**Full Stack Development with MERN**

**Project Documentation format**

**1. Introduction**

* **Project Title:** ShopEz – E-commerce Website
* **Team Members:**

A.Lathifunisa Tausifa [Team Leader]

E. Jasper

M. Janaki

R.Gowshika

**2. Project Overview**

* **Purpose:**

ShopEz is an interactive e-commerce platform designed to offer users a seamless online shopping experience. It allows browsing, filtering, and purchasing a variety of products with secure payment options.

* **Features:**
  + Browse products by categories and keywords
  + Advanced filtering options
  + Add to cart functionality
  + Secure payments using Stripe API
  + Address management for users
  + Admin panel for product and order management
  + User account system with password reset, recovery, and email notifications

**3. Architecture**

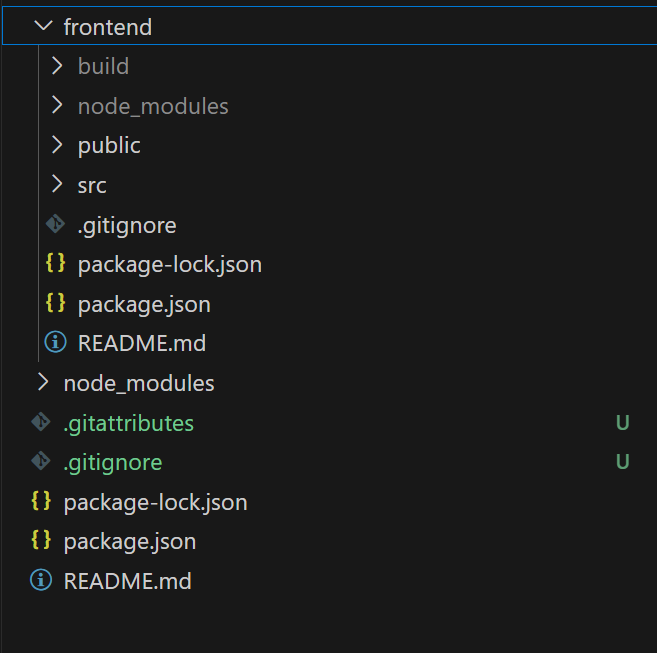
* **Frontend:** The React-based frontend ensures a responsive and interactive UI, utilizing React Router for navigation and Redux for state management.
* **Backend:** The backend, powered by Node.js and Express.js, provides RESTful APIs for user authentication, product management, and order processing , stripe for Payment and Mailtrap for sending emails.
* **Database:** MongoDB serves as the database, storing information about users, products, orders, and transactions. The schema is optimized for scalability and quick retrieval of data.

**4. Setup Instructions**

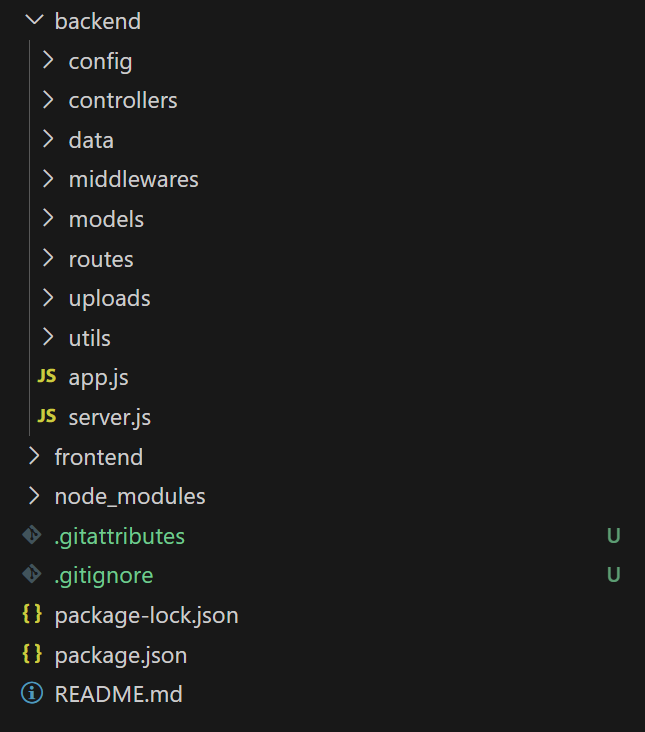
* **Prerequisites:**
* Node.js (v14 or later)
* MongoDB (locally or hosted)
* Git
* Stripe API credentials
* SMTP credentials through MailTrap
* **Installation:**
* Create a folder. With two subfolders for Frontend and Backend.
* In Backend, use command: **npm i mongodb express bcrypt jsonwebtoken**
* Also install other dependencies
* In Frontend Folder, use command**: npm i react**
* Add .env files in the client and server directories with necessary credentials (MongoDB URI, Stripe API key, etc.)

**5. Folder Structure**

* **Client:**



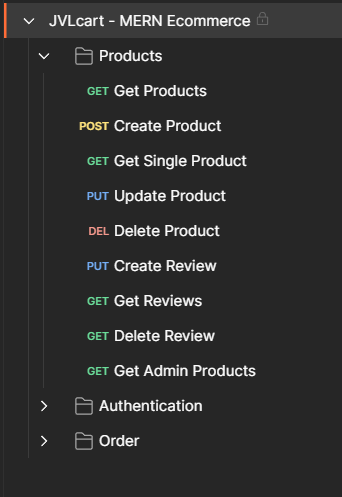
* **Server:**

****

* **6. Running the Application**
* Provide commands to start the frontend and backend servers locally.
  + **Frontend:** npm start in the client directory.
  + **Backend:** npm run dev in the server directory.

**7. API Documentation**

* API Endpoints for Products



* API Endpoints for Users

A screenshot of a black screen

Description automatically generated

* API Endpoints for Orders

A screenshot of a black screen

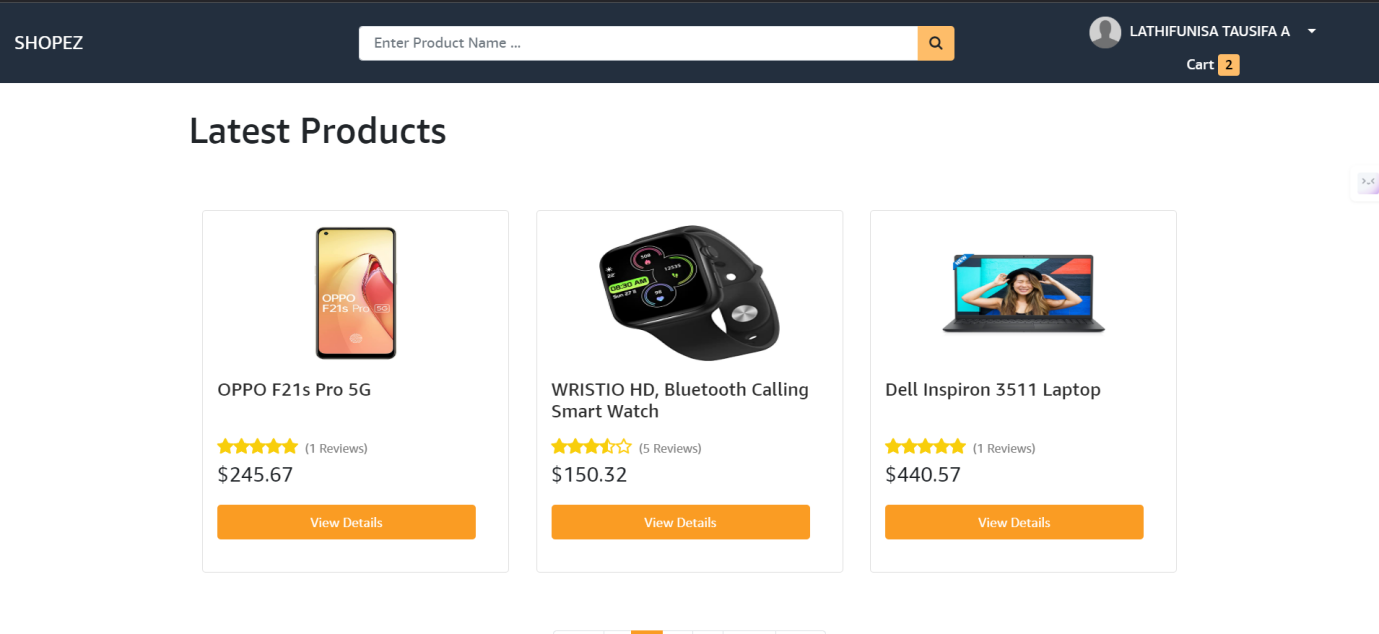
Description automatically generated

**8. Authentication**

* The application uses JSON Web Tokens (JWT) and Bcrypt for secure user authentication.
* Password recovery and reset processes are implemented with email verification via MailTrap
* The tokens are set to expire between 7 days
* Admins has several specific routes

**9. User Interface**

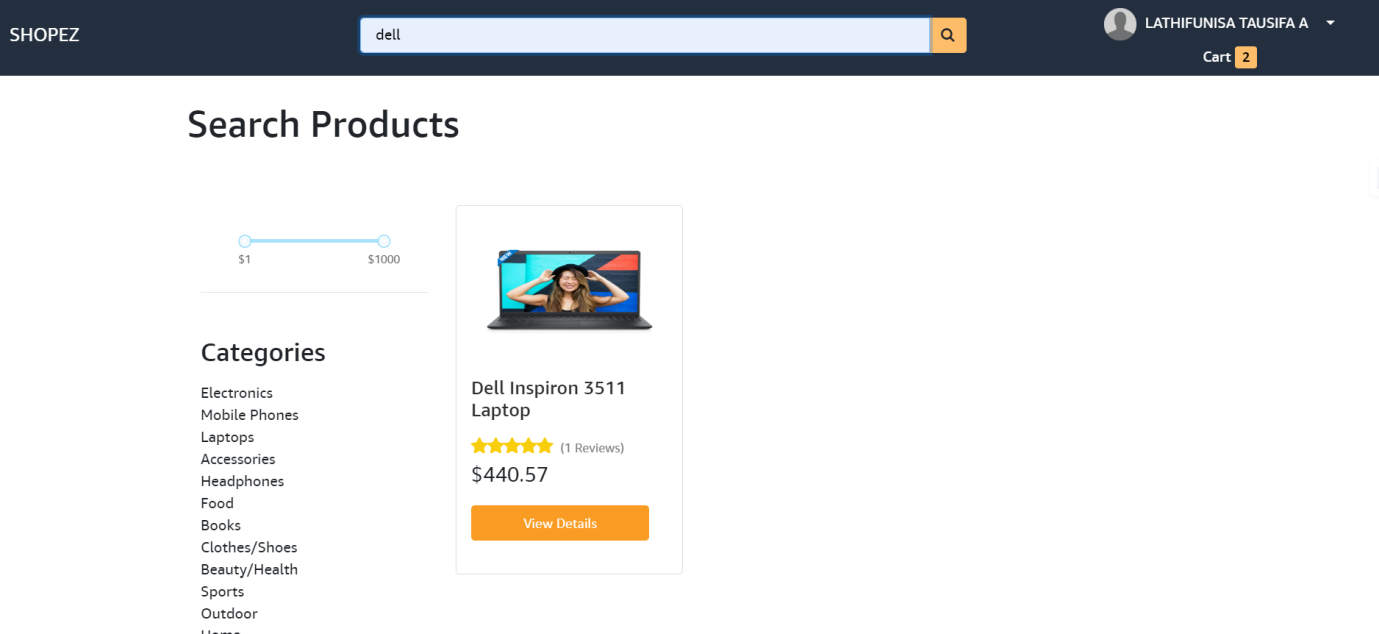
* The user interface is designed with modern styling libraries such as Tailwind CSS.
* Key features include a responsive navbar, product listing pages, and a dynamic shopping cart.
* Various features include searching, sorting, filtering, review writing, payment gateway and many more.



A screen shot of a phone

Description automatically generated

Searching , Filtering and Sorting



Review Submission

A screenshot of a computer

Description automatically generated

Orders

A screenshot of a computer

Description automatically generated

Shipping Information

A screenshot of a computer

Description automatically generated

Order Confirmation

A screenshot of a computer

Description automatically generated

Payment Gateway via Stripe

A screenshot of a computer

Description automatically generated

AUTHORISATION :

New Registration

**A screenshot of a computer screen

Description automatically generated**

Login

**A screenshot of a computer

Description automatically generated**

**Forgot password**

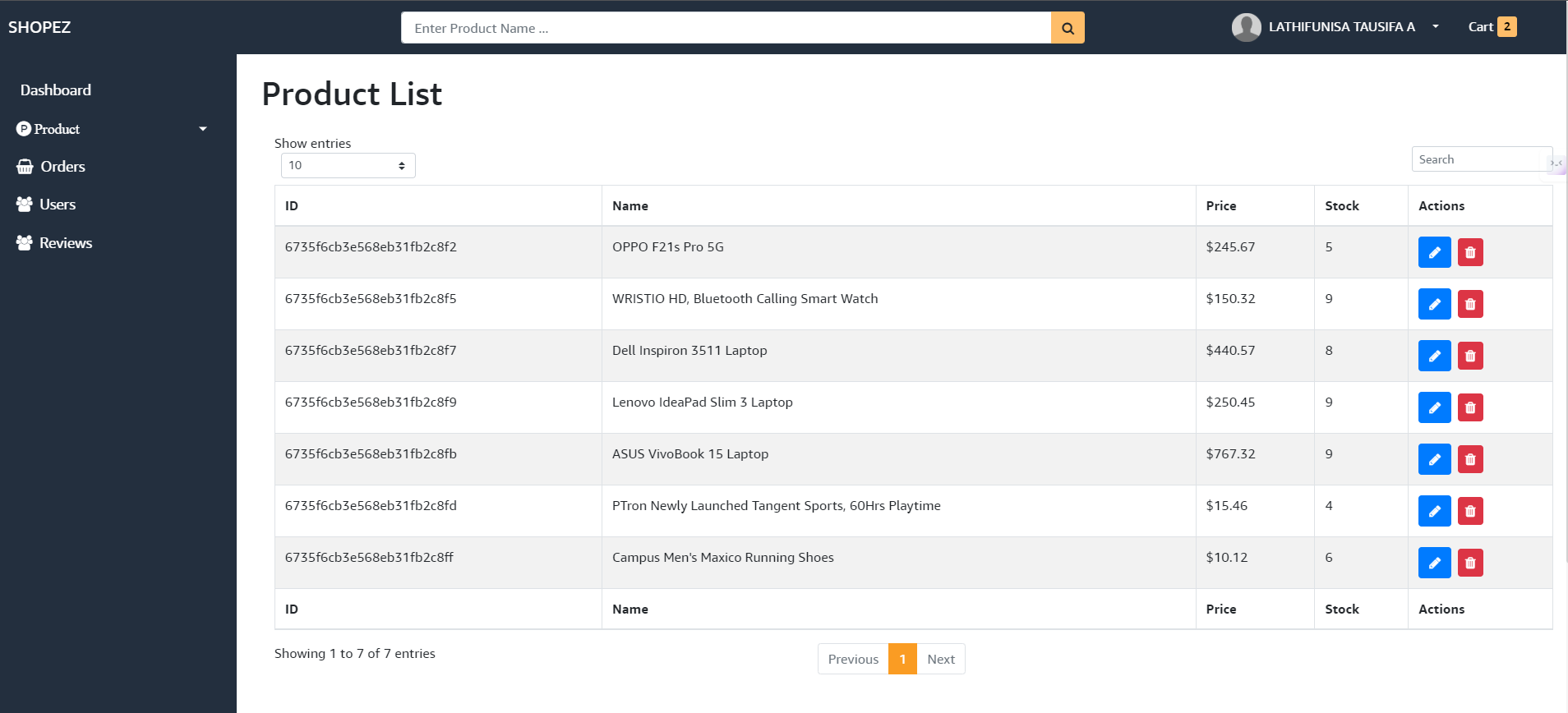
**A screenshot of a login form

Description automatically generated**

**Admin Dashboard**

**A screenshot of a computer

Description automatically generated**

****

**To add a new Product**

**A screenshot of a login form

Description automatically generated**

**10. Testing**

* Tools: Postman
* Unit tests are written for key components and APIs.
* End-to-end testing ensures the overall functionality of the system.

**11. Screenshots or Demo**

<https://drive.google.com/drive/u/1/folders/1Yf_A3s5OxxDJXHm4uL4lWfWy3h66ynJD> -The drive link to access the video presentation.

**12. Known Issues**

* Responsiveness needs to be improvised.

**13. Future Enhancements**

* Implement AI-driven product recommendations.
* Add a chatbot for customer support.
* Integrate real-time order tracking.