**Internship Project Report**

**Project Title: ShopEZ( One-Stop Shop for Online Purchases )**

Submitted in partial fullfillment of the requirements for the

**Full Stack Development with MERN**

conducted by

# SmartBridge

***Submitted by***

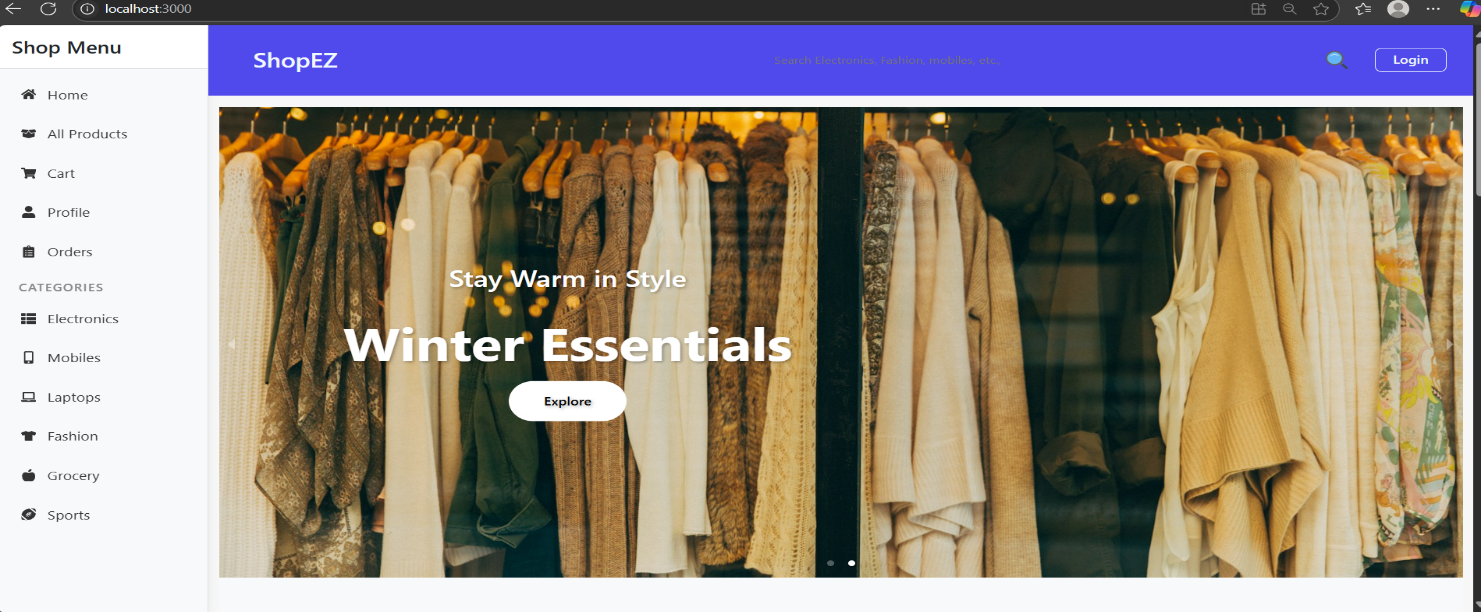
# Team Leader: Mygapula Durga Bhavani

# Team member: Lakshmi Narasimha Bokka

# Team member: Leela Sai Gannabathula

# Team member: Leela Krishna Yatam

# Team ID: LTVIP2025TMID58337



**Project Overview**

**Purpose:**

The purpose of this project is to design and develop a comprehensive and responsive full-stack e-commerce web application using the MERN (MongoDB, Express.js, React.js, and Node.js) stack.

The application aims to provide users with a seamless online shopping experience, including browsing through categorized product listings, managing their cart and orders, and accessing user profiles.

This project addresses the growing need for businesses to have a reliable digital presence. In today’s fast-paced environment, where users demand quick and intuitive access to products and services, our e-commerce platform bridges the gap by delivering dynamic and user-friendly interfaces powered by modern technologies.

The goals of this project include:

* Creating a responsive and aesthetically pleasing front-end experience.
* Ensuring secure authentication and user management.
* Enabling product browsing, detailed product views, and filtered searches.
* Allowing users to register, log in, manage carts, and place orders efficiently.

**Features:**

1. User Authentication and Authorization
2. Product Catalog Management
3. Shopping Cart and Wishlist
4. Order Placement and History
5. Profile Management
6. Admin Dashboard
7. Responsive UI/UX
8. Database Integration
9. Error Handling and Validation

**Architecture**

**Frontend:**

The frontend of this application is developed using React.js, following a modular, component-based architecture. Each page and UI segment is split into reusable components to enhance code maintainability and scalability.

React Router is employed for client-side routing, enabling navigation between views without full page reloads.

**Backend:** The backend is powered by Node.js with Express.js framework. It is structured into modular files and folders separating routes, controllers, middleware, and services for clarity and scalability.

Middleware is used for request logging, authentication, error handling, and input validation. JSON Web Tokens (JWT) are implemented to authenticate users and protect private routes.

**Database:**

MongoDB serves as the NoSQL database for the application. The data schema is designed using Mongoose, which provides schema-based modeling and validation.

Collections include:

1. **Users** – stores credentials, profile info, roles, and order history.
2. **Products** – details such as name, category, price, image, description, stock.
3. **Orders** – order summary, user association, list of products, status, timestamps.
4. **Carts** – temporary storage of users’ added products before checkout.
5. **Wishlist** – tracks user-marked items for future reference.

**Setup Instructions**

**Prerequisites:** Before beginning the installation process, ensure that the following software and tools are installed on your machine:

1. Node.js
2. MongoDB
3. Git
4. Code Editor
5. Postman

**Installation:** Step-by-step guide to setting up the application:

**Clone the Repository:** git clone https://github.com/your-org/mern-ecommerce.git

cd mern-ecommerce

**Install Server Dependencies:** cd server npm install

**Install Client Dependencies:** cd ../client npm install

**Configure Environment Variables:** MONGODB\_URI=mongodb://localhost:27017/ecommerce

**Running the Backend Server:** cd server npm run dev

**Running the Frontend React App:** cd client npm start

Make sure MongoDB is running: mongod

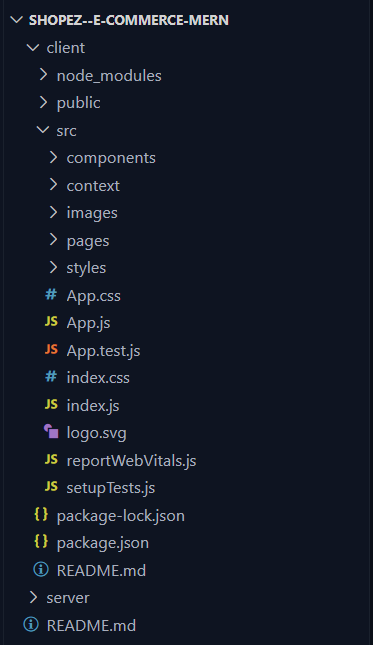
**Production Deployment:**

Use build tools: npm run build in the client directory.

Serve static files from Express in production mode.

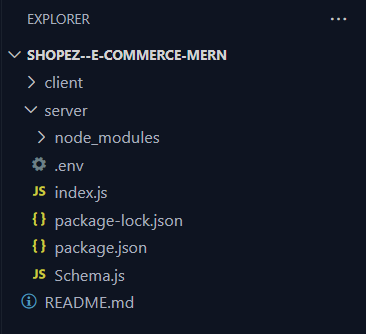
**Folder Structure**

**Clinet:**

****

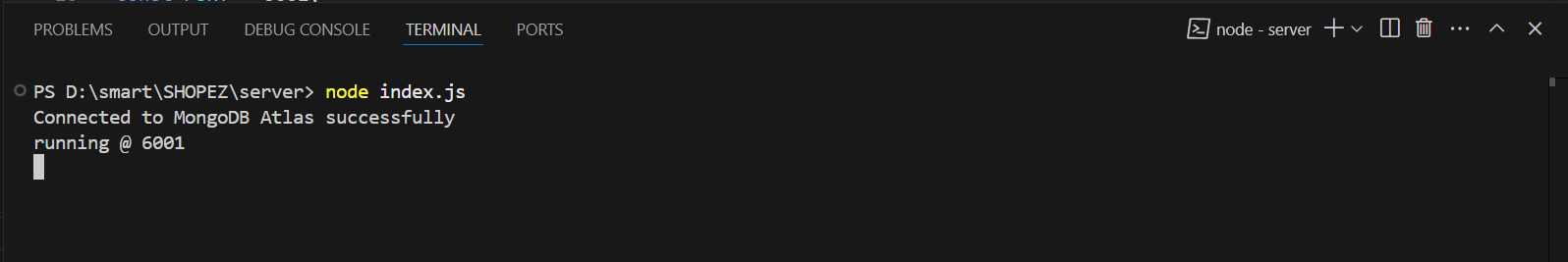
This structure assumes a React app and follows a modular approach. Here's a brief explanation of the main directories and files:

**Server:**

****

**Running the Application**

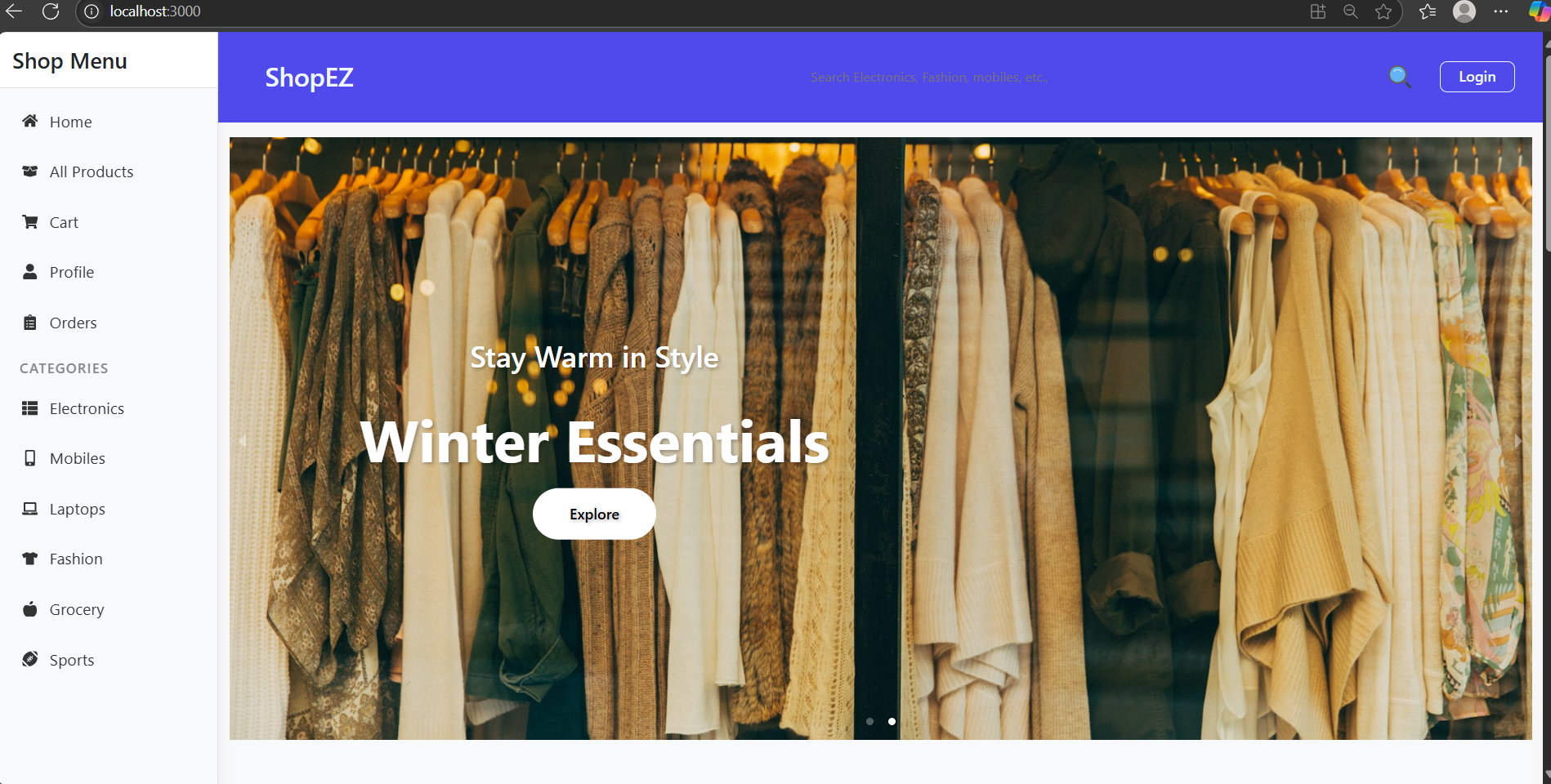
Running the Backend:



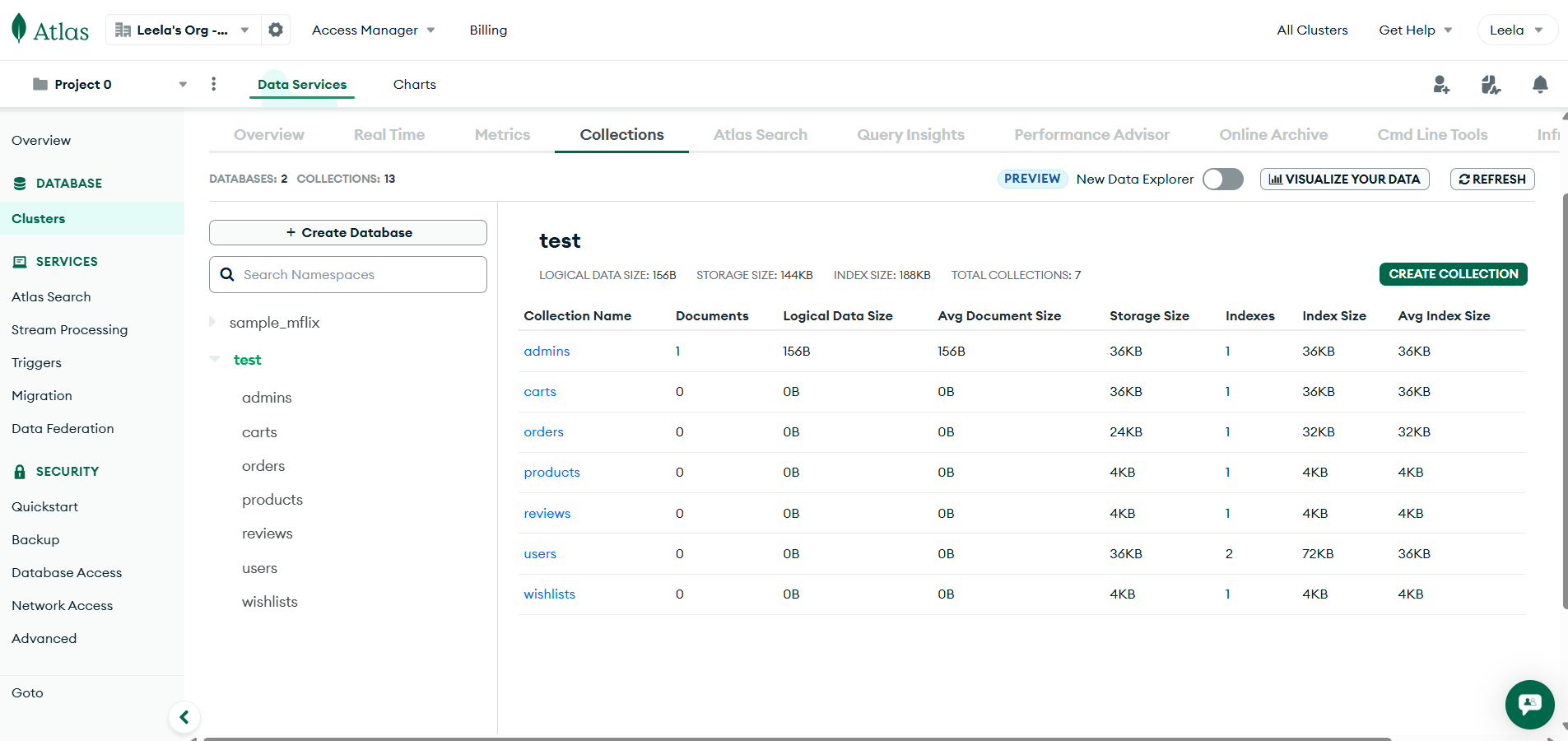
Running the Frontend:



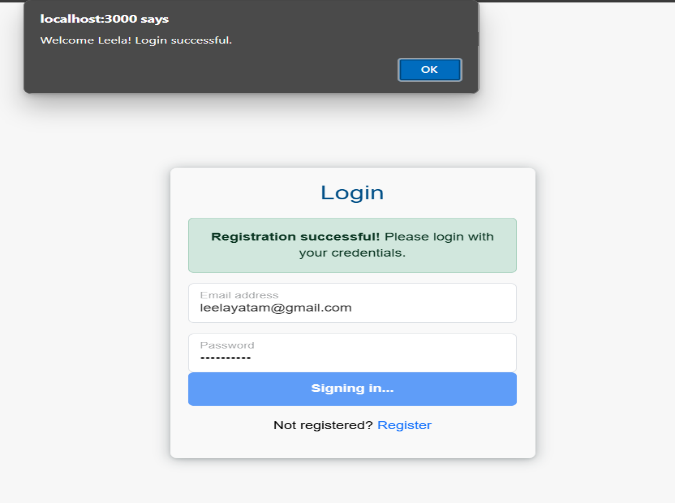
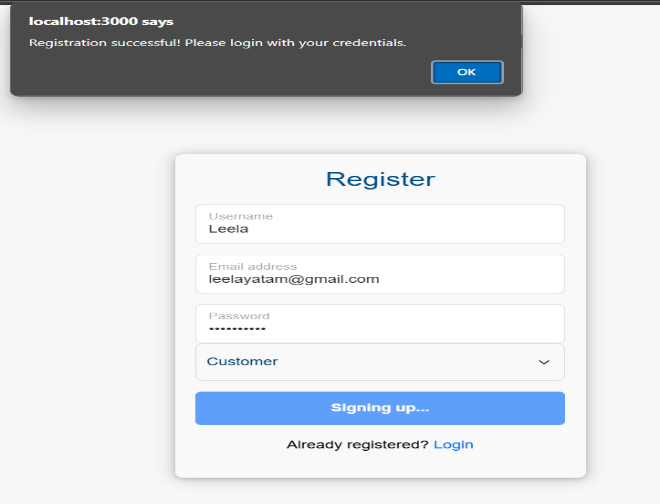
After Running the Frontend:



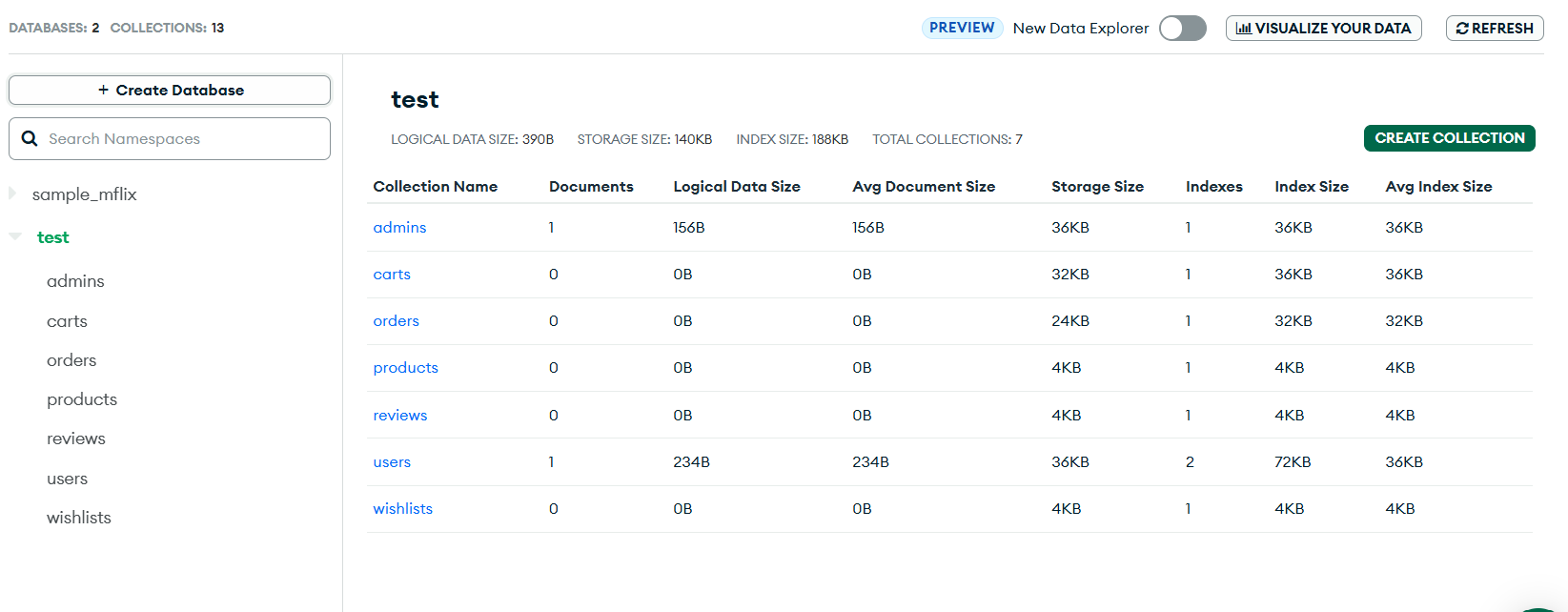
MangoDB Database:



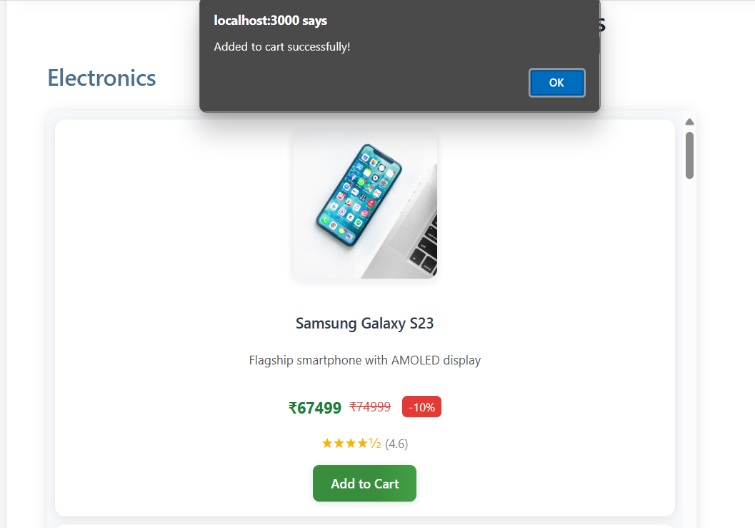
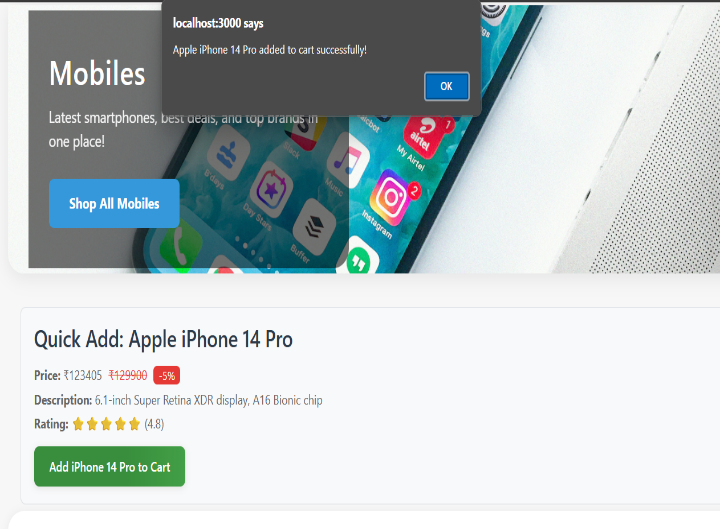
Register and Login Procedure:



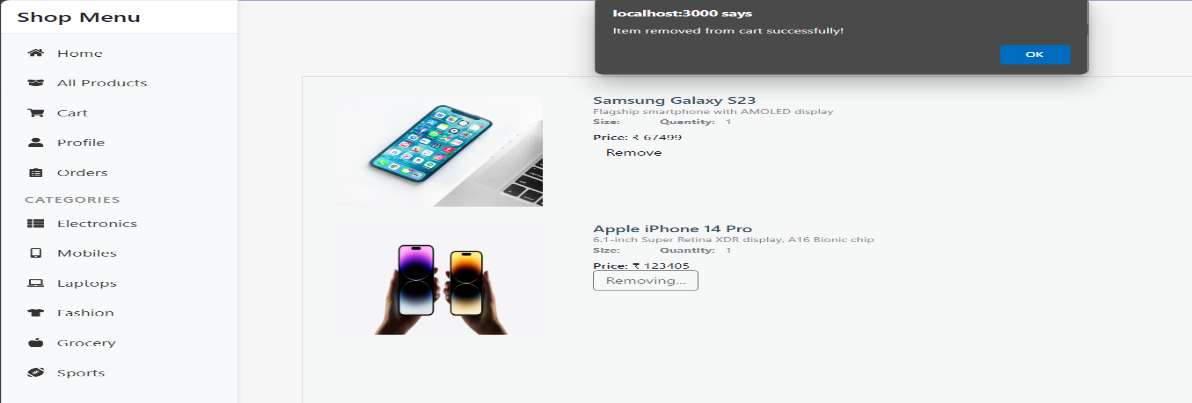
Checking in MongoDB: After Successfully Registered and Logged In



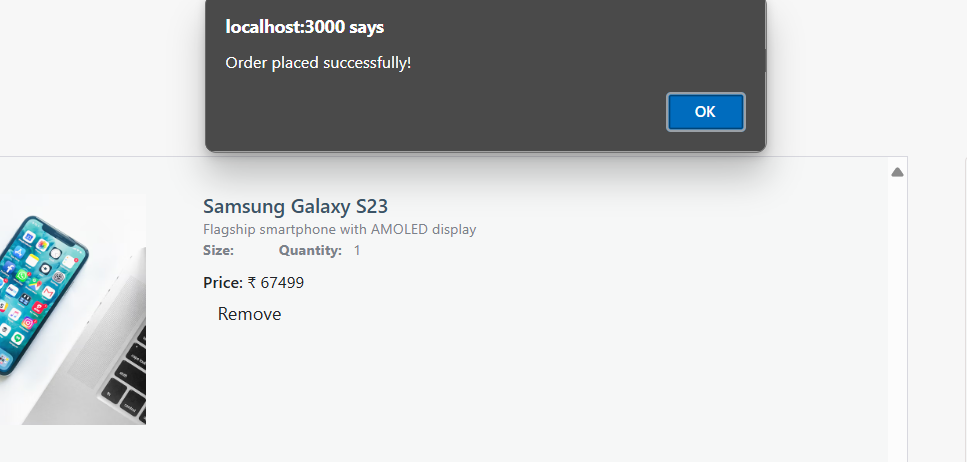
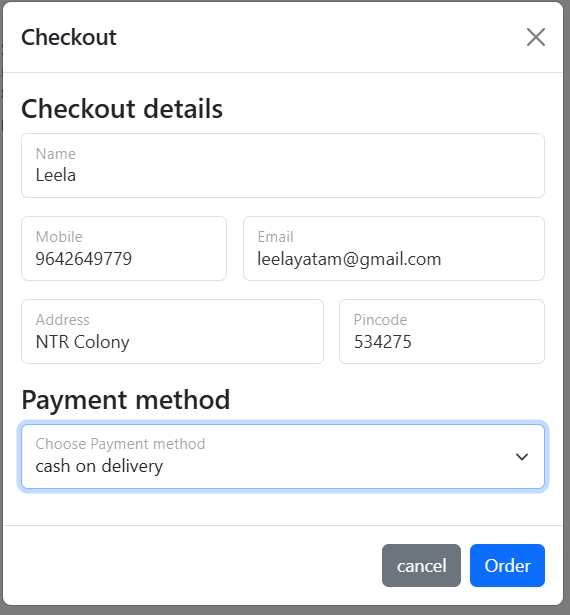
Adding Products to Cart:



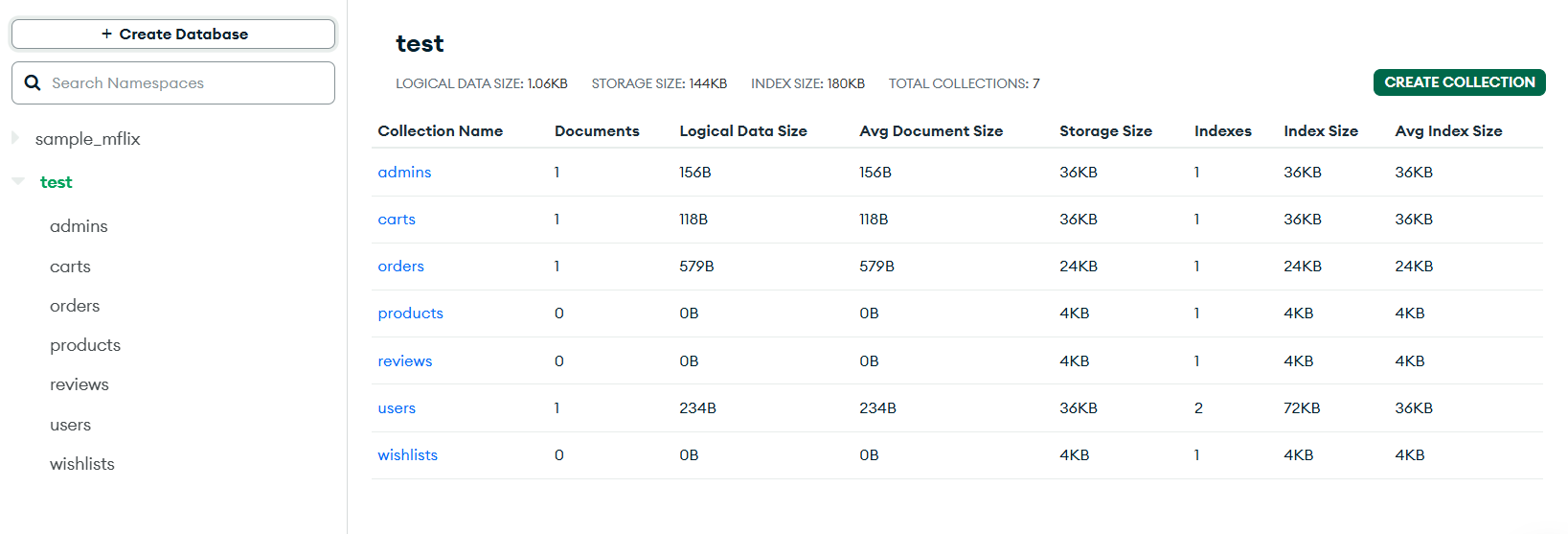
Removing a Product from Cart:



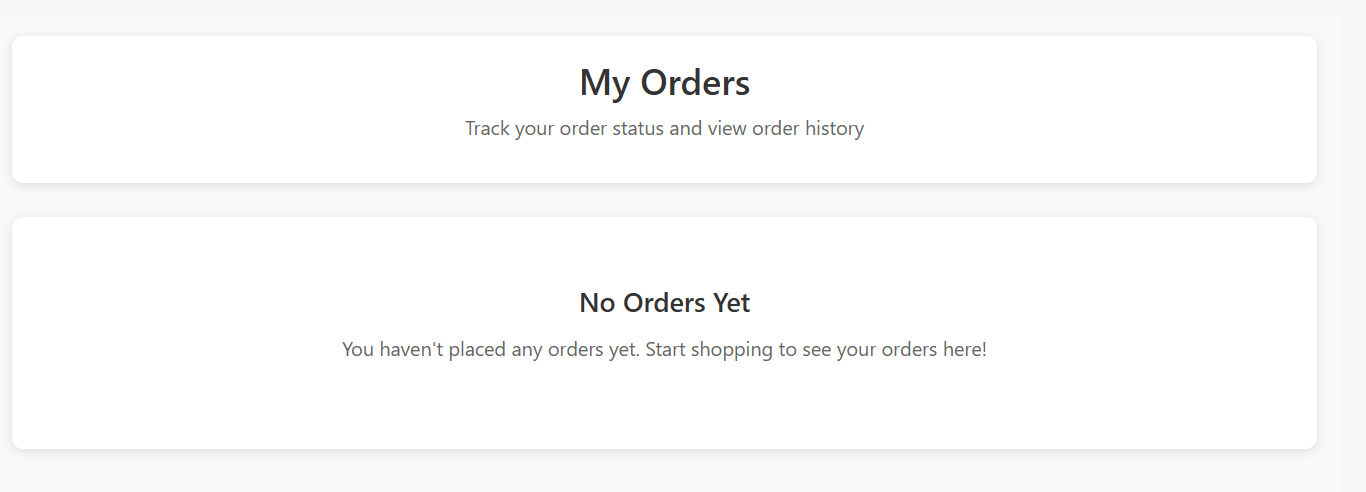
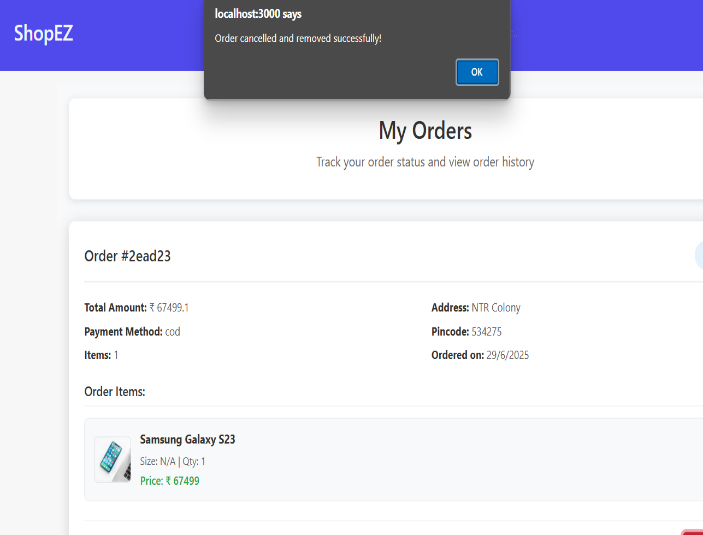
Placing the Order:



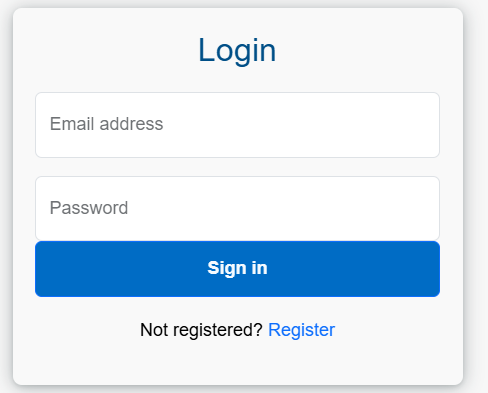
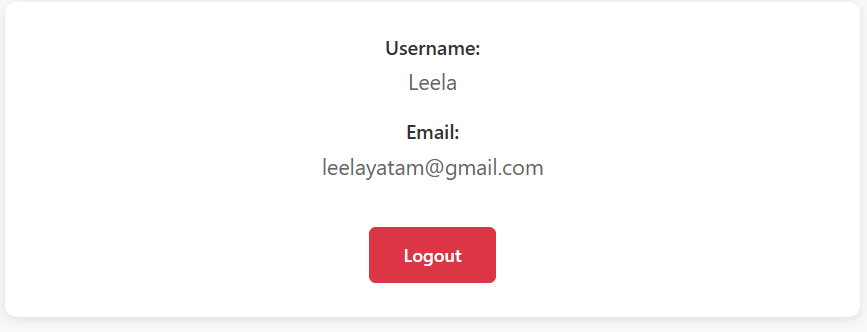
Checking in MongoDB: Successfully Order Placed



Cancelling an Order:



Logout:



**Demo Link:**

<https://drive.google.com/drive/u/0/folders/1F_NjLcdJKk-rz2jtqOxrtgZwFUoUvjgS>

**Future Enhancements:**

To ensure long-term usability and innovation, several enhancements are planned:

• Payment Gateway Integration: Integrate secure and reliable payment services such as Stripe, Razorpay, or PayPal to allow real-time payment processing within the platform.

• Real-Time Chat and Notifications: Implement socket-based communication for real-time updates on order status, chat support, or admin announcements.

• AI-Based Product Recommendations: Utilize machine learning to suggest personalized product recommendations based on user behavior, purchase history, and trending items.

• Progressive Web App (PWA) Support: Convert the web application into a PWA so users can install it on their devices, providing an app-like experience with offline support.

• Multi-language and Currency Support: Expand accessibility by supporting multiple languages and currencies, making the platform viable for global users.

• Enhanced Analytics Dashboard: Integrate visual analytics using tools like Chart.js or D3.js for admins to monitor user behavior, sales trends, and conversion rates.

• Voice Search and Smart Filters: Add voice-enabled search using Web Speech API and smart filters based on tags, categories, price, and availability.

• Mobile App Integration: Build cross-platform mobile applications using React Native for Android and iOS to extend reach and usability.

• Inventory Automation: Automate inventory updates by integrating with barcode scanners or IoT devices in physical stores.

• Social Media Sign-In & Sharing: Implement OAuth with Google, Facebook, etc., for easy login. Add shareable product links to promote engagement.