ShopMart

ShopMart is a full-stack e-commerce application that simplifies online shopping through seamless user interaction, robust backend services, and structured data management. Designed to provide a complete customer experience from product discovery to feedback collection, ShopMart is a scalable and modular platform ideal for learning full-stack development and deploying real-world online stores.

Overview

ShopMart mimics the core functions of a modern e-commerce system by implementing modules such as user registration, product listing, cart management, checkout and order tracking. The system leverages REST APIs built with Node.js and stores data in MongoDB, all tested using Thunder Client.

Features

Product Listing & Search

Users can explore all available products, apply filters, or search by keywords.

User Authentication

Register or login using email. Passwords are hashed securely using bcrypt.

Cart & Checkout

Users can add products to the cart, update quantities, and proceed to checkout with payment simulation.

Order Tracking

Users can view current and past orders, track delivery status, and cancel if needed.

Feedback System

After delivery, users can rate and review products for future buyers.

Admin Panel (Optional)

Admins can manage the product catalog, customer orders, and handle user permissions.

Tech Stack

- **Frontend**: HTML, CSS (optional admin panel)
- **Backend**: Node.js, Express.js
- **Database**: MongoDB (accessed via MongoDB Compass)
- **API Testing**: Thunder Client
- **Security**: bcrypt.js, JWT (authentication)
- **IDE**: Visual Studio Code
- **Version Control**: Git, GitHub

Architecture

- **Client Side**: Static or frontend components for UI interactions
- **Server Side**: Node.js REST APIs (register, login, cart, checkout, etc.)
- **Database**: MongoDB for storing users, products, orders, and reviews
- **Testing Tools**: Thunder Client for API testing, Postman-compatible

Project Structure

- server.js – Entry point of the application
- routes/ – All API endpoints for users, products, and orders
- models/ – MongoDB schemas for each entity
- controllers/ – Logic for handling each API route
- middleware/ – Authentication and error-handling functions
- config/ – MongoDB connection and JWT secrets
- package.json – Dependency and script definitions

Use Cases

- Learning full-stack app development
- Demonstrating database-backed CRUD operations
- Practice for building scalable backend systems

- Hands-on use of Express routing, middleware, and MongoDB integration

Contributors

- **Raahitya Illuri**
-

Goals Achieved

- Implemented end-to-end user flow: registration → product → cart → order → feedback
- Practiced secure authentication with hashed credentials and token-based sessions
- Structured RESTful APIs and modularized backend codebase
- Tested all routes via Thunder Client for reliability