

Full Stack Development with MERN

Project Documentation format

1. Introduction

Project Title: ShopEZ-ECommerce Site

Team Members:

- **ATHITHYA ANIL** - Role: Project Manager
- **NANDINI J NAIR** - Role: Frontend Developer
- **SHIVAM SUNDE** - Role: Backend Developer
- **HARIKRISHNAN V D** - Role: Database Administrator

2. Project Overview

Purpose:

The purpose of this project is to develop a full-featured e-commerce website that allows users to browse, search, and purchase products online. The platform aims to provide a seamless shopping experience with secure payment processing and efficient order management.

Features:

- **User Authentication:** Secure login and registration using JWT.
- **Product Listings:** Browse and search for products.
- **Shopping Cart:** Add, remove, and update items in the cart.
- **Checkout Process:** Complete purchases with payment integration.
- **Admin Dashboard:** Manage products, orders, and users.

3. Architecture

Frontend:

The frontend is built using React.

- **Components:** Organized into reusable components like Header, Footer, ProductList, ProductDetail, Cart, and Checkout.
- **State Management:** State is managed using Context API for local state.
- **Routing:** React Router is used for client-side routing, with routes for home, product details, cart, checkout, and user account.

Backend:

The backend is built using Node.js and Express.js.

1. **API Endpoints:** RESTful API endpoints to handle user authentication, product retrieval, cart management, and order processing.

User/Admin Authentication:

- **POST /api/v1/auth/register** – To register a new user
- **POST /api/v1/auth/login** – To login into the website
- **POST /api/v1/auth/forgot-password** – If the user forgets his/her password they can update it.

User Management:

- **GET /api/v1/auth/orders** – to get the order details done by the user.
- **PUT /api/v1/auth/profile** – To update the users profile
- **GET/api/v1/product/braintree/payment** – take users to payment gateway
- **GET/api/v1/product/product-category/{params.slug}** – to get product by category
- **GET /api/v1/category/get-category** – To get all categories of products
- **GET /api/v1/product/product-list/{page}**– To get product list by loading more in the page
- **GET/api/v1/product/product-count** – To get total count of products
- **POST/api/v1/product/product-filters** – To filter the products
- **GET/api/v1/product/related-product** – To get similar product details
- **GET/api/v1/product/get-product/{params.slug}** -To get a particular product

Admin Management

- **GET/api/v1/auth/all-orders** – To get all the orders of the users
- **PUT/api/v1/auth/order-status/{orderId}** - to change/ update the status of the order done by the user
- **POST/api/v1/category/create-category** – to create category
- **GET/api/v1/category/update-category/{selected._id}**: to update category based on the id of the category
- **DELETE/api/v1/product/delete-category/{pid}** – to delete the category based on the id
- **POST/api/v1/product/create-product** – to create product
- **GET/api/v1/product/get-product** – to get all products
- **PUT/api/v1/product/update-product** – to update product
- **DELETE/api/v1/product/delete-product/{id}** – to delete product based on the id
- **GET/api/v1/product/search/{values.keyword}** to search products based on keywords
 - .

Database:

MongoDB is used as the database.

- **Collections:** users, products, orders, cartItems.
- **Schemas:**
 - User Schema: { name, email, password, role, address ,phone ,answer }
 - Product Schema: { name, description, price, category, quantity, photo }
 - Order Schema: { products , payment, status }

4. Setup Instructions

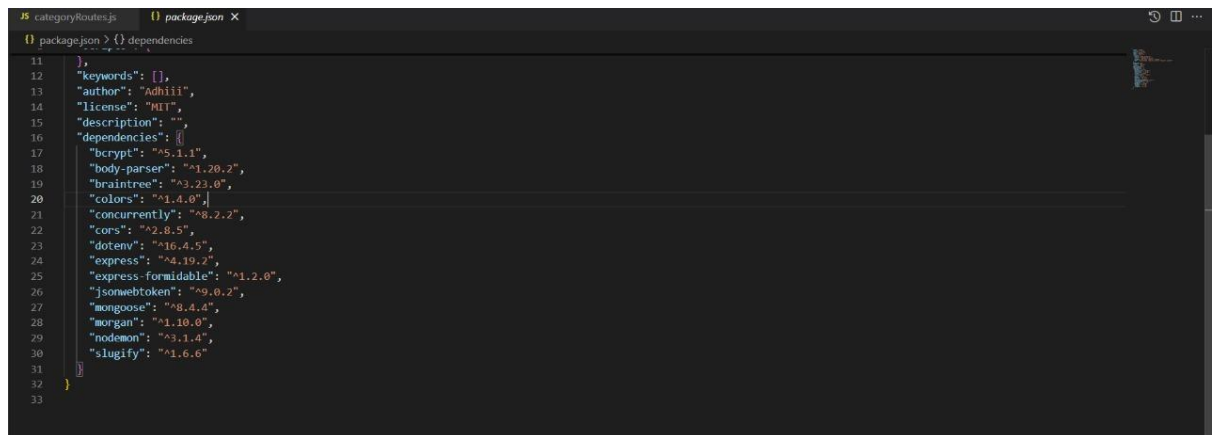
Prerequisites:

- Node.js
- MongoDB
- Express.js
- VS Code
- React

Installation:

Dependencies:

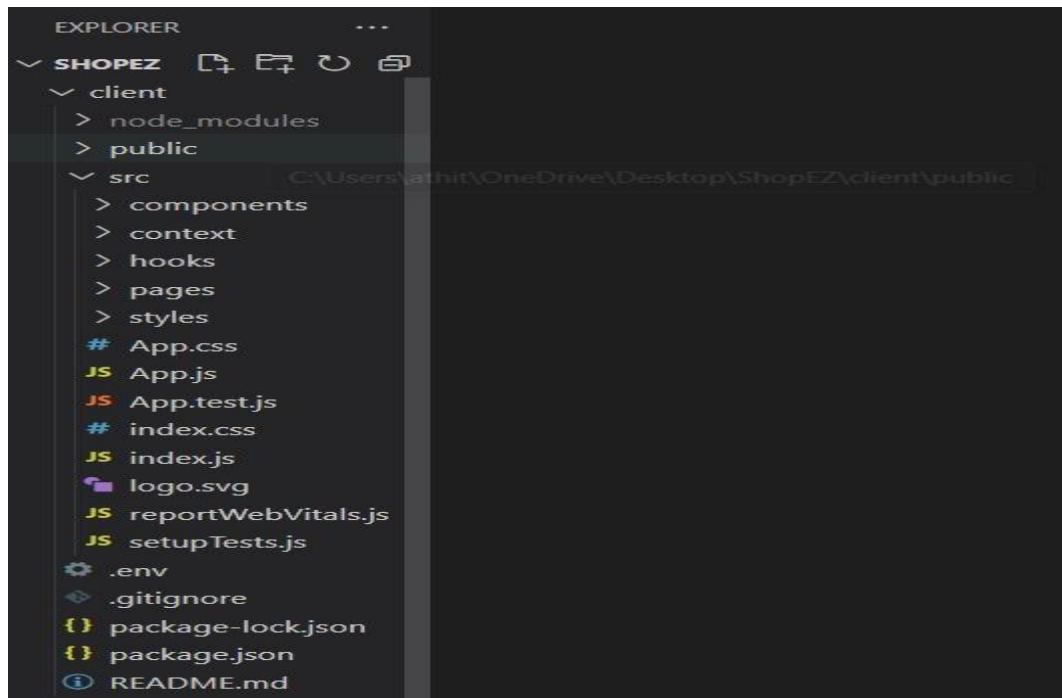
1. npm i bcrypt
2. npm i body-parser
3. npm i braintree
4. npm i colors
5. npm i concurrently
6. npm i cors
7. npm i dotenv
8. npm i express-formidable
9. npm i mongoose
10. npm i morgan



```
11 },
12 "keywords": [],
13 "author": "Adhiii",
14 "license": "MIT",
15 "description": "",
16 "dependencies": {
17   "bcrypt": "^5.1.1",
18   "body-parser": "^1.20.2",
19   "braintree": "^3.23.0",
20   "colors": "^1.4.0",
21   "concurrently": "^8.2.2",
22   "cors": "^2.8.5",
23   "dotenv": "^16.4.5",
24   "express": "^4.19.2",
25   "express-formidable": "^1.2.0",
26   "jsonwebtoken": "^9.0.2",
27   "mongoose": "^8.4.4",
28   "morgan": "^1.10.0",
29   "nodemon": "^3.1.4",
30   "slugify": "^1.6.6"
31 }
32 }
33 }
```

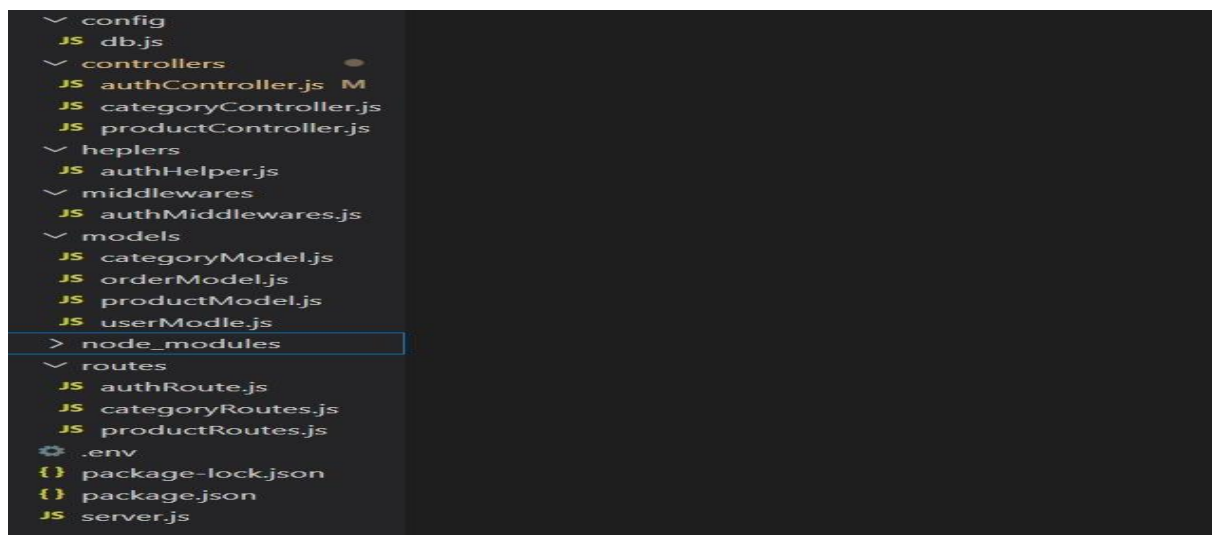
5. Folder Structure

- **Client:** Describe the structure of the React frontend.



Frontend part of the app contains the above elements. The layout of the app is decided by Form, Layout, Routes which comes under components. Hooks are used for getting categories of product while searching and filtering process. Under the page section many landing pages for admin ,user, categories, cart, payment, etc.

- **Server:** Explain the organization of the Node.js backend.



6. Running the Application

- Provide commands to start the frontend and backend servers locally.
 - **Frontend:** npm start in the client directory
 - **Backend:** npm run server /node server.js in the server directory.

- **Frontend and Backend(simultaneously):**`npm run dev`

7. API Documentation

- Document all endpoints exposed by the backend.
- Include request methods, parameters, and example responses.

User/Admin Authentication:

- **POST /api/v1/auth/register** – To register a new user
- **POST /api/v1/auth/login** - To login into the website
- **POST /api/v1/auth/forgot-password** – If the user forgets his/her password they can update it.

User Management:

- **GET /api/v1/auth/orders** – to get the order details done by the user.
- **PUT /api/v1/auth/profile** – To update the users profile
- **GET/api/v1/product/braintree/payment** – take users to payment gateway
- **GET/api/v1/product/product-category/{params.slug}** – to get product by category
- **GET /api/v1/category/get-category** – To get all categories of products
- **GET /api/v1/product/product-list/{page}**- To get product list by loading more in the page
- **GET/api/v1/product/product-count** – To get total count of products
- **POST/api/v1/product/product-filters** – To filter the products
- **GET/api/v1/product/related-product** – To get similar product details
- **GET/api/v1/product/get-product/{params.slug}** -To get a particular product

Admin Management

- **GET/api/v1/auth/all-orders** – To get all the orders of the users
- **PUT/api/v1/auth/order-status/{orderId}** - to change/ update the status of the order done by the user
- **POST/api/v1/category/create-category** – to create category
- **GET/api/v1/category/update-category/{selected._id}**: to update category based on the id of the category
- **DELETE/api/v1/product/delete-category/{pid}** – to delete the category based on the id
- **POST/api/v1/product/create-product** – to create product
- **GET/api/v1/product/get-product** – to get all products
- **PUT/api/v1/product/update-product** – to update product
- **DELETE/api/v1/product/delete-product/{id}** – to delete product based on the id
- **GET/api/v1/product/search/{values.keyword}** to search products based on keywords

8. Authentication

Authentication involves user registration and login, utilizing JSON Web Tokens (JWT) for secure authentication. During registration, user details are hashed with bcrypt and stored in MongoDB. Upon login, user credentials are verified, and a JWT is generated and sent to the client, typically stored in local storage or cookies, with tokens set to expire after a predefined period. Authorization is handled through middleware that checks for the JWT in request headers, verifying it using a secret

key. Role-Based Access Control (RBAC) is implemented by assigning roles to users and defining permissions based on these roles, ensuring that only authorized users can access certain routes and functionalities. This ensures that the application remains secure and only authenticated and authorized users can perform specific actions.

9. User Interface

The screenshot displays the SHOPEZ website's registration page. The header includes the SHOPEZ logo, a search bar, and navigation links for HOME, CATEGORIES, REGISTER, LOGIN, and a CART icon with a red notification badge. The main content area features a yellow background with a shopping cart and balloons. A white 'SIGNUP FORM' is centered, containing input fields for username (filled with 'user'), email (filled with 'user@user.com'), password (masked with dots), phone number (filled with '687898065787'), location (filled with 'Mumbai'), and a nickname (filled with 'Spiderman'). A 'SignUp' button is at the bottom of the form. The footer contains the text 'All Right Reserved © ShopEZ' and links for 'About', 'Contact', and 'Privacy Policy'.

The screenshot displays the SHOPEZ website's user profile page. The header is identical to the previous page, but the 'USER' link in the navigation bar is active. On the left, a 'Dashboard' sidebar lists 'Profile' (highlighted in blue) and 'Orders'. The main content area has a yellow background with a shopping cart and balloons. A white 'User Profile' form is centered, with input fields for username (filled with 'user'), email (filled with 'user@user.com'), password (labeled 'Enter Your Password' and filled with '123456789'), location (filled with 'mumbai'), and a nickname (filled with 'Spiderman'). An 'Update' button is at the bottom of the form. The footer contains the text 'All Right Reserved © ShopEZ'.



SHOPEZ

[Search](#)[HOME](#)[CATEGORIES ▾](#)[REGISTER](#)[LOGIN](#)[CART](#)

LOGIN FORM

harkishman

From this website

Manage Passwords

LOGIN

All Right Reserved © ShopEZ

[About](#) | [Contact](#) | [Privacy Policy](#)



SHOPEZ

[Search](#)[HOME](#)[CATEGORIES ▾](#)[USER ▾](#)[CART](#)

Filter By Category

- ☐ wall clocks
☐ watches

Filter By Price

- ☐ Rs 0 to 160
☐ Rs 161 to 500
☐ Rs 500 to 1000
☐ Rs 1000 to 5000
☐ Rs 5000 to 10000
☐ Rs 10000 to 100000

RESET FILTERS

All Products



Apple Watch Series 9

Apple Watch Series 9 [GPS + Ce...

Rs 49999

[More Details](#)

[ADD TO CART](#)



Prowatch ZN

Prowatch ZN with 2 Year Warran...

Rs 9999

[More Details](#)

[ADD TO CART](#)

Loadmore



SHOPEZ

[Search](#)[HOME](#)[CATEGORIES ▾](#)[USER ▾](#)[CART](#)

Product Details



Name : Apple Watch Series 9

Description : Apple Watch Series 9 [GPS + Cellular 45mm] Smartwatch with Starlight Aluminum Case with Starlight Sport Band S/M. Fitness Tracker, Blood Oxygen & ECG Apps, Always-On Retina Display, Water Resistant


Price : 49999

Category : watches

[ADD TO CART](#)

Similar Products







SHOPEZ

[HOME](#)
[CATEGORIES](#)
[USER](#)
[CART](#)

Dashboard

[Profile](#)
[Orders](#)

All Orders

#	Status	Buyer	date	Payment	Quantity
1	Not Process	user	a few seconds ago	Success	2
<div>  <div> <p>Giordano Fashionista Collection Analog Bracelet Wrist Watch for Women with Color Variant</p> <p>Product details Case diamete</p> <p>Price : 100000</p> </div> </div>					
<div>  <div> <p>Giordano Fashionista Collection Analog Bracelet Wrist Watch for Women with Color Variant</p> <p>Product details Case diamete</p> <p>Price : 100000</p> </div> </div>					

https://drive.google.com/file/d/1N_jRPmgulbBtUOeD1jxIU3WjTzieneoi/view?usp=sharing

12. Known Issues

- Payment through Paypal is not responsive

13. Future Enhancements

- Comparison of two or more products
- Order Tracking