processing. Admin manage their stock with ease, and admins oversee all system interactions,add products and remove products.

4.2 Proposed Solution

The proposed solution features three roles:

- **User Portal:** Browse, filter, search, add to cart, and purchase products.
- **Admin Panel:** Access system data, review user activity, manage performance

4.3 Solution Architecture

React frontend fetches and posts data via REST APIs.

Express backend handles business logic, session tokens, and validations.

MongoDB stores user, product, and order data.

Paypal processes payments using secure callbacks.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The development of **ShopEZ** followed a structured and phased approach to ensure smooth progress and timely completion. The planning process involved breaking down the entire project into manageable stages, each with specific goals and deliverables. Agile methodology was used to accommodate iterative development and incorporate feedback efficiently.

The project was divided into the following key phases:

- **Requirement Gathering:** Identifying user needs, defining features, and preparing design documents.
- **Design Phase:** Creating wireframes, mockups, UI components, and overall system architecture.
- **Development Phase:** Building the frontend and backend modules, setting up the database, and implementing APIs.
- **Integration & Testing:** Connecting modules, performing functionality tests, fixing bugs, and ensuring security.
- **Deployment:** Hosting the application on a cloud platform and ensuring accessibility.
- **Documentation & Final Review:** Preparing technical documentation and project reports, and reviewing the overall performance.

Each phase had predefined timelines and responsibilities assigned, ensuring accountability and consistent progress throughout the development cycle.

5. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing for **ShopEZ** was conducted to ensure that the platform functions efficiently under various conditions and maintains a smooth user experience even during peak loads. The primary goals were to assess response time, system stability, and resource usage.

Key areas tested:

- **Page Load Speed:** All core pages (Home, Product Listing, Cart, Checkout) were optimized for fast loading using React's virtual DOM and efficient component rendering.
- **API Response Time:** Backend endpoints were monitored to maintain quick responses under different data loads using optimized queries and Express middleware.
- **Database Operations:** MongoDB queries were indexed and optimized to ensure faster data retrieval and minimal latency in fetching user, product, and order data.
- **Concurrent Users:** The platform was tested with multiple simulated users to check consistency in performance and identify any bottlenecks or slowdowns.

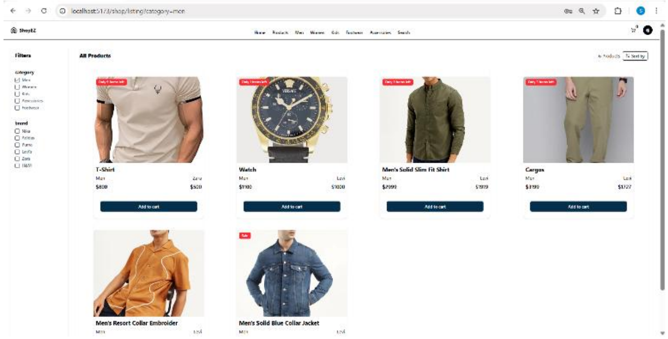
Results indicated that the system consistently handled user actions like browsing, adding to cart, and placing orders within acceptable response times. Performance remained stable, ensuring reliability and scalability for real-world use.

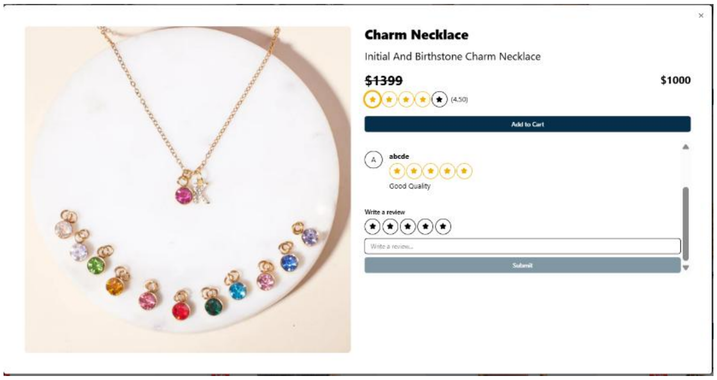
Performance monitoring

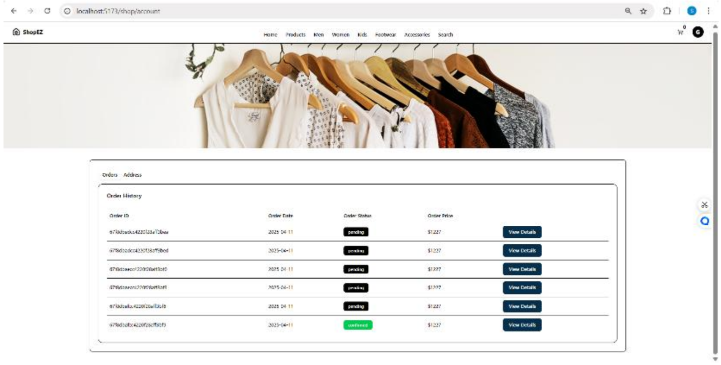
- Google DevTools used to measure initial page loads
- Backend monitored using logs and response time tracking
- Checked memory usage, CPU load during peak simulations

6. RESULTS

6.1 Output Screenshots






localhost:5173/shop/account
ShopEZ
Home Products Men Women Kids Footwear Accessories Search
Orders Address

Order History

Order ID	Order Date	Order Status	Order Price	
67f8dbadcc4220f28af73baa	2025-04-11	pending	$1227	View Details
67f8dbadcc4220f28aff9bed	2025-04-11	pending	$1227	View Details
67f8dbaecc4220f28aff3bf0	2025-04-11	pending	$1227	View Details
67f8dbaecc4220f28aff3bf3	2025-04-11	pending	$1227	View Details
67f8dbafcc4220f28aff3bf6	2025-04-11	pending	$1227	View Details
67f8dbafcc4220f28aff3bf9	2025-04-11	confirmed	$1227	View Details

7. ADVANTAGES & DISADVANTAGES

Advantages:

- User-Friendly Interface: Built using React.js for a responsive and intuitive user experience.
- Secure Checkout Process: Ensures data security through encrypted transactions and validation.
- Efficient Product Browsing: Fast product retrieval and display using optimized MongoDB queries.

- Real-Time Updates: Cart and order information are updated instantly for better user interaction.
- Admin Management: Admins can manage users, products, and monitor activities effectively.

Disadvantages:

- No Seller Module: Current version lacks features for sellers to manage their own product listings.
- Dependent on Internet Connection: As with most web apps, functionality depends on stable connectivity.
- Limited Scalability in Free Hosting: Hosting on free-tier platforms may not handle heavy traffic effectively.

8. CONCLUSION

ShopEZ is a dynamic and user-friendly e-commerce platform built on the powerful **MERN stack** (MongoDB, Express.js, React.js, Node.js). Designed to deliver a seamless shopping experience for users while providing a robust backend system for sellers and administrators, ShopEZ ensures scalability, security, and performance across all its features.

The platform efficiently serves three core user roles: **customers**, **sellers**, and **administrators**. Customers enjoy an intuitive interface for browsing products, adding items to their cart, and completing secure checkouts. Sellers are equipped with a dedicated dashboard that simplifies product management, inventory tracking, and order processing, allowing them to scale their operations with ease. Administrators are given powerful tools to oversee platform activity, manage user accounts, and maintain the integrity of product listings—ensuring a safe and efficient marketplace for all.

ShopEZ showcases the strength of modern web technologies in building scalable and engaging digital commerce solutions. With its modular architecture, thoughtful design, and focus on user experience, it lays a solid foundation for ongoing innovation and future expansion.

In summary, **ShopEZ not only addresses the current demands of the e-commerce ecosystem but also sets the stage for continued growth—enhancing both the shopping experience and the management capabilities for all users involved.**

9. FUTURE SCOPE

- Seller Dashboard Integration: Adding seller-side functionality for inventory and order management.
- Payment Gateway Integration: Enabling real-time transactions using services like Razorpay or Stripe.
- Wishlist and Review System: Allowing users to save products and share feedback.
- Mobile App Version: Extending platform accessibility through a dedicated mobile app.
- AI-Based Recommendations: Implementing product suggestions based on user behavior.

10. APPENDIX

GitHub

https://github.com/SnehaYadavWorld11/Mern_Stack-ShopEZ

Project Demo Link

https://drive.google.com/file/d/1sf0OhwgIi3d6qpX8PHjiiuWHaaonJbmT/view?usp=sharing

https://drive.google.com/file/d/1znDPKbi1CsA1oLESelnp5fbzPIrtxCfw/view?usp=sharing