

ShopEZ: Full Stack E-commerce Platform

Complete Project Documentation

1. Introduction

Project Title: ShopEZ One-Stop Shop for Online Purchases

Team Members

- **Chemalamarri Vyshnavi** – Full Stack Developer: Manages end-to-end development, authentication systems, and deployment.
- **B Praveen Varma** – Frontend Developer: Responsible for React.js UI components, user experience design, and client-side functionality.
- **Amballa Vigneswari** – Backend Developer: Handles Node.js/Express.js server development, API creation, and database integration.
- **B Manogna** – Database Administrator: Designs MongoDB schemas, optimizes queries, and manages data relationships

2. Project Overview.

Purpose

ShopEZ is a comprehensive e-commerce platform designed to revolutionize the online shopping experience by providing a seamless interface for customers while empowering sellers with robust management tools. The platform bridges the gap between buyers and sellers through an intuitive, secure, and scalable solution that handles everything from product discovery to order fulfillment.

Key Features

- **User Management:** Secure registration, login, and profile management with role-based access control
- **Product Catalog:** Dynamic product listings with advanced filtering, search, and categorization
- **Shopping Cart:** Persistent cart functionality with real-time updates and quantity management

- **Secure Checkout:** Streamlined checkout process with multiple payment options and delivery tracking
 - **Order Management:** Comprehensive order tracking from placement to delivery
 - **Admin Dashboard:** Powerful administrative tools for managing products, orders, users, and platform analytics
 - **Responsive Design:** Mobile-first approach ensuring optimal experience across all devices
 - **Real-time Updates:** Live inventory updates, order status changes, and notification systems
 - **Personalization:** User-specific recommendations and browsing history
 - **Banner & Categories Management:** Dynamic content management for promotional materials and product organization
-

3. Architecture

Frontend Architecture (React.js)

The frontend follows a component-based architecture utilizing React.js for building dynamic user interfaces:

Component Hierarchy:

- **App Component:** Root component managing routing and global state
- **Layout Components:** Header, Footer, Navigation, and Sidebar components
- **Page Components:** Home, Products, Cart, Checkout, Profile, and Admin pages
- **Feature Components:** ProductCard, FilterBar, SearchBox, CartItem, OrderSummary
- **Utility Components:** Loading spinners, error boundaries, and modal dialogs

State Management:

- Implements Context API for global state management

- Manages user authentication state, cart data, and application-wide settings
- Local component state for form handling and UI interactions

Routing:

- React Router for client-side navigation
- Protected routes for authenticated users and admin panels
- Dynamic routing for product categories and individual product pages

Backend Architecture (Node.js & Express.js)

The backend follows a RESTful API architecture with clear separation of concerns:

Server Structure:

- **app.js:** Main application file with Express configuration
- **Routes:** Organized API endpoints for different functionalities
- **Controllers:** Business logic handlers for each route
- **Middleware:** Authentication, validation, error handling, and logging
- **Models:** Mongoose schemas defining data structures
- **Utils:** Helper functions, validation schemas, and utility modules

API Design:

- RESTful endpoints following standard HTTP methods (GET, POST, PUT, DELETE)
- Consistent response format with proper HTTP status codes
- Input validation and sanitization for all incoming requests
- Error handling middleware for centralized error management

Database Architecture (MongoDB)

MongoDB provides the NoSQL foundation with the following collections:

Core Collections:

- **Users:** Stores user credentials, profiles, and authentication data

- **Products:** Contains product information, images, pricing, and inventory
- **Cart:** Manages shopping cart items linked to specific users
- **Orders:** Comprehensive order data including delivery and payment information
- **Categories:** Product categorization for organized browsing
- **Banners:** Promotional content and featured product displays
- **Admin:** Administrative user data and permissions

Relationships:

- One-to-Many: User to Cart items, User to Orders
 - Many-to-Many: Products to Categories
 - Referenced relationships using ObjectId for data integrity
-

4. Project Setup & Installation

Prerequisites

Ensure the following software is installed on your system:

- **Node.js** (version 14.x or higher)
- **npm** or **yarn** package manager
- **MongoDB** (version 4.x or higher) or MongoDB Atlas account
- **Git** for version control
- **Code Editor** (VS Code recommended)

Setup Instructions

1. Clone the Repository:

<https://github.com/vyshu0111/ShopSell>

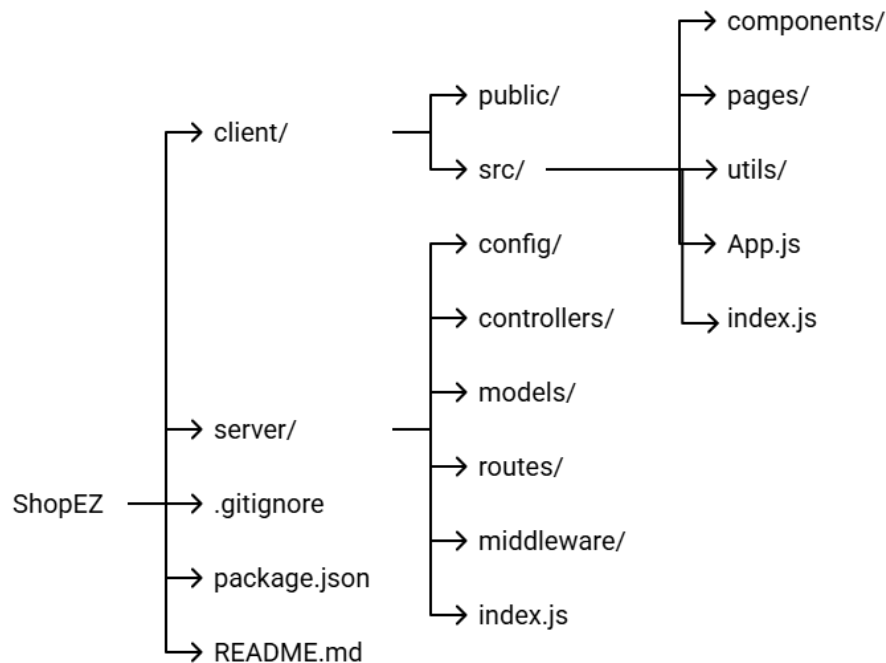
2. Backend Setup:

```
cd ShopEZ/backend
npm install
npm start
```

3. Frontend Setup:

```
cd ../client  
npm install  
npm start
```

4. Folder Structure



5. System Workflow – End-to-End Scenario

Scenario: Sarah's Birthday Gift

1. Effortless Product Discovery

- Sarah visits ShopEZ to find a birthday gift.
- Navigates to the "Fashion Accessories" category.
- Applies filters ("Bracelets", price range, style preferences).

2. Personalized Shopping Experience

- Sarah notices a "Recommended for You" section.
- Based on her search pattern and user profile, she is shown curated results.

- She selects a gold bangle and adds it to her cart.

3. Seamless Checkout Process

- Sarah enters delivery details (Emily's address).
- Selects preferred payment method.
- Confirms order with secure checkout.

4. Order Confirmation & Tracking

- Sarah receives an order confirmation email instantly.
- She can track the status via the "My Orders" section.

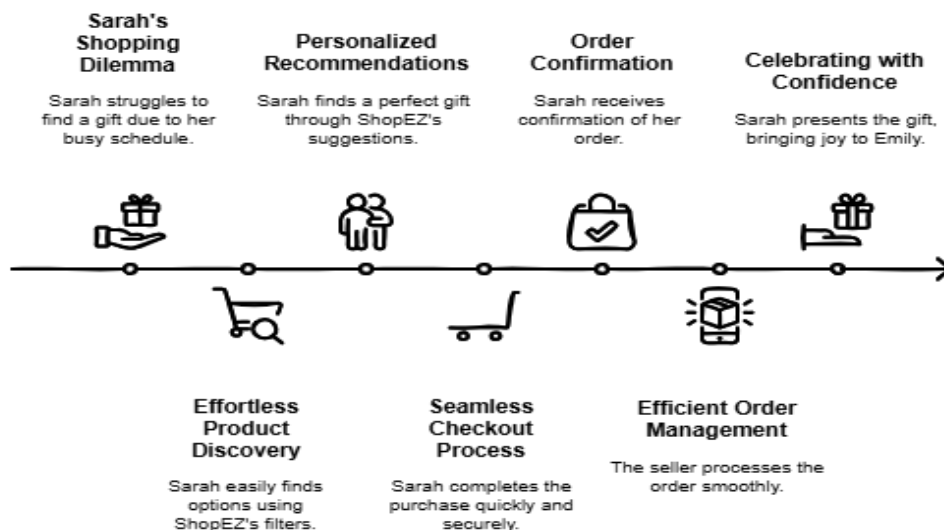
5. Seller Order Notification & Fulfillment

- The seller gets notified via the ShopEZ Seller Dashboard.
- Processes the order, updates shipment status.
- Packs and dispatches the gift promptly.

6. Celebration with Confidence

- Emily receives the gift on time.
- Sarah receives delivery confirmation.
- Both are happy and satisfied with the streamlined experience.

Sarah's Birthday Gift Shopping Journey



7. API Endpoint Documentation – ShopEZ



Authentication

- POST /register – Registers a new user (Customer/Admin/Seller).
- POST /login – Logs in a user and returns user data on success.



Product APIs

- GET /fetch-products – Fetch all products.
- GET /fetch-product-details/:id – Get details of a single product by ID.
- POST /add-new-product – Add a new product (Admin only).
- PUT /update-product/:id – Update an existing product by ID (Admin only).
- GET /fetch-categories – Fetch all product categories.
- POST /update-banner – Update banner image (Admin only).
- GET /fetch-banner – Get current homepage banner.



Order APIs

- POST /buy-product – Buy a product directly.
- POST /place-cart-order – Place order for all cart items.
- GET /fetch-orders – Get all placed orders.
- PUT /cancel-order – Cancel a specific order.
- PUT /update-order-status – Admin updates the order status (e.g., dispatched, delivered).



Cart APIs

- GET /fetch-cart – Fetch all cart items.
- POST /add-to-cart – Add a product to cart.

- PUT /increase-cart-quantity – Increase quantity of a cart item.
- PUT /decrease-cart-quantity – Decrease quantity or remove item if 0.
- DELETE /remove-item/:id – Remove item from cart by ID.

User/Admin APIs

- GET /fetch-users – Get all registered users.

8. Authentication & Authorization – ShopEZ

User Signup

- Users register via POST /register, providing:
 - username, email, usertype (admin/user/seller), password
- Passwords are hashed securely using **bcrypt**.
- Duplicate email check is implemented.

User Login

- Endpoint: POST /login
- Validates credentials.
- On success, returns user data (currently no JWT used, but session/token-based auth can be added).

Authorization Logic

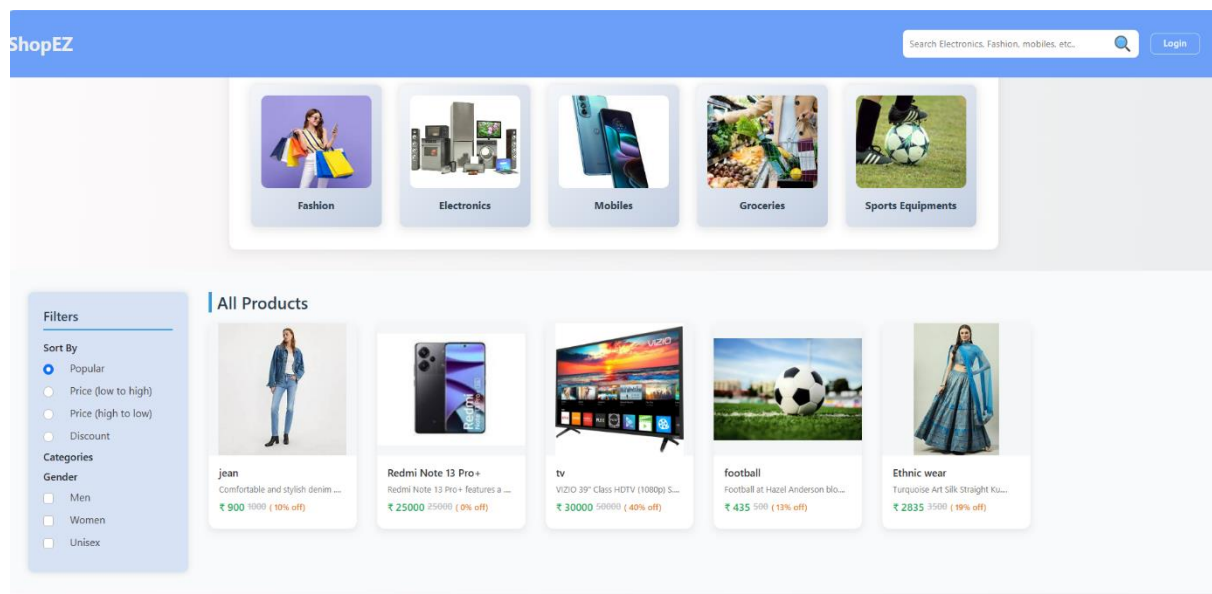
- **Usertype Roles:**
 - admin: Can manage products, categories, users, and orders.
 - seller: Will get access to seller dashboard (future enhancement).
 - user: Can browse, add to cart, and place orders.

Protected Actions (by logic)

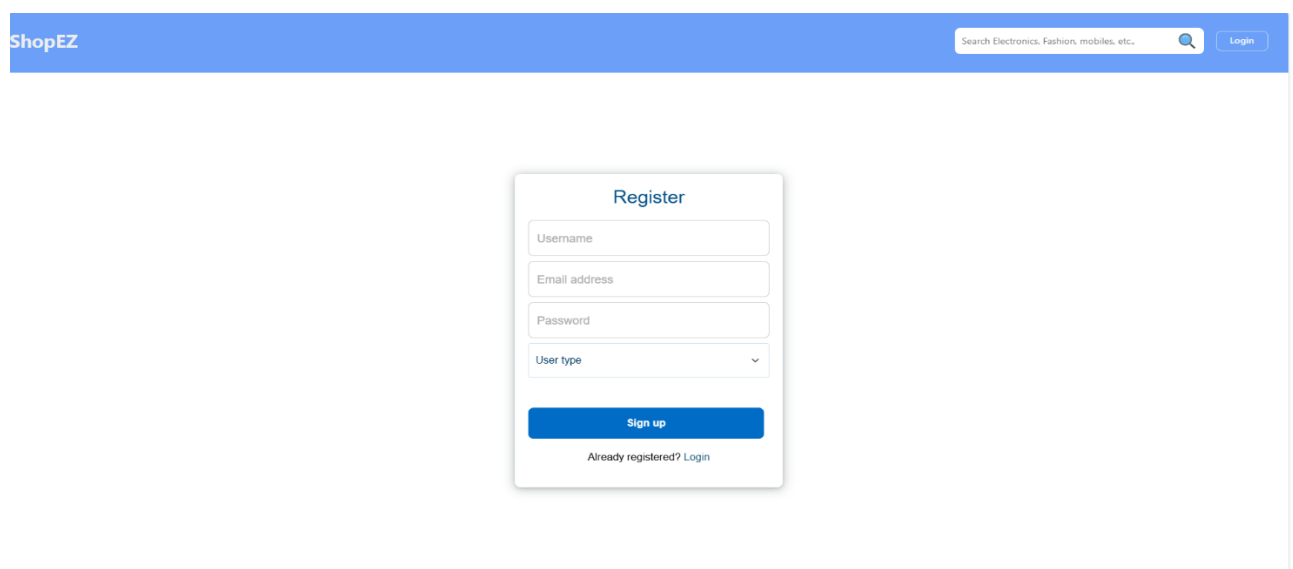
- Product management, banner updates → Admin only.
- Order placing, cart → Logged-in users.
- Order status update → Admin.

9. User Interface

1. Landing page



Authentication



All Users

ShopEZ (admin)

HomeUsersOrdersProductsNew ProductLogout

All Users

USER ID
605ae539c92658c898613cc1

USER NAME
Srilu

EMAIL ADDRESS
srilu@gmail.com

ORDERS
0

USER ID
605c3c882403cf1149dd0a2d

USER NAME
asha

EMAIL ADDRESS
asha@gmail.com

ORDERS
1

Admin Dashboard

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

Orders

ShopEZ (admin)

HomeUsersOrdersProductsNew ProductLogout

Orders



jean

Comfortable and stylish denim jeans perfect for everyday wear.

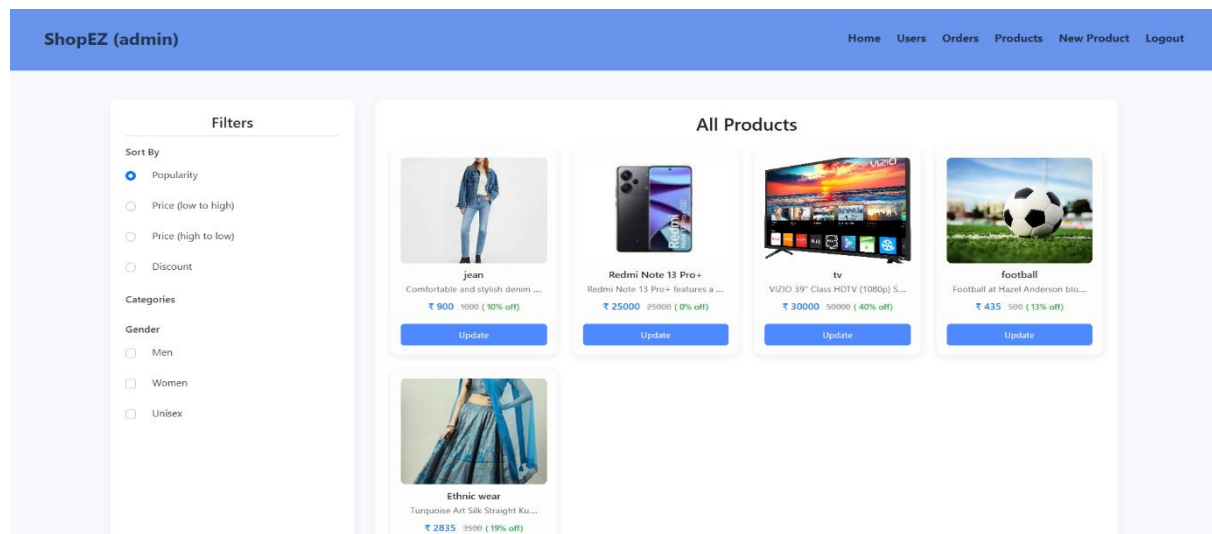
Size:
Quantity: 1
Price: ₹ 900
Payment method: upi

Userid: 605c3c882403cf1149dd0a2d
Name: asha
Email: asha@gmail.com
Mobile: 7780625095

Ordered on: 2025-06-25
Address: door number 11-220, near chinna high school,panditivari palem,piduguralla
Pincode: 522413

Order status: cancelled

Products



11. Known Issues

1. No Email Notification Support

- The system doesn't send emails for order confirmations, cancellations, or account events.





2. No Real-Time Unread Message Indicators




- While basic functionality exists, there's no badge or indicator for unread chats/messages.

3. Minimal Front-End Validation

- Required field checks exist, but format checks (like email format or password strength) are minimal.

12. Future Enhancements

-  **Email/SMS Notifications** for order status and promotional offers.
-  **Real-Time WebSocket Chat** using Socket.IO for buyer-seller communication.
-  **Admin Analytics Dashboard** with charts (product views, sales trends, user growth).
-  **Mobile Responsiveness** for all pages.

-  **Advanced Search & Filtering** (by category, price, brand, reviews, etc.)
-  **Product Reviews and Ratings**
-  **Wishlist Feature** for users