

# **A PROJECT REPORT**

on

## **ShopEZ: One-Stop Shop for Online Purchases**

(Discovering Opportunities, Unlocking Potential)

by

**SANGEPU HEMA LATA**

**PATHAN SUBHANI**

**PAGIDIPOGU DIVYA**

**PRANEETHA CHILLARA**

Under the guidance of

**Ganesh M**

# TABLE OF CONTENTS

NAME OF THE CONTENT	PAGE NOs
1.ABSTRACT	3
2.INTRODUCTION	4
3.MODULE DESCRIPTION	5-10
4.USER INTERFACE	11-12
5.TESTING	13
6.RESULT	14-15
7.FUTURE ENHANCEMENT.	16
8.CONCLUSION	17

# ABSTRACT

In today's fast-paced digital world, the demand for seamless, user-centric online shopping experiences has grown exponentially. **ShopEZ** is a full-stack e-commerce web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) that addresses this demand by offering a feature-rich, intuitive, and responsive platform for both customers and sellers.

The project aims to streamline the online shopping process with functionalities like product discovery, personalized recommendations, secure checkout, and real-time order tracking. Users can register, browse a wide catalog of products, add items to a cart, and place orders effortlessly. Integrated product reviews and search filtering improve user decision-making, while multi-option payments ensure convenience and security.

For sellers and administrators, ShopEZ offers a robust dashboard that simplifies inventory management, tracks orders, and provides insightful analytics to aid business growth. The platform also includes secure user authentication, order history management, and real-time notifications to create a smooth and reliable shopping environment.

By leveraging modern web technologies and a modular architecture, ShopEZ demonstrates the development of a scalable and dynamic e-commerce solution that caters to the evolving needs of online shoppers and digital retailers.

# INTRODUCTION

The rapid growth of digital commerce has transformed the way people shop, making online platforms an essential part of modern life. As consumers seek convenience, speed, and variety in their shopping experience, businesses are increasingly moving online to meet these expectations. In this context, **ShopEZ** was developed as a full-stack e-commerce web application designed to simplify and enhance the online shopping journey for both customers and sellers.

Built using the powerful MERN stack—**MongoDB**, **Express.js**, **React.js**, and **Node.js**—ShopEZ delivers a dynamic and responsive interface that ensures smooth navigation and interaction. Users can register, log in, explore products, apply filters, receive personalized recommendations, and complete purchases using a secure and efficient checkout process. The application also supports features like order tracking, product reviews, and profile management to enrich the customer experience.

For sellers and administrators, ShopEZ offers a dedicated dashboard to manage product listings, monitor sales, handle inventory, and analyse customer behaviour through actionable insights. This dual functionality makes ShopEZ a comprehensive platform that bridges the needs of both consumers and vendors.

In summary, ShopEZ is more than just an online store—it is a scalable, user-friendly, and efficient solution built to meet the evolving demands of the e-commerce industry.

# MODULE DESCRIPTION

## 1. Project Overview

### **Purpose:**

The purpose of the **ShopEZ** project is to create a seamless and efficient e-commerce platform that enhances the online shopping experience for both customers and sellers. It aims to simplify product discovery through intuitive navigation, smart filtering, and personalized recommendations, while ensuring a secure and smooth checkout process. For sellers, ShopEZ provides a powerful dashboard to manage products, track orders, and analyse sales performance effectively. By leveraging modern web technologies like React.js, Node.js, Express, and MongoDB, the application ensures speed, scalability, and real-time responsiveness. Overall, ShopEZ bridges the gap between consumers and businesses by offering a reliable, full-stack solution for modern digital commerce.

### **Features:**

**Comprehensive Product Catalog:** ShopEZ offers a vast collection of products across multiple categories. Users can easily explore items using intuitive navigation, category sorting, and detailed product descriptions that include images, prices, and available discounts.

**Shopping Cart and Checkout:** Users can add products to their cart, review their selections, update quantities, and remove items. The checkout process supports multiple payment options and is designed for speed, simplicity, and security.

**User Authentication and Security:** ShopEZ uses secure user registration and login functionality. Sensitive data like user credentials and payment information are handled with care using authentication tokens and encryption techniques.

**Admin Dashboard:** Admins monitor platform activity, ensure security, and manage reported issues.

## 2. Architecture

### Frontend:

The frontend of **shopEZ** is built using **React.js**, a component-based library that supports dynamic user interfaces and seamless data rendering. The UI is styled using **Bootstrap** and **Material UI**, ensuring responsive design and accessibility across devices. **Axios** is used for making HTTP requests to the backend APIs. The application uses React Router for page navigation and context or state management hooks (like `useState`, `useContext`) to handle user sessions, form data, and UI state. The frontend provides role-specific views and forms tailored for user and admin.

### Backend:

The backend of **ShopEZ** is built using **Node.js** with the **Express.js** framework, providing a scalable and efficient foundation for handling server-side operations. It acts as the bridge between the frontend and the database, managing business logic, routing, authentication, and data storage seamlessly. The backend includes middleware for token-based authentication (using JWT), request validation, error handling, and access control. Controllers manage the logic for each route, ensuring separation of concerns and clean code organization.

### Database:

The **ShopEZ** e-commerce application uses **MongoDB** as its database solution. MongoDB is a **NoSQL** database known for its flexibility, scalability, and performance, making it ideal for storing large volumes of structured and semi-structured data in JSON-like documents.

Key collections include:

Users: Stores user information such as:

- Name, email, password (hashed), role (user/admin), and contact details.

Products: Holds all product listings, each with:

- Product name, category, price, image URL, stock count, ratings, and description.

Carts: Tracks items users have added to their cart.

Orders: Contains all order-related information.

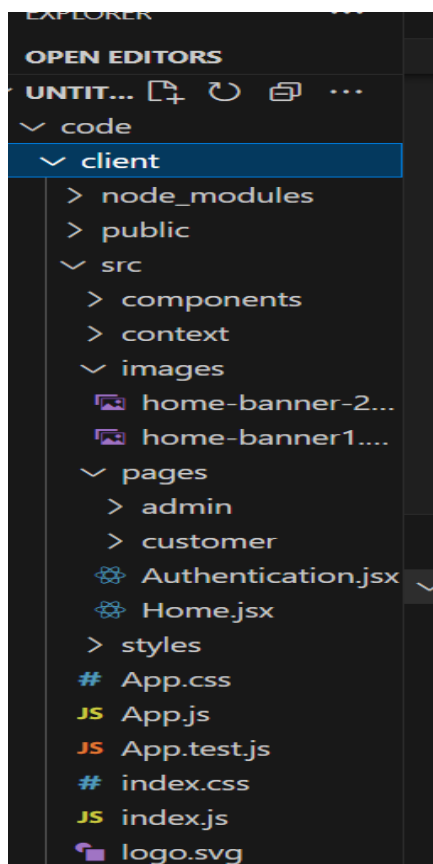
Admin: Stores banner images, product categories, admin credentials.

The database is optimized for quick access and supports relational referencing where needed (e.g., linking users to projects or reviews). Environment variables securely store the MongoDB connection URI.

### 3.Folder Structure

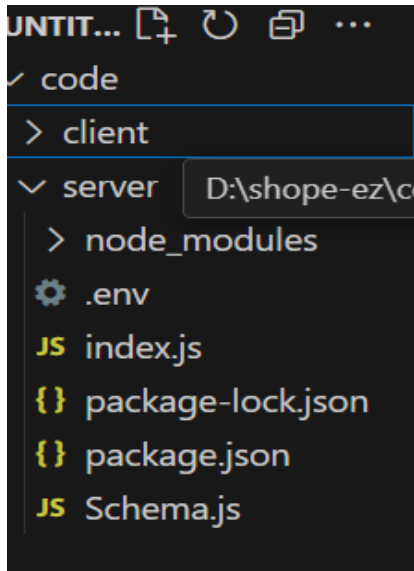
#### Client (React Frontend):

The client folder contains all the source code related to the frontend built with React. It follows a modular structure to separate UI components, pages, services, and static assets.



## Server (Node.js Backend):

The server folder contains backend logic, route handlers, controllers, models, and middleware necessary to power the API.



## 4.Setup Instructions

### Prerequisites:

Before setting up the project, ensure the following software is installed on your system:

- [Node.js](#) (v18 or higher)
- [MongoDB](#) (local or MongoDB Atlas cloud setup)
- npm (comes with Node.js) or yarn
- Git (for cloning the repository)
- Code editor (e.g., VS Code)

### Installation:

Follow the steps below to set up the project locally:

#### 1. Clone the Repository:

git clone <https://github.com/your-username/sb-works.git>

#### 2. Navigate to the Project Directory:



```
cd code
```

### **3. Install Frontend Dependencies:**

```
npm install
```

### **4. Install Backend Dependencies:**

```
cd server
```

```
npm install
```

### **5. Setup Environment Variables:**

- Create a **.env** file in the server directory and add the following:

```
MONGO_URI=your_mongodb_connection_string
```

```
JWT_SECRET=your_jwt_secret_key
```

```
PORT=5000
```

### **6.Start the Backend Server:**

```
npm start
```

### **7.Start the Frontend Server:**

```
npm start
```

Once both servers are running, open <http://localhost:3000> in your browser to access the application.

## **ShopEZ – REST API Documentation:**

### **Authentication APIs:**

- POST /api/auth/register
- POST /api/auth/login

### **User APIs:**

- GET /api/users/profile

### **Product APIs:**

- GET /api/products
- GET /api/products/:id

### **Cart APIs:**

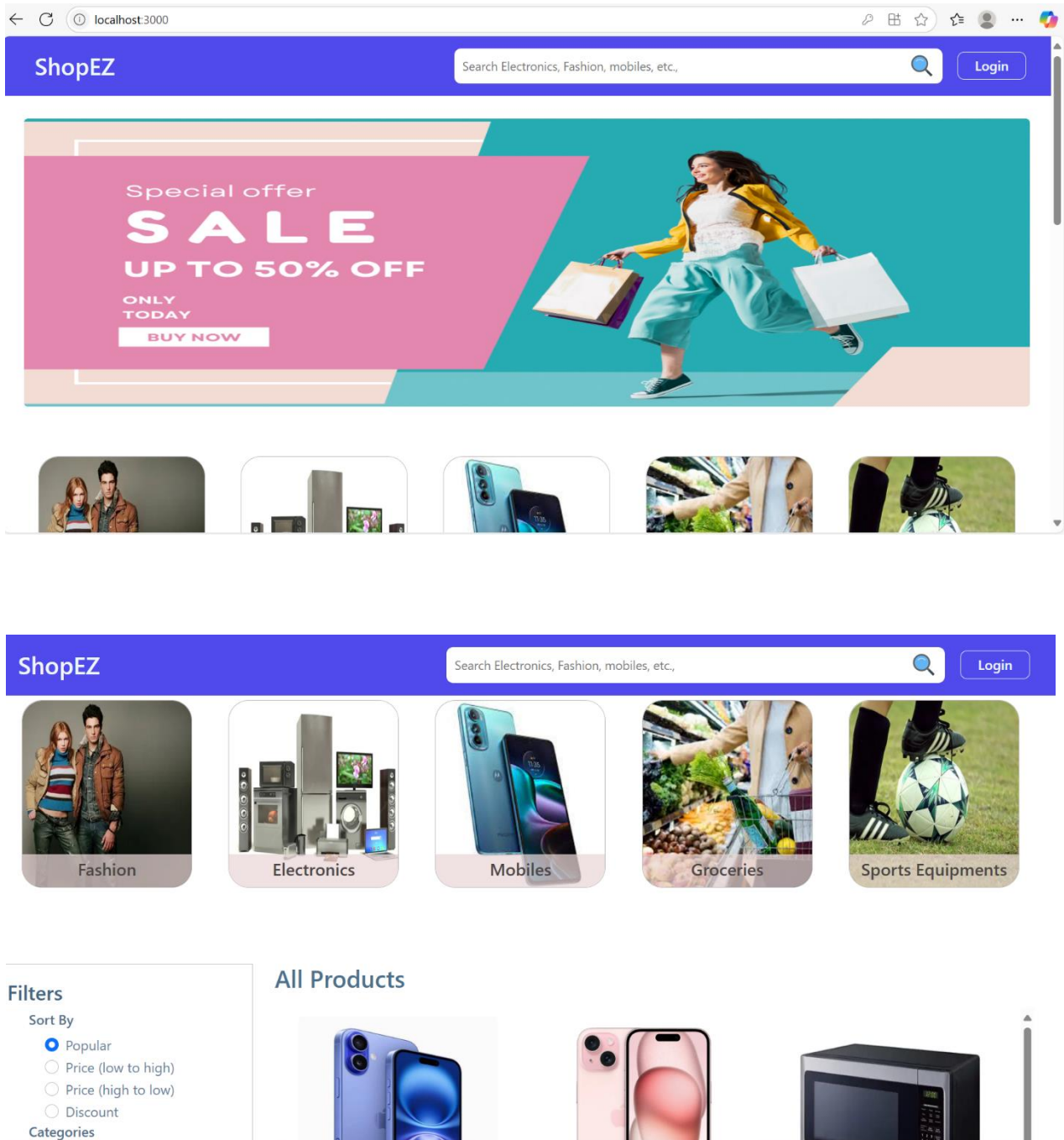
- POST /api/cart

### **Order APIs:**

- POST /api/orders

# USER INTERFACE

## HOME PAGE:



## ADMIN PAGE:

ShopEZ (admin)

HomeUsersOrdersProductsNew ProductLogout

New Product

Product name

Product Description

Thumbnail Img url

Add on img1 url

Add on img2 url

Add on img3 url

Available Size

☐ S☐ M☐ L☐ XL

Gender

Add product

localhost:3000/admin

ShopEZ (admin)

HomeUsersOrdersProductsNew ProductLogout

Total users

2

View all

All Products

5

View all

All Orders

1

View all

Add Product

(new)

Add now

Update banner

Banner url

Update

# TESTING

Testing is a crucial phase of the development process to ensure the functionality, performance, and reliability of the ShopEz. Both manual and automated testing methods were employed to validate the frontend and backend components.

## Manual Testing:

- **Frontend:**

Manual testing was performed using different user roles such as **customer** and **admin** to validate Page rendering and component visibility based on user permissions and Input validations etc.

- **Backend:**

Used **Postman** to test all API endpoints with valid and invalid inputs to ensure proper request/response handling and error management.

- **Responsiveness:**

Manually tested the UI across different screen sizes (mobile, tablet, desktop) to ensure a consistent and responsive user experience.

## Automated Testing (Planned / Partial):

- **Frontend(React):**

Integration of **Jest** and **React Testing Library** was initiated to test components like forms, buttons, and API responses.

- **Backend(Node.js+Express):**

Planned to integrate **Mocha**, **Chai**, and **Supertest** to automate endpoint testing and simulate various edge cases.

- **Key Areas Tested:**

- User registration and login validation
- Product listing, cart operations, and checkout process
- Admin access to product and user management
- Error message handling for invalid actions
- UI responsiveness across different devices

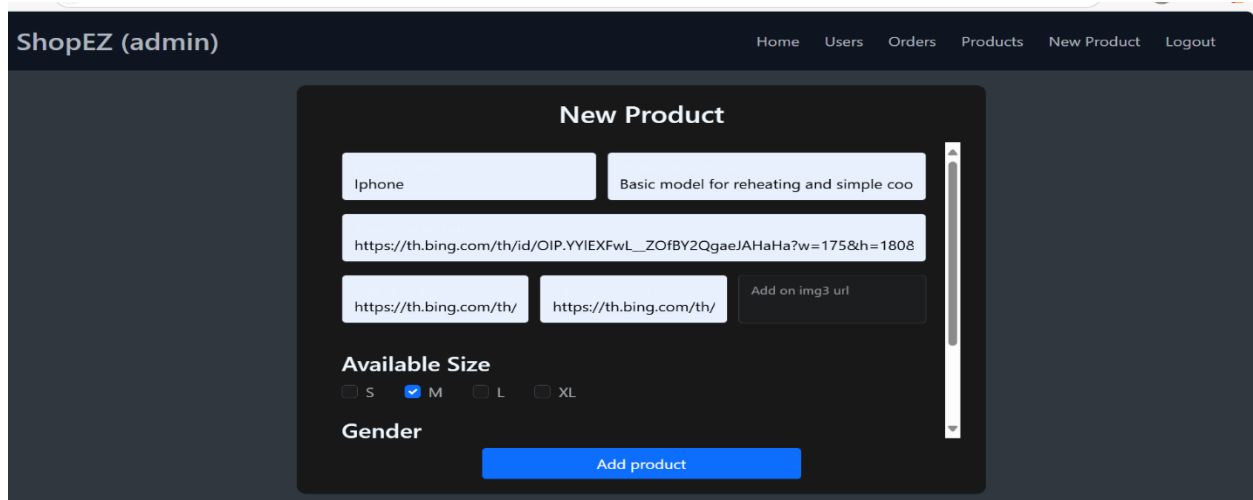
# RESULT

To demonstrate the functionality and user experience of shopEZ-one stop shop for online purchases a working demo and key UI screenshots have been provided:

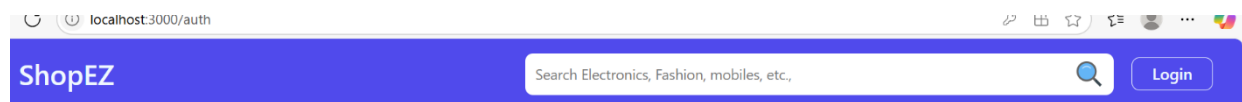
- **LiveDemo:**

<https://drive.google.com/file/d/1psHGX3gGc90fL6qiFiGPLKLG3110HiMM/view?usp=sharing>

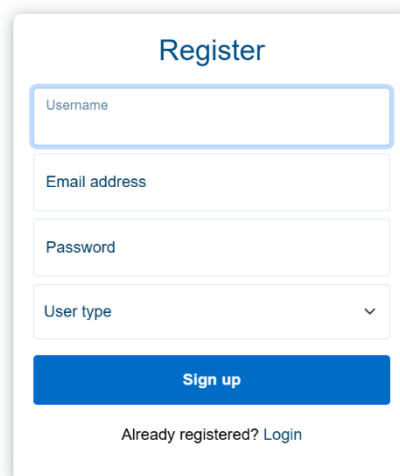
- **GitHub Link:**



The screenshot shows the 'ShopEZ (admin)' dashboard. At the top, there is a navigation bar with links: Home, Users, Orders, Products, New Product, and Logout. The main content area features a 'New Product' form. The form includes input fields for 'Name' (containing 'Iphone'), 'Description' (containing 'Basic model for reheating and simple coo'), and 'Image URL' (containing a Bing image link). There are also fields for 'Thumbnail 1' and 'Thumbnail 2', both containing the same Bing image link. A button labeled 'Add on img3 url' is next to the second thumbnail field. Below these fields, there is a section for 'Available Size' with radio buttons for S, M (selected), L, and XL. A 'Gender' section is also present. At the bottom of the form is a blue button labeled 'Add product'.



The screenshot shows the ShopEZ user interface. At the top, there is a blue header bar with the ShopEZ logo on the left, a search bar in the center with the placeholder text 'Search Electronics, Fashion, mobiles, etc.', and a 'Login' button on the right.



The screenshot shows a 'Register' form. It has a title 'Register' at the top. Below the title are four input fields: 'Username', 'Email address', 'Password', and 'User type' (a dropdown menu). At the bottom of the form is a blue button labeled 'Sign up'. Below the button, there is a link that says 'Already registered? Login'.



## Filters

## Sort By

- ☒ Popular  
☐ Price (low to high)  
☐ Price (high to low)  
☐ Discount

## Categories

- ☐ Mobiles  
☐ Electronics  
☐ Electronics  
☐ Sports Equipment

## Gender

- ☐ Men  
☐ Women  
☐ Unisex

## All Products



## Iphone

it's the latest iphone 16....

₹ 59399 ~~59999~~ ( 1% off)

## Iphone

Sleek design with improved cam....

₹ 54450 ~~55000~~ ( 1% off)

## microwave oven

Basic model for reheating and ....

₹ 8645 ~~17290~~ ( 50% off)

## Login

Email address

Password

Sign in

Not registered? [Register](#)

# FUTURE ENHANCEMENTS

To keep ShopEZ competitive, scalable, and aligned with evolving user expectations, several enhancements can be considered for future development. These improvements aim to enrich the shopping experience, boost platform efficiency, and extend functionality for both users and sellers.

1. **Wishlist Functionality:** Allow users to save their favourite products for future purchases. This will enhance user retention and improve the shopping experience.
2. **Product Recommendation Engine (AI-based):** Implement machine learning algorithms to deliver smarter, personalized product suggestions based on user behaviour, browsing history, and purchase trends.
3. **Real-Time Chat Support:** Integrate a chatbot or live chat feature to allow users to get instant help from customer support or sellers regarding product queries and order issues.
4. **Advanced Search with Voice & Image Recognition:** Enhance the search experience by allowing users to search using voice commands or images (e.g., upload a product photo to find similar items).
5. **Push Notifications (PWA or WebSockets):** Enable push notifications for order updates, deals, and promotions. This can improve engagement and bring users back to the app more frequently.
6. **Coupon & Discount Management:** Add a dynamic coupon system that allows admins/sellers to create and manage discount codes for marketing and promotions.
7. **Product Return & Refund Workflow:** Introduce a return/refund module where users can request product returns with reason codes, and admins can approve or reject based on policies.
8. **Multi-Vendor Support:** Expand the platform to support multiple vendors/sellers under one marketplace with seller-specific dashboards, ratings, and product management.



## CONCLUSION

In conclusion, **ShopEZ** is a robust and user-friendly full-stack e-commerce application designed to streamline the online shopping experience for both customers and administrators. By leveraging the MERN stack, the platform ensures seamless performance, real-time data handling, secure authentication, and scalable architecture. It successfully integrates essential features such as product browsing, shopping cart, secure checkout, order tracking, and admin management. Through responsive design and efficient backend operations, ShopEZ demonstrates the potential to serve as a reliable and modern e-commerce solution. With planned future enhancements, it holds great promise for further growth and real-world deployment.