PROJECT REPORT

ShopEZ:One-Stop Shop for Online Purchases

Team Name - Team ShopVerse

Team Members:

Gorrela Radhika Devi	23MH1A05G8 (Team Leader)
Palla Nani	23MH1A05D2
Bathina Reena	23MH1A05Q0
Bonthu Sandeep	23MH1A05S6

INTRODUCTION:

Project Overview

ShopEZ is a **MERN-based e-commerce web application** designed to simulate a real-world online shopping system. It is divided into two main sections: the **User Section** and the **Admin Section**.

User Section Features:

User Authentication: Sign up, login/logout with JWT-based security.

Product Browsing: View all products with filters (category, price, etc.).

Product Details: Detailed page with product image, description, price, and reviews.

Cart Functionality: Add/remove products and update quantities.

Checkout Process: Place orders with order confirmation.

User Profile: View order history and manage account settings.

Admin Section Features:

Admin Dashboard: Overview of all users, products, and orders.

Product Management: Add, edit, or delete products.

Order Management: View and update order status.

User Management: Manage user roles and access.

Technologies Used:

Frontend: React.js (with HTML, CSS for styling)

Backend: Node.js + Express.js

Database: MongoDB (via Mongoose)

Authentication: JWT + Bcrypt.js

API Communication: RESTful APIs

Purpose of the Project:

The primary purpose of **ShopEZ** is to build a **fully functional, user-friendly online shopping platform** where users can easily browse, search, and purchase products. It aims to deliver a seamless shopping experience for customers while offering powerful tools for administrators to manage inventory, users, and orders.

Key Objectives:

Provide a centralized platform for purchasing a wide range of products.

Enable users to register, log in, browse products, add to cart, and place orders securely.

Offer an admin interface to manage products, users, and orders.

Ensure responsive, fast, and intuitive user interaction on both web and mobile devices.

Build the application using the **MERN stack (MongoDB, Express, React, Node.js)** for full-stack development.

IDEATION PHASE

PROBLEM STATEMENT:

As the world increasingly embraces the digital age, E-commerce stands at the forefront of transformative business models. However, the rapid evolution of this electronic marketplace brings forth a myriad of challenges that necessitate thorough investigation and strategic solutions. This research aims to address and analyze the persistent problems within the realm of E-commerce, identifying key issues that hinder optimal functionality and inhibit the realization of its full potential.

Empathy Map Canvas

ONLINE SHOPPER

SAYS

- I want the best deals without wasting time.
- I need filters to quickly fnd what I want
- The checkout process should be simple and secure.

THINKS

- Is this site trustworthy?
- Will I receive the right product on time?
- Are there hidden charges or delivery delays?

SEES

- Competitor websites with flashy but cluttered designs.
- Ads and product suggestions on social media
- Multiple tabs osen for price comparison.

DOES

- Compares prices across platforms.
- Uses wishlist and cart features
- Applies coupons and filters during checkout

HEARS

- Friends recommend specific online stores.
- Negative reviews or

FEELS

- Compares prices across platforms.
- Uses wishlist and cart features

Brainstorming

Problem Statement:

In today's fast-paced digital world, consumers face challenges in finding reliable, user-friendly platforms that offer a seamless online shopping experience across diverse product categories. Many existing platforms are cluttered, lack personalization, or are difficult for small vendors to onboard. Additionally, managing multiple purchases from various categories can be time-consuming and frustrating for users.

Proposed Solution:

ShopEZ is a full-stack e-commerce platform built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It offers a one-stop destination for consumers to browse, select, and purchase products across various categories with ease. The application provides features like user authentication, product search, personalized user profiles, a dynamic shopping cart, order history, and an intuitive admin dashboard for inventory and order management.

The design focuses on a clean, responsive UI with role-based access for users and admins. Vendors can add or update products, while customers can view detailed product listings and track their orders. Integration with secure payment gateways and smart search functionalities further enhances the overall user experience.

Target Users:

Everyday Consumers: Looking for an easy and secure platform to shop for a variety of products.

Vendors & Small Businesses: Who want a platform to showcase and sell their products without heavy technical knowledge.

Admin/Store Managers: To manage product listings, orders, and customer queries efficiently.

Expected Outcome:

The final product will be a responsive, robust web application that simplifies the online shopping process for both customers and sellers. ShopEZ will provide a seamless user experience, support secure transactions, and enable scalable management of products and orders through an intuitive admin panel. The application aims to become a reliable, all-in-one solution for online retail needs.

Requirement Analysis:

Customer Journey Map

	Awareness	Consideration	Decision	Purchase	Post Purchase	Loyalty
Customer Goal	Find a rellable site for shopping	Compare prices & products	Choose products & place an order	Complete secure transaction	Track and receive product	Return if satisfied
Actions						
Emotions	Curious, Hopeful	Interested	Confident	Excited, Cautious	Impatient, Anxious	Happy, Loyal
Pain Points	Trust issues with new platforms	Overwhelmed by product variety	Worries about payment or quality	Payment failure, long loading times	Delivery delay, poor packaging	No rewards for repest purchase
Opportunities	 Build credibility through social proof	 Use filters, sorting, and clean UI	 Offer COD, secure cheekout, reviews	 Multiple payment options fast UX	 Loyalty po-back, referan-earn	 Loyalty points refer and earn system

Solution Requirements for shopEZ:

1. Functional Requirements (FRs):-These define what the system should do:

1. User Registration & Login

Users can sign up, log in, and reset passwords securely.

2. Product Browsing & Search

Users can view product listings, search by name/category/brand.

3. Filters & Sorting

Users can filter by price, brand, rating, and sort results.

4. Product Detail View

Each product shows name, price, description, images, ratings, and availability.

5. Cart Management

Users can add/remove items, update quantity, and view cart summary.

6. Order Placement & Checkout

Checkout page for shipping address, payment method, and order review.

7. Payment Integration

Integration with multiple gateways (UPI, Card, Net Banking, COD).

8. Order Tracking

Users can view order status (confirmed, shipped, out for delivery, delivered).

9. User Profile & Order History

Manage personal info, past orders, returns, and wishlist.

2. Non-Functional Requirements (NFRs):-

These define how the system should perform:

1. Performance

Pages must load within 2–3 seconds for good user experience.

2. Scalability

Must support growth in users, products, and transactions.

3. Security

Encrypt user data; secure authentication and payment processing.

4. Usability

Responsive and user-friendly UI/UX on mobile and desktop.

5. Availability

System uptime of 99.9%; proper error handling and fallback.

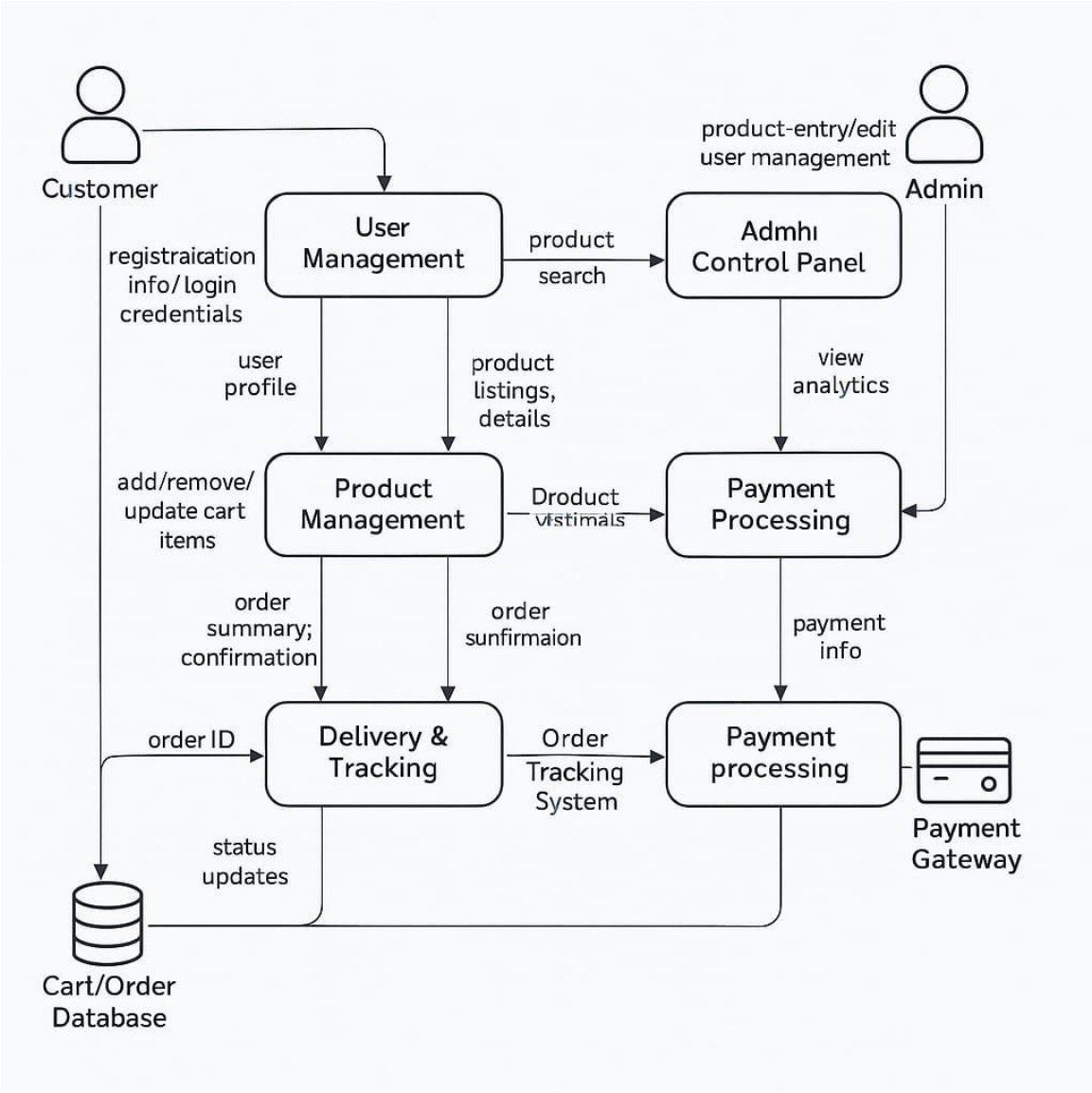
6. Maintainability

Clean, modular code for easy updates and debugging.

7. SEO Optimization

Ensure pages are optimized for search engines for organic traffic.

Data Flow Diagram:



Technical Requirements:

Frontend:

HTML5, CSS3, JavaScript

React.js (v18+)

Tailwind CSS or Bootstrap (for responsive UI)

Backend:

Node.js (v16+)

Express.js (v4+)

Database:

MongoDB (NoSQL Database)

Development Tools:

VS Code

Postman (for API testing)

Git & GitHub (Version Control)

Others:

JWT for Authentication

Mongoose (MongoDB ODM)

Cloudinary or Firebase (for image upload & storage)

Razorpay/Stripe (for payment gateway integration – optional)

Functional Requirements:

User Authentication:

Signup/Login for customers and admins

Role-based access control

Product Management:

- Admin dashboard to add, update, and delete products
- Product categories and filtering

Shopping Experience:

- Product search and sort
- Product detail view
- Add to Cart functionality
- Dynamic Cart Management
- Checkout and Order Placement

User Account Features:

- User profile with past orders
- Order tracking/status update
- Option to update profile details

Admin Features:

- Manage orders and users
- Inventory tracking
- Add new categories and offers

Frontend Features:

- Responsive design for all devices
- Error handling and validation
- Toasts/alerts for actions

Constraints & Challenges:

Scalability of the Platform:
Ensuring smooth performance as the number of users or products increases.

Security:
Protecting user data and transactions using secure authentication and encryption methods.

Payment Integration:
Handling real-time payments and dealing with potential errors or gateway failures.

Database Performance:
Efficient handling of queries and relationships, especially with product filtering and user orders.

User Experience (UX):
Designing a clean, intuitive UI for both users and admins.

PROBLEM STATEMENT:

As the world increasingly embraces the digital age, E-commerce stands at the forefront of transformative business models. However, the rapid evolution of this electronic marketplace brings forth a myriad of challenges that necessitate thorough investigation and strategic solutions. This research aims to address and analyze the persistent problems within the realm of E-commerce, identifying key issues that hinder optimal functionality and inhibit the realization of its full potential.

Shop EZ e-commerce platform:

Front-end:

1. HTML/CSS: Use semantic HTML and CSS to create a responsive and visually appealing design.

2. JavaScript: Use JavaScript to create interactive elements and enhance user experience.

3. Front-end Framework: Use a front-end framework like React, Angular, or Vue.js to build reusable components and manage state.

4. Responsive Design: Ensure the platform is optimized for various devices and screen sizes.

Back-end:

1. Programming Language: Use a programming language like Python, Java, or Node.js to build the server-side logic.

2. Framework: Use a framework like Django, Spring, or Express.js to build a scalable and maintainable architecture.

3. Database: Use a database management system like MySQL, PostgreSQL, or MongoDB to store and manage data.

4. API: Design a RESTful API to handle requests and responses between the front-end and back-end.

Database:

1. Database Schema: Design a database schema to store product information, user data, and order details.

2. Data Modeling: Use data modeling techniques to define the structure and relationships between data entities.

3. Database Indexing: Use indexing to improve query performance and reduce latency.

Security:

1. Authentication: Implement authentication mechanisms to secure user accounts and protect sensitive data.

2. Authorization: Implement authorization mechanisms to control access to sensitive data and features.

3. Data Encryption: Use encryption to protect sensitive data, such as payment information and user credentials.

4. Security Updates: Regularly update dependencies and patch vulnerabilities to ensure the platform remains secure.

Server and Infrastructure:

1. Cloud Hosting: Use a cloud hosting provider like AWS, Google Cloud, or Microsoft Azure to ensure scalability and reliability.

2. Server Configuration: Configure servers to optimize performance, security, and scalability.

3. Load Balancing: Use load balancing to distribute traffic and ensure high availability.

4. Monitoring and Logging: Use monitoring and logging tools to track performance, identify issues, and troubleshoot problems.

Testing and Quality Assurance:

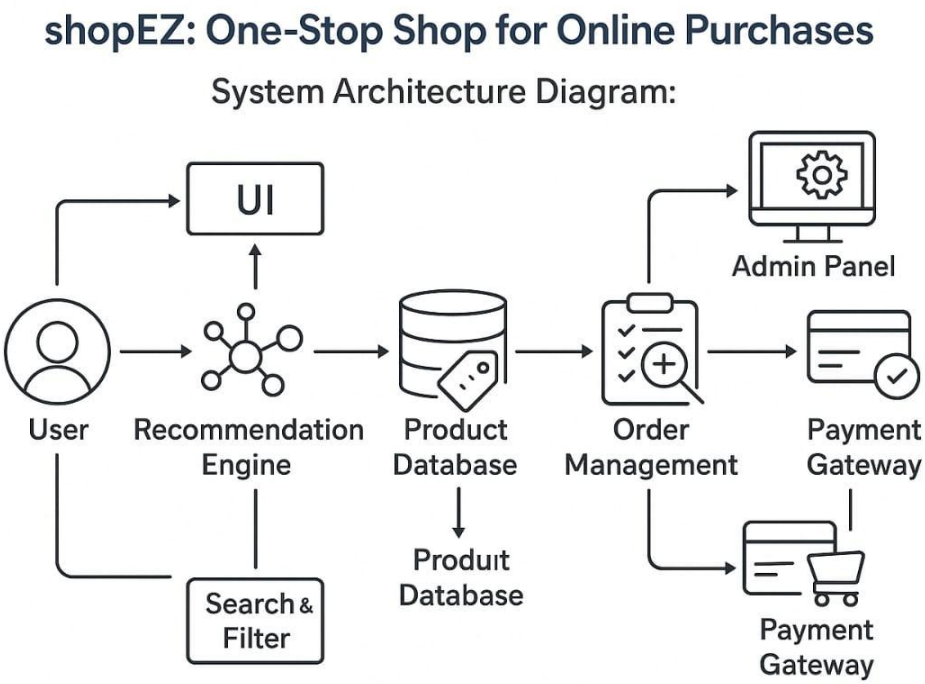
1. Unit Testing: Write unit tests to ensure individual components are working correctly.

2. Integration Testing: Write integration tests to ensure different components are working together correctly.

3. End-to-End Testing: Perform end-to-end testing to ensure the platform is working as expected.

4. Performance Testing: Perform performance testing to ensure the platform can handle high traffic and load.

Project Design



Product Input ➜ Backend Processing ➜ MongoDB Storage

• Dynamic Product Display ➜ User Interaction (React Frontend) ➜ Cart & Checkout System

• Admin Dashboard ➜ Product Management ➜ Order Tracking

• Integrated Search & Filters ➜ Enhanced User Navigation

• Uses `product_data.json` / MongoDB for dynamic content loading

• App built using MERN stack (MongoDB, Express.js, React.js, Node.js)

User Flow:

User lands on the ShopEZ homepage.

Browses categories or uses **search bar** to find products.

Adds items to the cart from the product listing page or product detail page.

Proceeds to **checkout**, completes payment, and gets confirmation.

Admins manage products, users, and orders through the admin panel.

UI/UX Considerations:

Responsive layout for all devices (desktop, tablet, mobile)

Clear navigation bar for Home, Categories, Cart, Profile

User-friendly cart and smooth checkout experience

Product cards with image, title, price, rating

Real-time updates using React states and backend APIs

Admin Dashboard with clear product add/edit/delete options

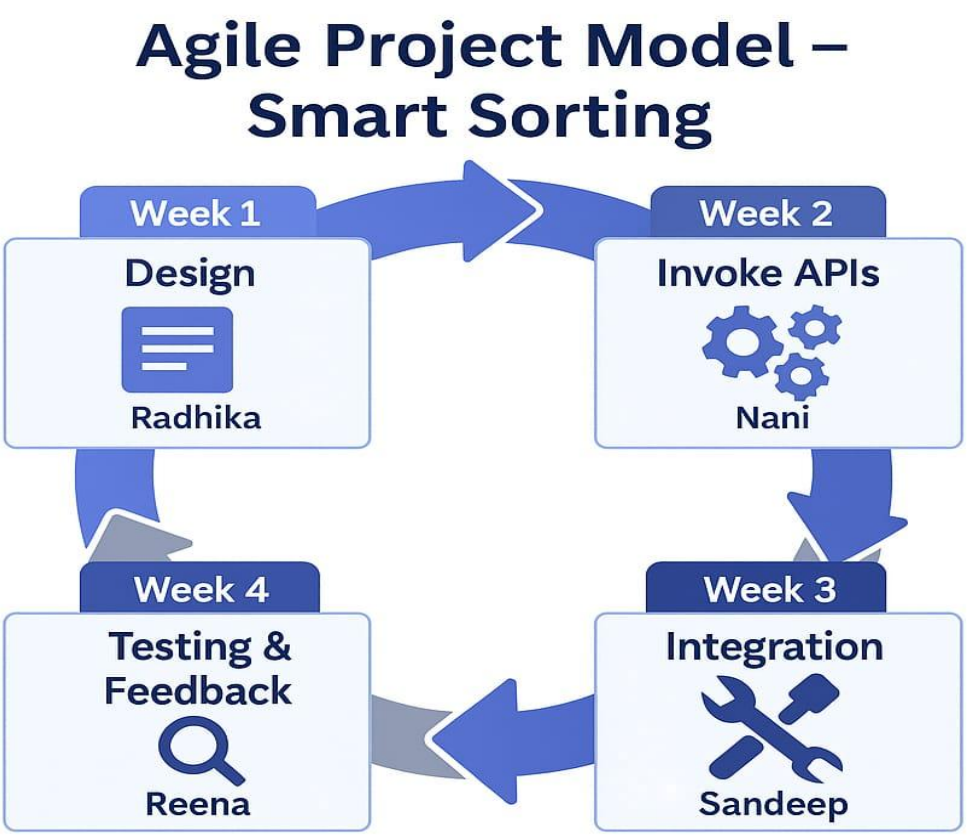
Project Planning

shopEZ

One-stop shop for online purchases

Project Planning

Sprint Planning & Task Allocation :



Timeline & Milestones:

4 Weeks development plan with checkpoints after each phase

Project Development

Technology Stack Used:

MongoDB – NoSQL database used to store product, user, and order information.

Express.js – Backend web application framework running on Node.js.

React.js – Frontend library used to build a responsive and dynamic user interface.

Node.js – JavaScript runtime used to build the backend logic and API routes.

HTML/CSS – Used along with React to style components and build layouts.

Mongoose – ODM (Object Data Modeling) library for MongoDB and Node.js.

JWT & Bcrypt.js – Used for secure authentication and password hashing.

Development Process:

Requirement Analysis – Identified core features like product listing, user authentication, cart, checkout, and admin panel.

Database Design – Created schemas for products, users, orders, and categories using Mongoose.

Frontend Development – Built reusable components in React (Navbar, ProductCard, Cart, etc.) and styled them using CSS.

Backend API Development – Developed RESTful APIs using Express.js for all major functionalities (CRUD, login, checkout).

Authentication & Authorization – Implemented JWT-based login system and protected routes for admin operations.

Admin Dashboard – Built a dedicated admin panel for managing inventory and tracking orders.

Challenges & Fixes:

Page load delays due to large images – Optimized by compressing and lazy-loading product images.

Inconsistent cart state after refresh – Solved using localStorage and Redux for persistent cart state.

Authentication failures on protected routes – Fixed by refining JWT middleware and token expiration logic.

Inefficient product filtering – Improved with optimized MongoDB queries and dynamic React filters.

Styling issues on mobile view – Resolved using media queries and responsive layout techniques.

Admin actions were not updating instantly – Addressed by adding real-time state updates and proper API response handling.

Functional & Performance Testing

Functional & Performance Testing

- Tested product search, filtering, and recommendation featuress varoios categories
- Verified cart operations: add/remove; quantity update, and total cost calculation
- Tested secure login, registration, and checkout workflows
- Checked mobile and desktop responsivenesss for all major screens
- Performance tests on:
  - Product listing load time
  - Payment gateway response
  - Search/filter query speed

Test Case	Expected Result	Status
Product Search (Mobile)	Returns accurate results for keywords	✅ Passed
Cart Update Function	Correctly adds/removees	✅ Passed
Checkout (Card Payment)	Secure and smooth transaction flow	✅ Passed

Functional & Performance Testing – shopEZ

✅ Test Cases Executed:

Tested product search, filtering, and recommendation features for various categories.

Verified cart operations: add/remove, quantity update, and total cost calculation.

Tested secure login, registration, and checkout workflows.

Checked mobile and desktop responsiveness for all major screens.

⚡ Performance tests on:

Product listing load time

Payment gateway response

Search/filter query speed

Benefits:

1. Convenience: Allow customers to shop for all their needs in one place.
2. Time-saving: Save customers time by providing a single platform for various products.
3. Competitive prices: Offer competitive prices and discounts to attract customers.
4. Wide product selection: Provide a vast selection of products to cater to different tastes and preferences.

Target Audience:

1. Demographics: Target a wide range of demographics, including age groups, genders, and income levels.
2. Psychographics: Target customers who value convenience, ease of use, and competitive prices.

Marketing Strategies:

1. Social media marketing: Utilize social media platforms to promote products and engage with customers.
2. Influencer marketing: Partner with influencers to promote products and reach a wider audience.
3. Email marketing: Send regular newsletters and promotional emails to subscribers.
4. Search engine optimization (SEO): Optimize the website for search engines to improve visibility and drive organic traffic.

Future Scope of shopEZ:

1. AI-Based Product Recommendations

Use machine learning to personalize product suggestions based on user behavior and preferences.

Improve user engagement and conversion rate.

2. Voice Search Integration

Add support for voice-based product searches.

Useful for mobile users or differently-abled users.

3. Inventory & Supplier Management System

Automate stock updates, reorder alerts, and supplier dashboards.

Improve backend operations for marketplace sellers.

4. Dedicated Mobile App (Android & iOS)

Offer smoother user experience and push notifications for offers and updates.

5. Enhanced Security Features

Two-factor authentication, fraud detection, and data encryption for enhanced trust and safety.

6. Multi-Language & Multi-Currency Support

Reach global customers with local language and currency conversions.

7. Customer Review & Rating System

Allow customers to leave reviews, rate products, and flag issues.

Increases transparency and trust.

8. Loyalty Programs & Coupons

Implement reward points, seasonal discounts, and referral codes to retain users.

9. Business Analytics Dashboard

Real-time analytics for admins: user activity, product performance, sales trends, etc.

10. Live Order Tracking & Delivery Partner Integration

Real-time map-based order tracking with delivery agent contact options.

Conclusion:

ShopEZ aims to combine modern web technologies with intuitive design to offer a complete e-commerce solution. The platform ensures robust functionality for both users and administrators, and the optional AI integration demonstrates the potential of machine learning in enhancing product management and user experience. With scalability and future growth in mind, ShopEZ can evolve into a feature-rich marketplace.