STOPPERS’S SHOP

STOPPER’S SHOP

STOPPERSHOP.VERCEL.APP

ONE STOP SOLUTION FOR YOUR ANYDAY SHOPPING, JUST VISIT STOPPER’S SHOP

OR

WANT TO BUCKET SOME EXCLUSIVE OUT OF THE BOX STUFFS FOR YOUR LOVED ONES, VISIT STOPPER’S SHOP

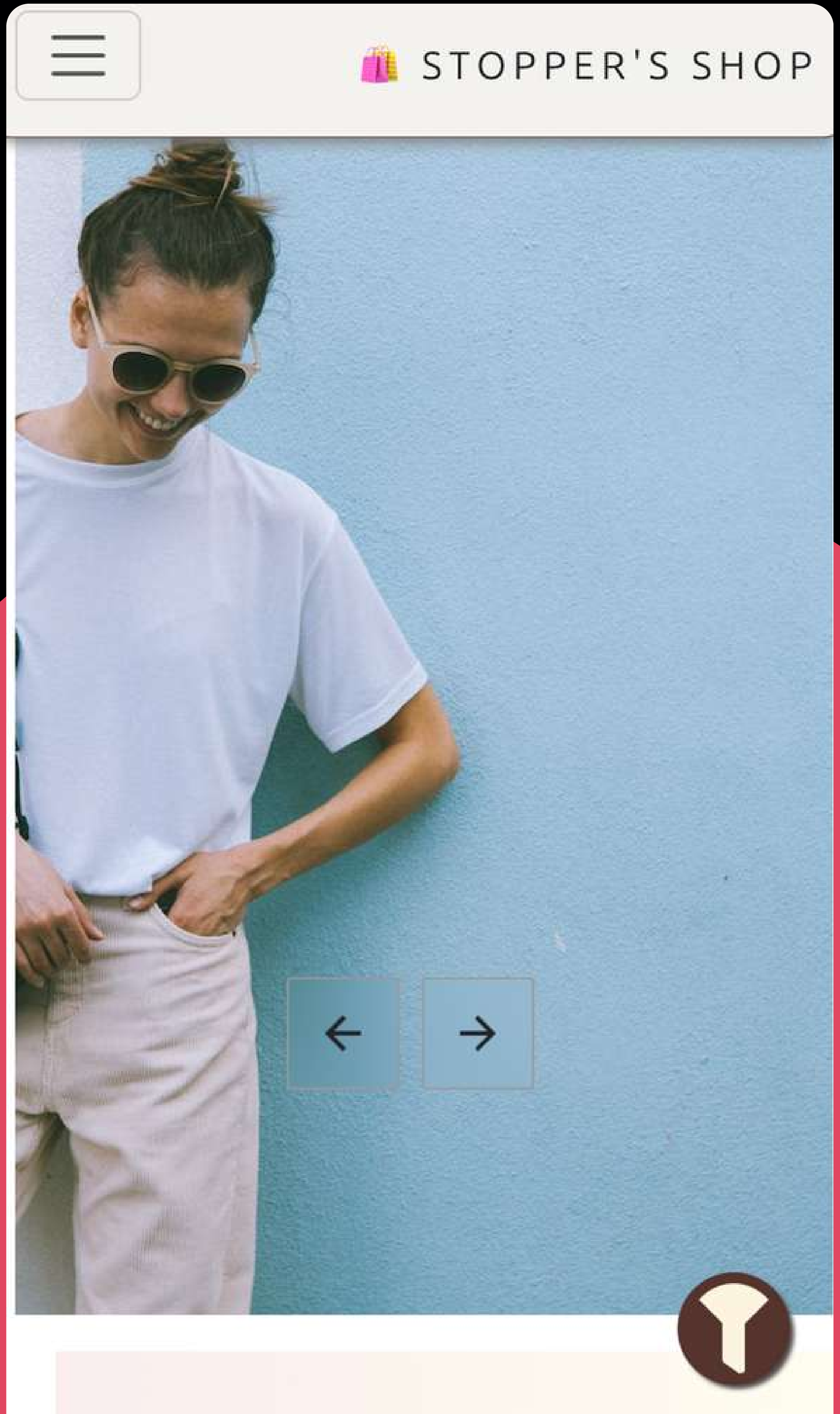
TEAM NAME- CODE TECHIES

MEMBERS -

- RIYA GUPTA ~ 2110991186
- ANKIT MISHRA ~ 2110991966
- RISHAV KUMAR ROY ~ 2110991165

GROUP- 19

SUBMITTED TO- ASHWANI SIR



IDEATION

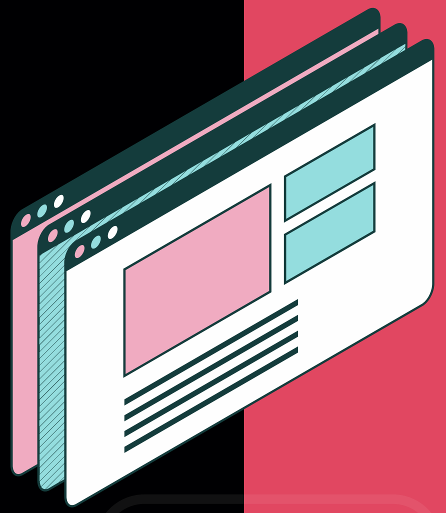
"wanted to create something that can meet a real life scenario with actual userbase"

"ecommerce has always been on our checklist since a website like this serves for different purposes according to need of the owner"

"that's why we decided to create such a platform where the owner can easily list their products & users can shop for the same watching movies at home. "

STOPPERS'S SHOP

APPROACH AND IMPLEMENTATION



Frontend

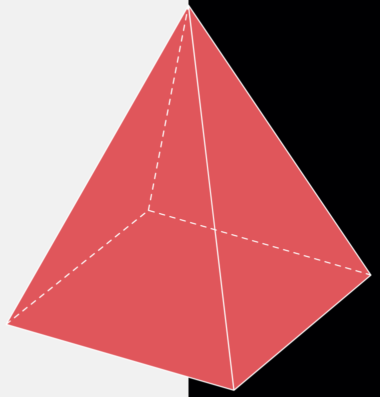
Built a cool and responsive UI using React-JS , Bootstrap, Material UI & CSS.

Database

Used MongoDb for storing different type of collections like user, products, orders etc.

Backend

Developed an Express.js server with routes, controllers and middlewares.



STOPPERS'S SHOP

MODELS & SCHEMA

1. User Model

- Stores the User details along with the time in the DB.
- Defines the datafields that are required and their respective datatypes along with the role assigned .

2. Category Model

- Stores the multiple category details created by the store administrator.
- Contains the info about individual categories of the website.

3. Order Model

- Represents an Order schema that gets created for an individual user as well as the admin if a order is placed for their products.
- Contains fields like products, payment-status ,buyers and delivery status .

4.Product Model

- Represents the Product schema that gets created when a new product is listed by the admin.
- Defines datafields like product-name, description, price, img source ,available quantity and category of the product.

TECH STACK

->FRONTEND

- REACT-JS
- BOOTSTRAP
- MATERIAL UI
- CSS

->BACKEND+DB

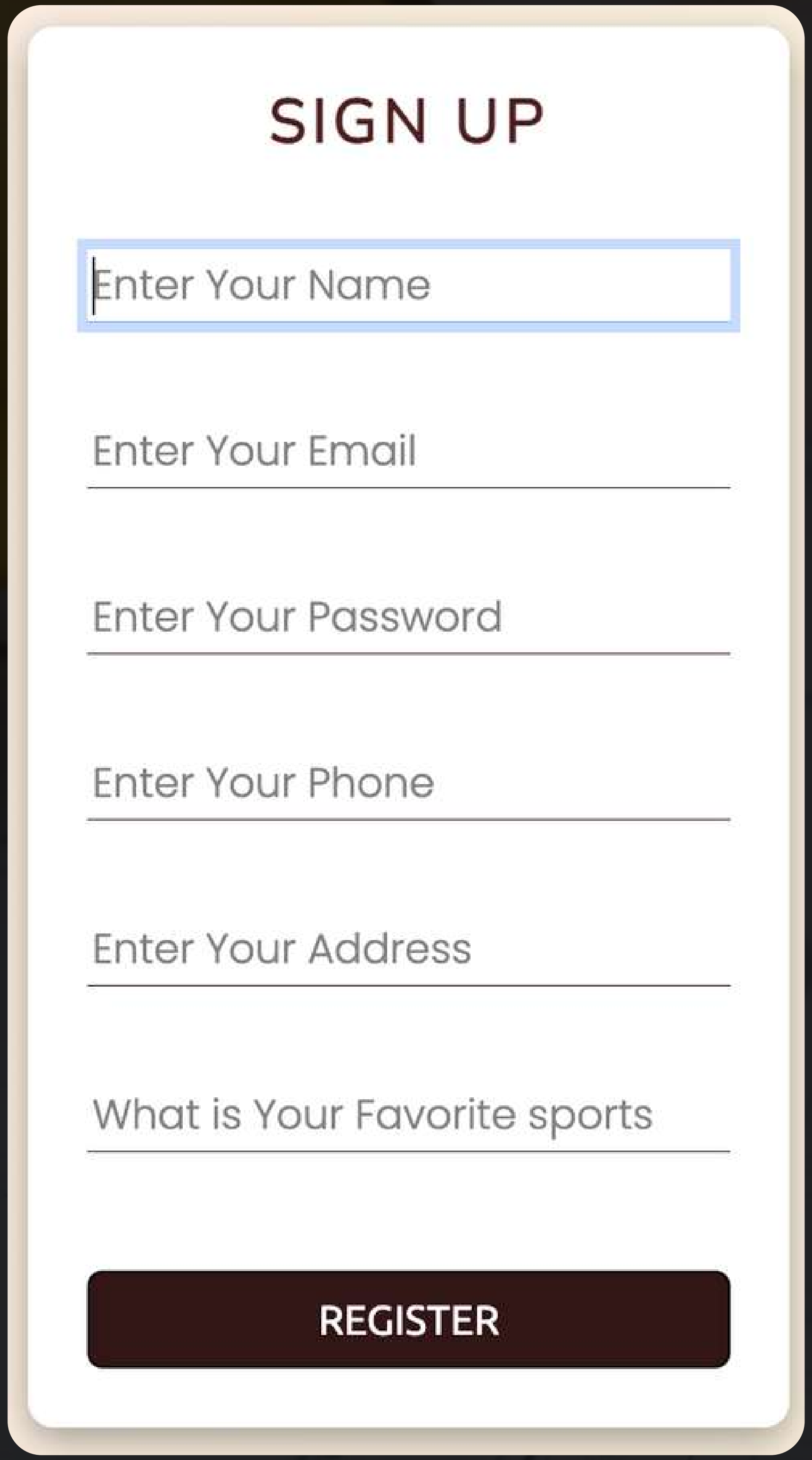
- EXPRESS JS
- MONGO DB
- NODE

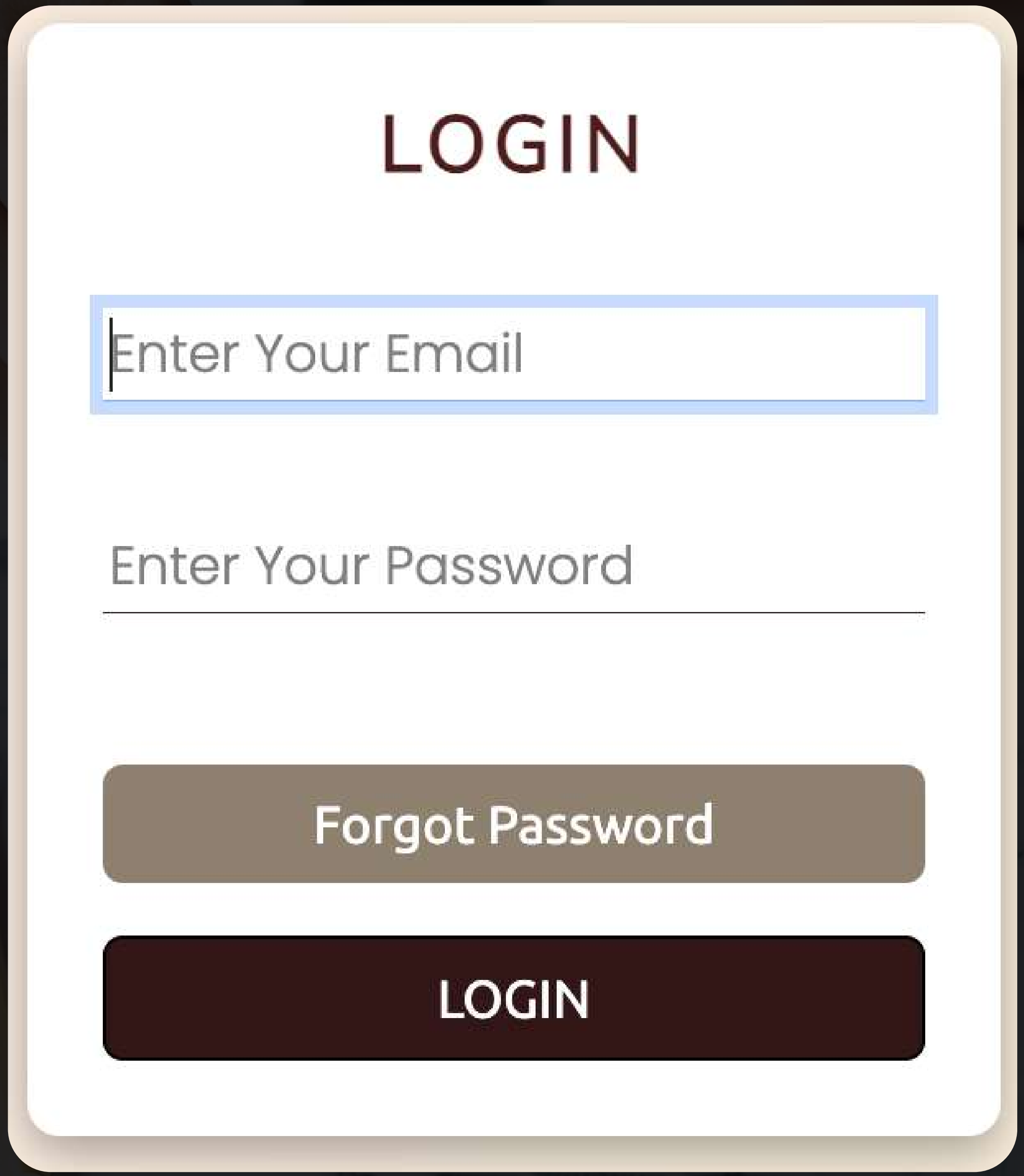
DEPENDENCIES

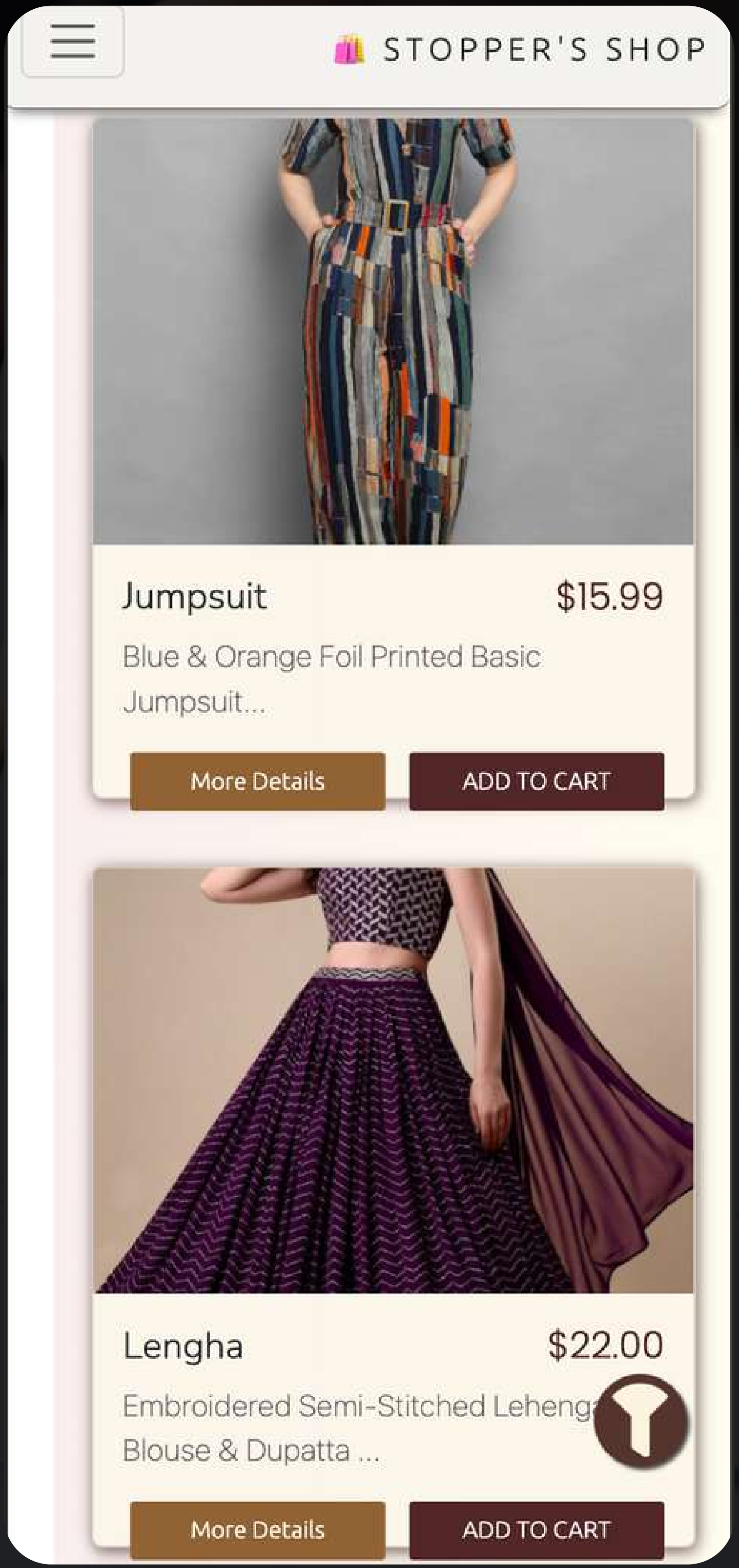
- braintree
- bcrypt
- dotenv
- cors
- react-hot-toast
- react-icons
- jsonwebtoken
- nodemon
- mongoose
- ant-design
- axios

STOPPERS'S SHOP

PROJECT OVERVIEW







Jumpsuit $15.99

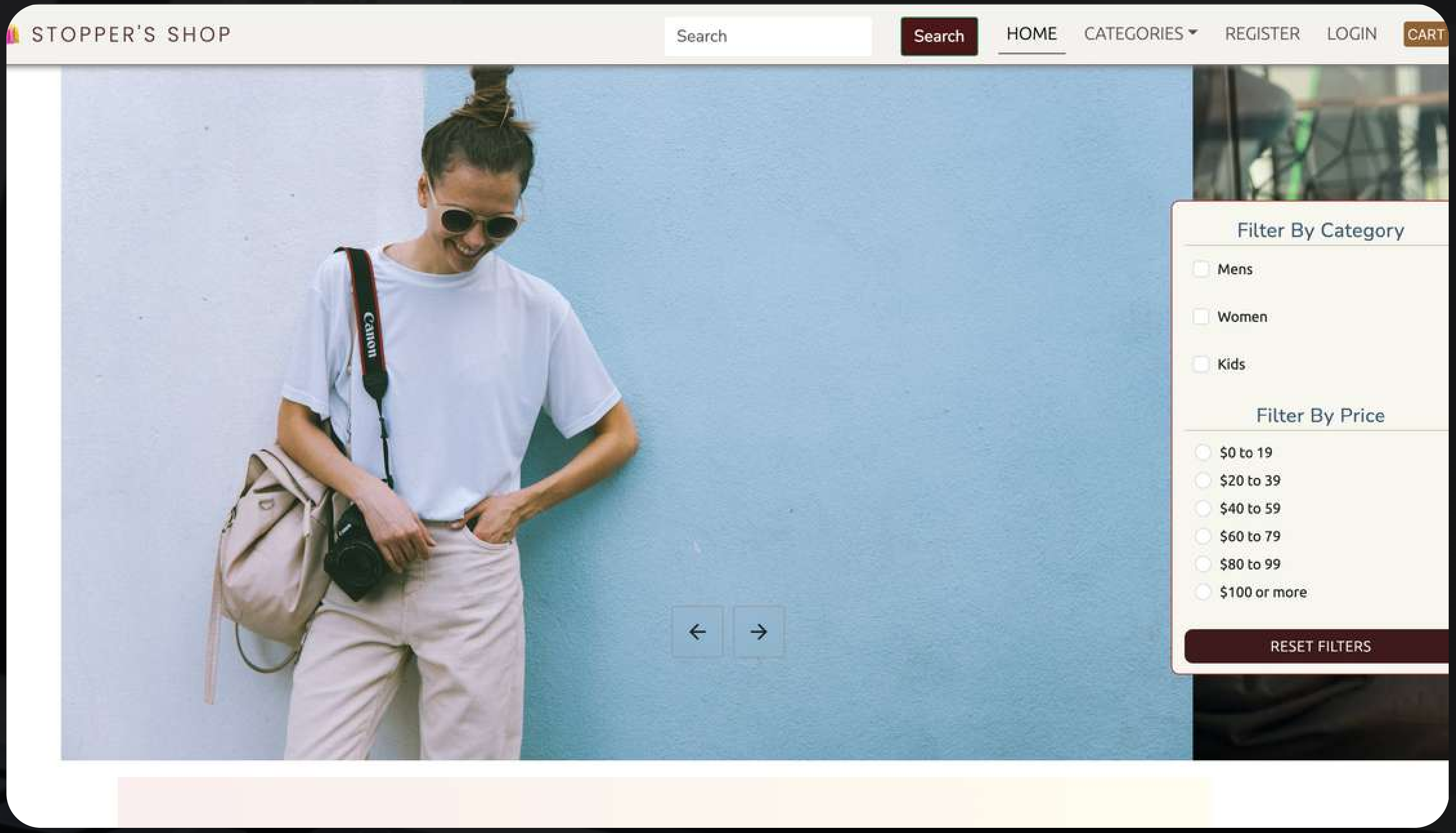
Blue & Orange Foil Printed Basic Jumpsuit...

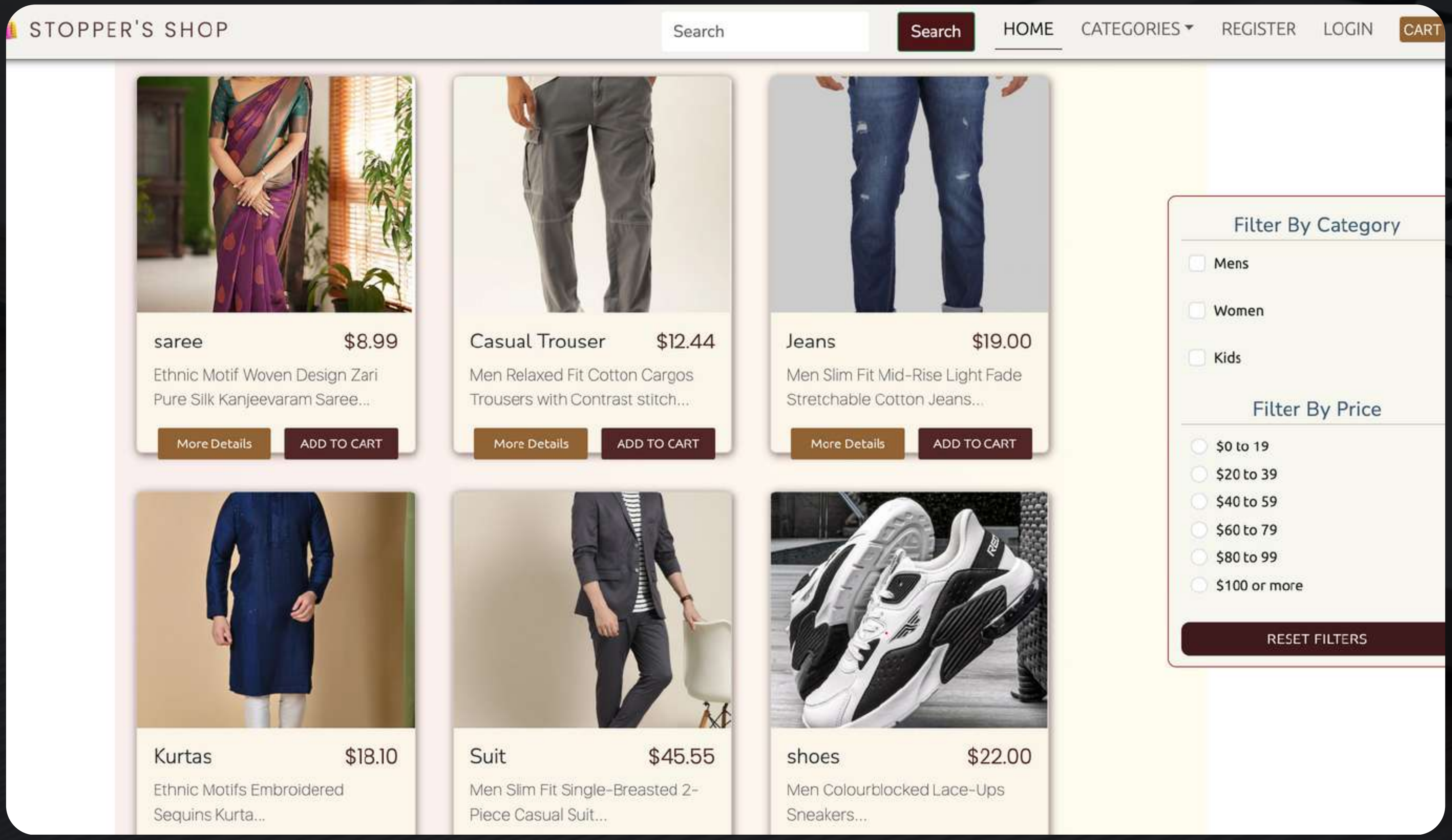
Lengha $22.00

Embroidered Semi-Stitched Leheng Blouse & Dupatta ...

STOPPERS’S SHOP

- THE HOME PAGE WITH ALL THE LISTED PRODUCTS AND EASY TO USE NAVIGATION.
- USER CAN ADD USE FILTER AS PER THEIR REQUIREMENTS.
- THE PRODUCTS CARDS ARE LISTED JUST BELOW THE SLIDER ANIMATION WITH THEIR RESPECTIVE DETAILS.





saree $8.99
Ethnic Motif Woven Design Zari Pure Silk Kanjeevaram Saree...
More Details | ADD TO CART

Casual Trouser $12.44
Men Relaxed Fit Cotton Cargos Trousers with Contrast stitch...
More Details | ADD TO CART

Jeans $19.00
Men Slim Fit Mid-Rise Light Fade Stretchable Cotton Jeans...
More Details | ADD TO CART

Kurtas $18.10
Ethnic Motifs Embroidered Sequins Kurta...

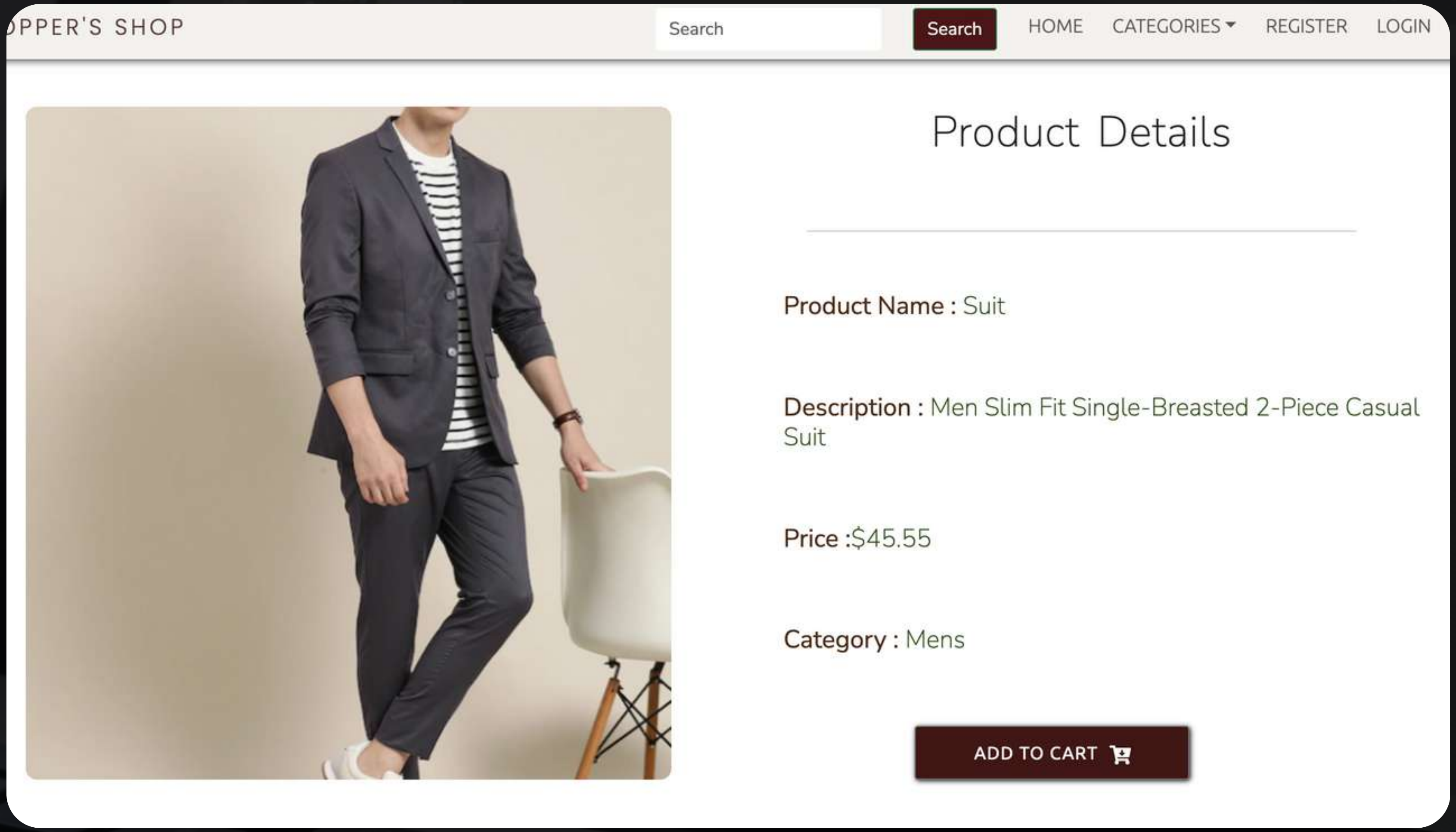
Suit $45.55
Men Slim Fit Single-Breasted 2-Piece Casual Suit...

shoes $22.00
Men Colourblocked Lace-Ups Sneakers...

- THE PRODUCTS LIST GETS UPDATED AS PER THE FILTER SELECTIONS .
- THE MORE DETAILS BUTTON NAVIGATES TO THE PRODUCT PAGE DIRECTLY WITH THE COMPLETE INFO OF THE PRODUCT.
- IN CASE IF THE USER DIRECTLY WANTS TO PURCHASE A PRODUCT, HE/SHE JUST NEED TO CLICK ON THE ADD TO CART BUTTON.

STOPPERS'S SHOP

- THIS IS THE PRODUCT DETAILS PAGE FOR EACH PRODUCT AND THEIR EACH INFO.
- USER CAN USE THE ADD TO CART BUTTON IN CASE THEY WANT TO BUY THAT PRODUCT.
- SIMILAR PRODUCTS ARE SHOWN JUST BELOW THIS PRODUCT DETAILS CARD.



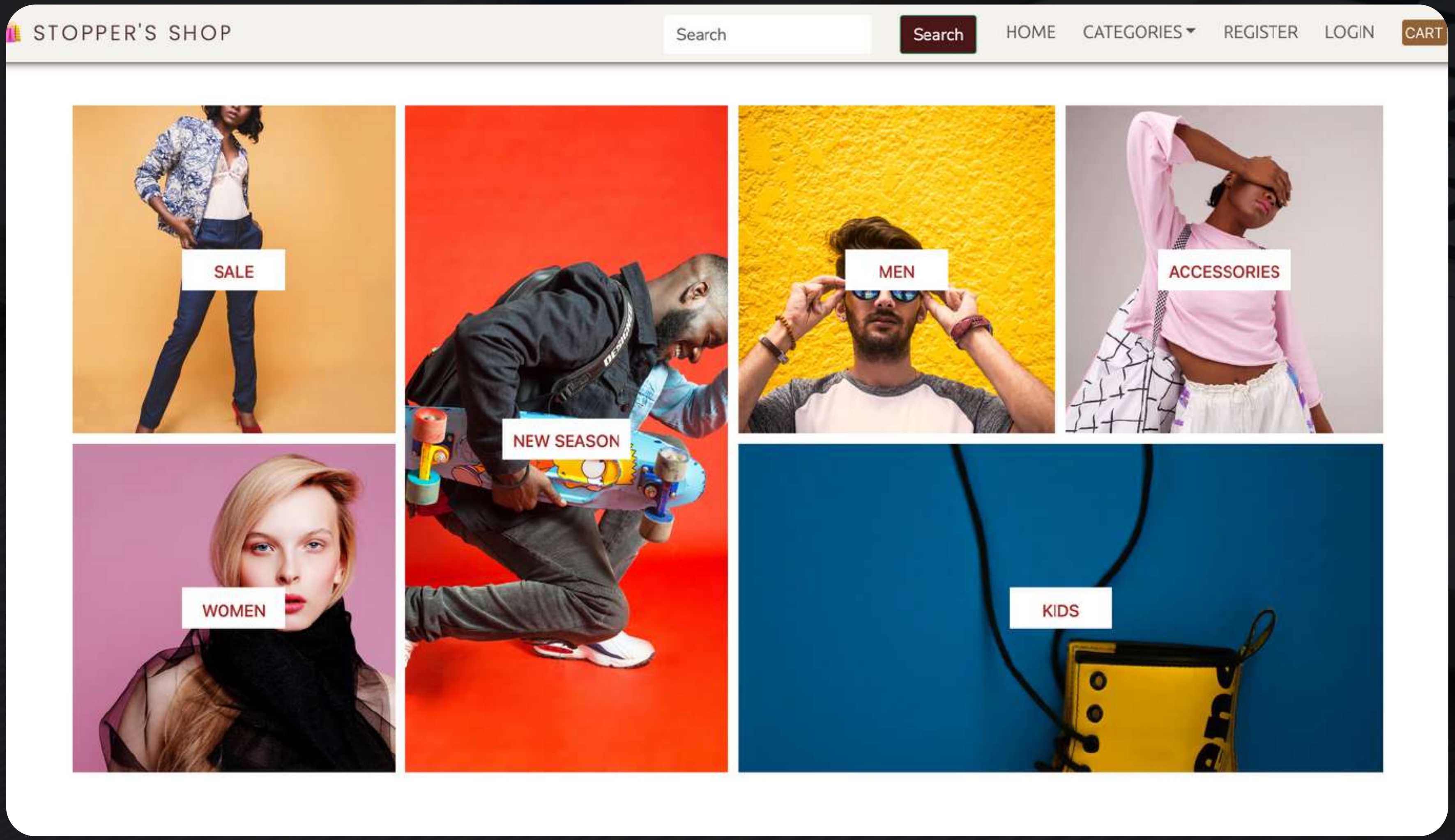
Product Details

Product Name : Suit

Description : Men Slim Fit Single-Breasted 2-Piece Casual Suit

Price :$45.55

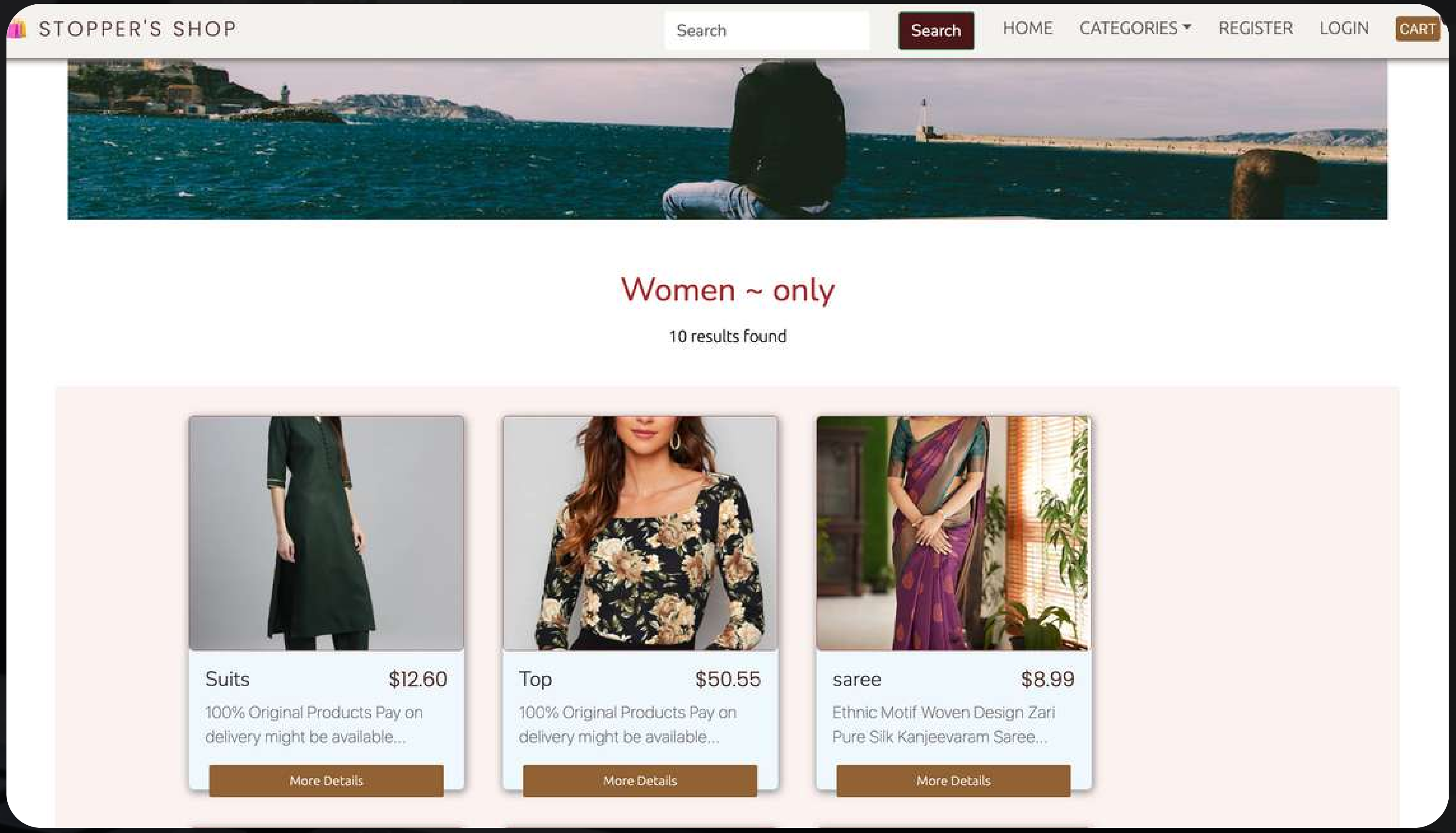
Category : Mens



- USERS CAN USE THE NAVIGATION MENU AND CLICK ALL CATEGORIES TO VISIT THIS PAGE .
- EACH CARD TAKES YOU TO A PARTICULAR CATEGORY PAGE WITH ALL PRODUCTS BELONGING TO THAT CATEGORY.
- EXTRA CATEGORIES ARE JUST PRESENT BELOW THIS VIEWPORT , GET SHOWN WHEN SCROLLED.

STOPPERS’S SHOP

- THE INDIVIDUAL CATEGORY PAGES SHOWS PRODUCTS BELONGING TO A PARTICULAR CATEGORY.
- FOR EX- IF A USER CLICKS ON THE WOMEN CATEGORY ,THEN THIS PAGE GETS OPENED WITH ALL THE PRODUCTS OF WOMEN CATEGORY..



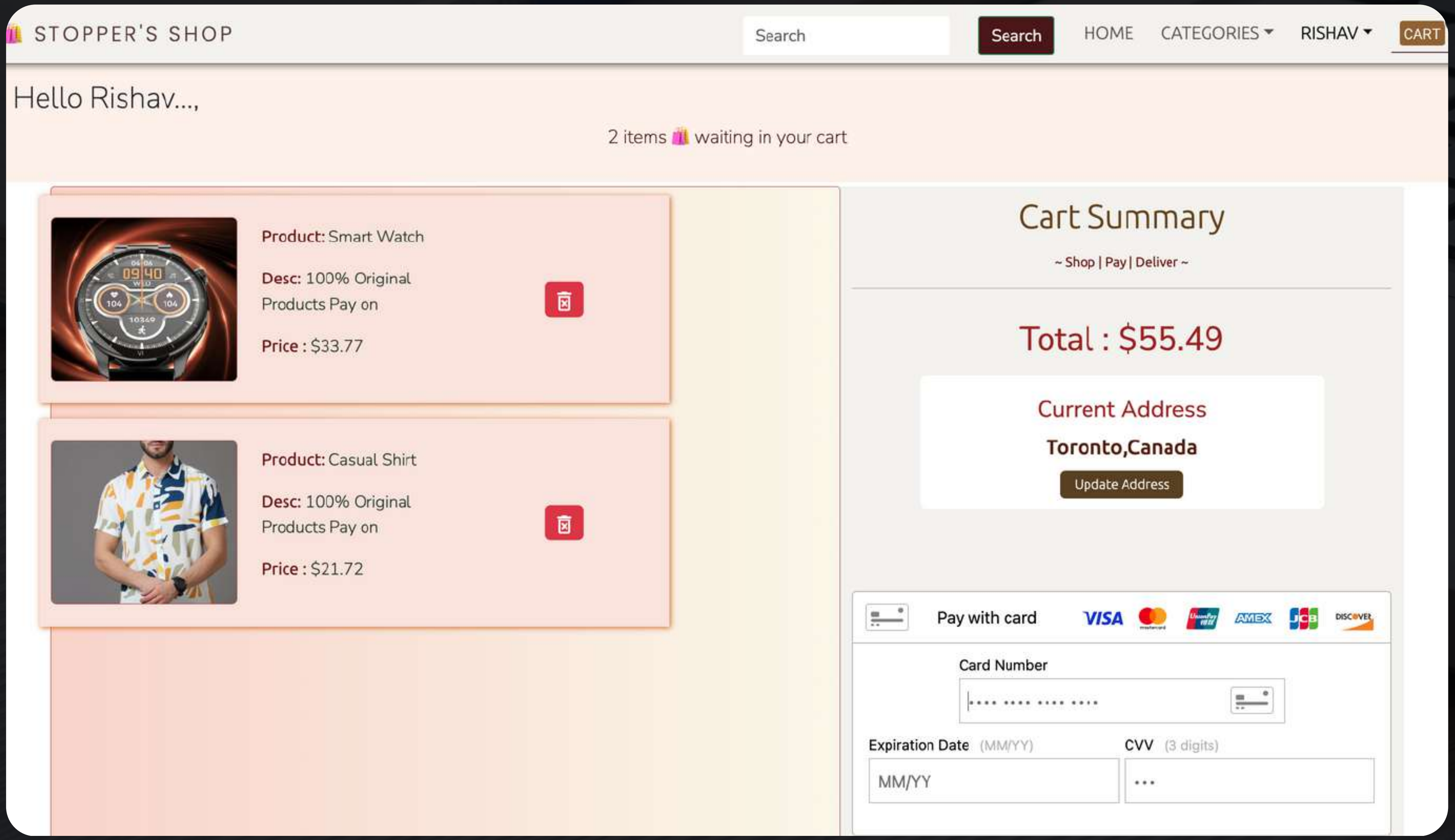
Women ~ only

10 results found

Suits $12.60
100% Original Products Pay on delivery might be available...
More Details

Top $50.55
100% Original Products Pay on delivery might be available...
More Details

saree $8.99
Ethnic Motif Woven Design Zari Pure Silk Kanjeevaram Saree...
More Details



Hello Rishav...,

2 items 🛍 waiting in your cart

Product: Smart Watch
Desc: 100% Original Products Pay on
Price : $33.77

Product: Casual Shirt
Desc: 100% Original Products Pay on
Price : $21.72

Cart Summary

~ Shop | Pay | Deliver ~

Total : $55.49

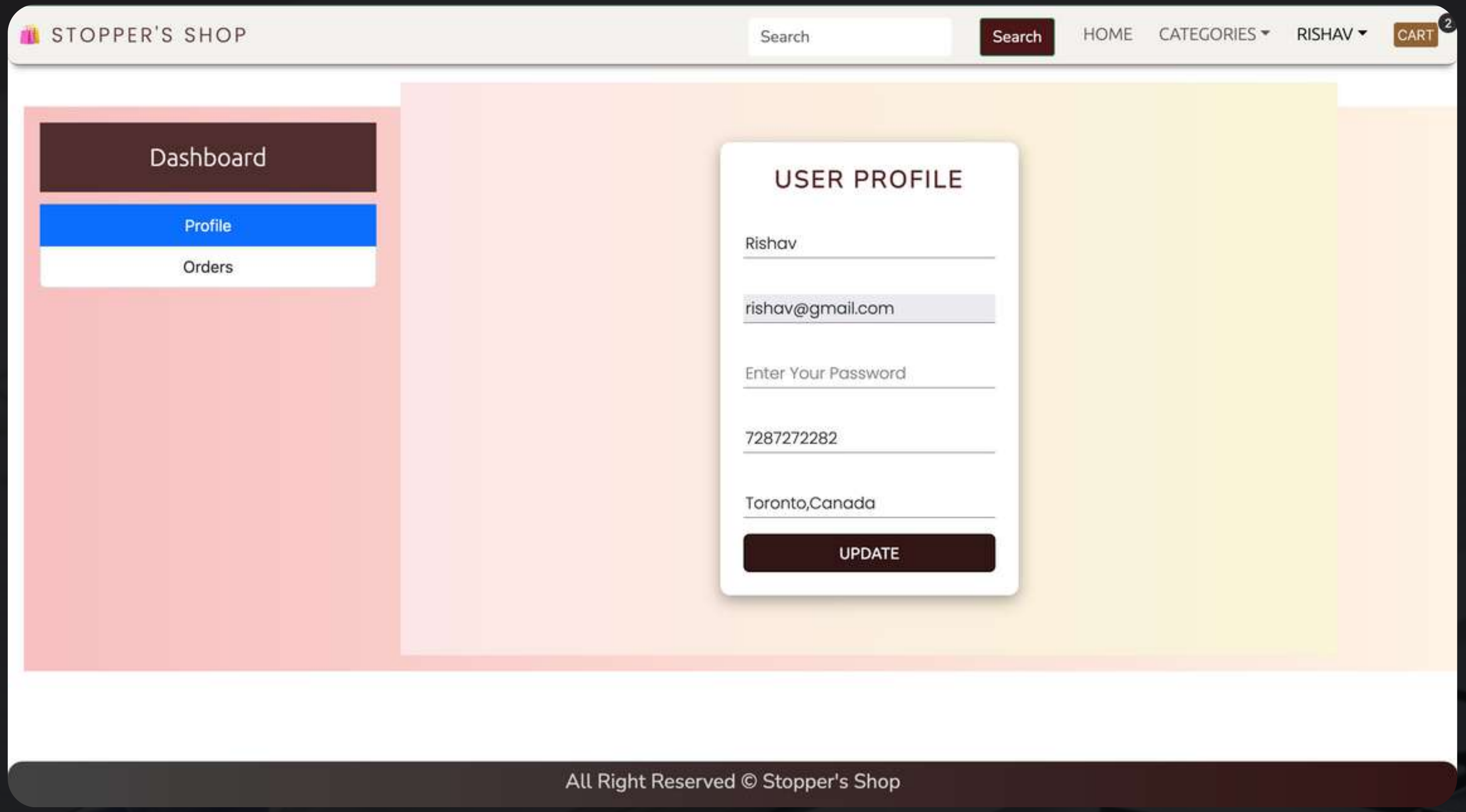
Current Address
Toronto,Canada
Update Address

Pay with card
Card Number
Expiration Date (MM/YY)
MM/YY
CVV (3 digits)

- WHEN THE USER IS DONE WITH THE SHOPPING ,HE/SHE CAN CLICK ON THE CART OPTION IN THE MENUBAR.
- THE PRODUCTS SELECTED BY THE USER GETS ADDED TO CART AND THE TOTAL AMOUNT AND THE DEFAULT ADDRESS IS VISIBLE .
- AFTER THE PAYMENT IS DONE ,USER CAN SEE THEIR ORDERS IN THE DASHBOARD.

STOPPERS’S SHOP

MORE PAGES ~ USER



USER PROFILE

Rishav

rishav@gmail.com

Enter Your Password

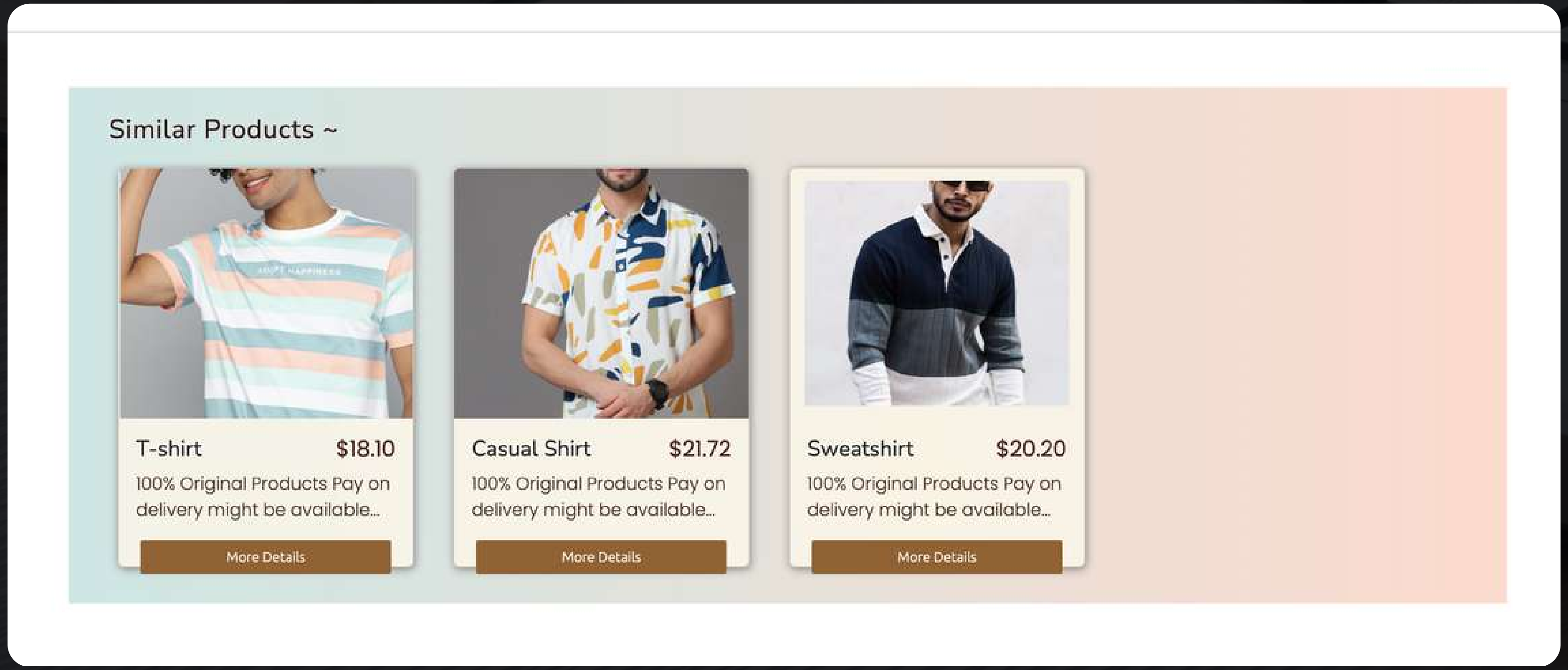
7287272282

Toronto,Canada

UPDATE

All Right Reserved © Stopper's Shop

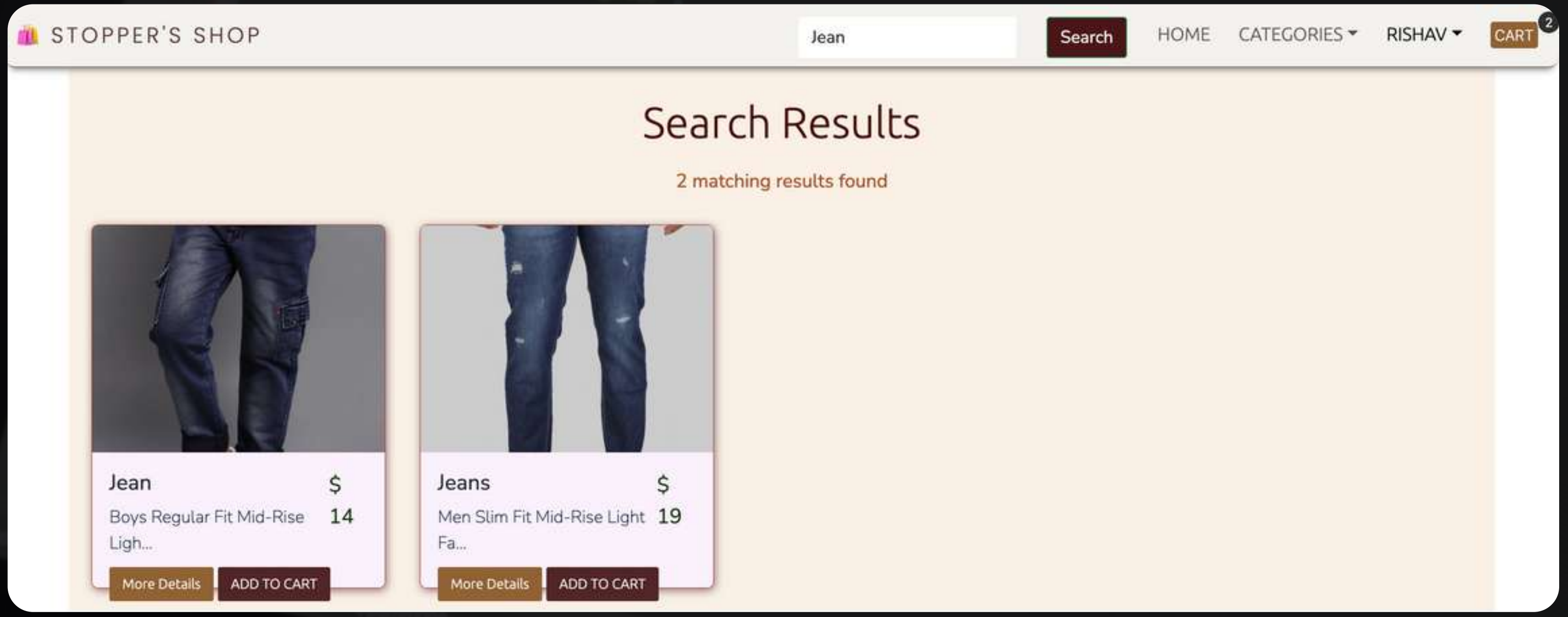
Profile Page


Similar Products ~

T-shirt $18.10
100% Original Products Pay on delivery might be available...

Casual Shirt $21.72
100% Original Products Pay on delivery might be available...

Sweatshirt $20.20
100% Original Products Pay on delivery might be available...



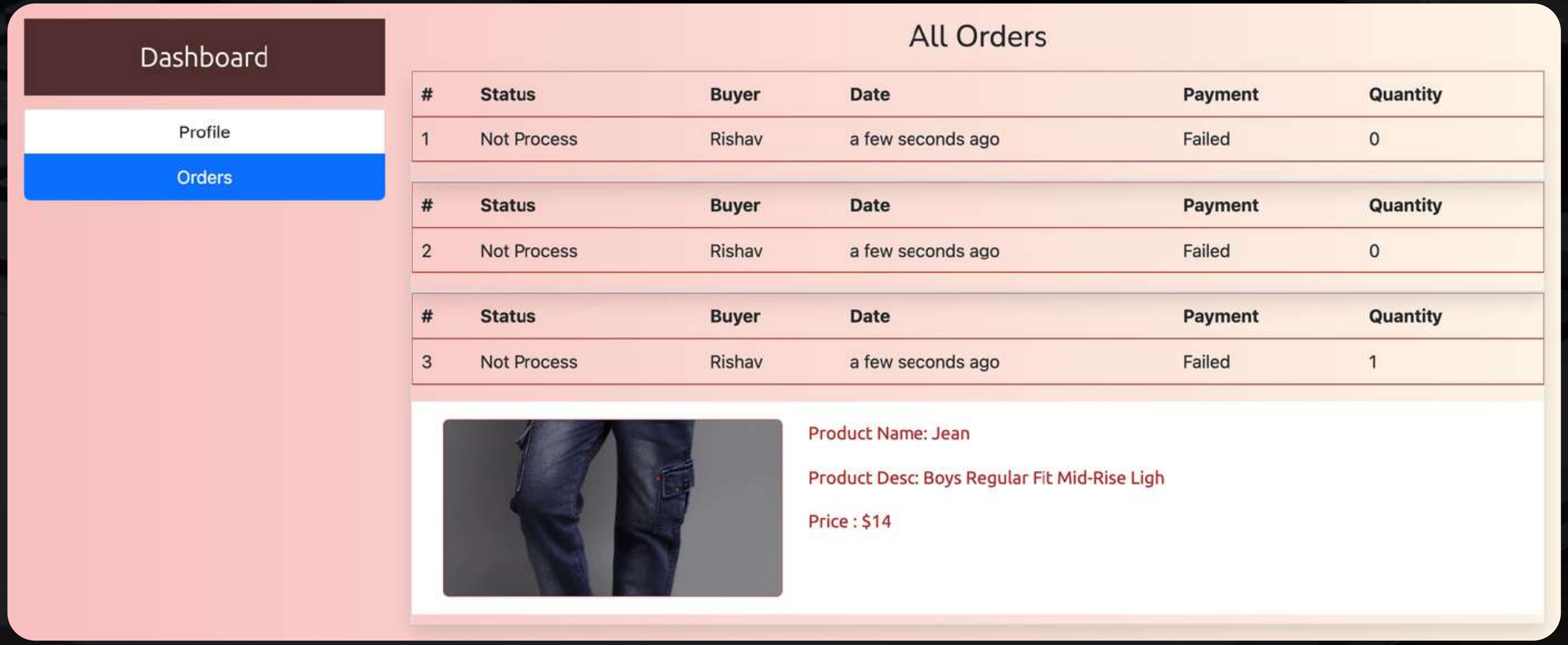
Search Results

2 matching results found

Jean $ 14
Boys Regular Fit Mid-Rise Ligh...

Jeans $ 19
Men Slim Fit Mid-Rise Light Fa...

Search Filter Page



All Orders

#	Status	Buyer	Date	Payment	Quantity
1	Not Process	Rishav	a few seconds ago	Failed	0

#	Status	Buyer	Date	Payment	Quantity
2	Not Process	Rishav	a few seconds ago	Failed	0

#	Status	Buyer	Date	Payment	Quantity
3	Not Process	Rishav	a few seconds ago	Failed	1

Product Name: Jean

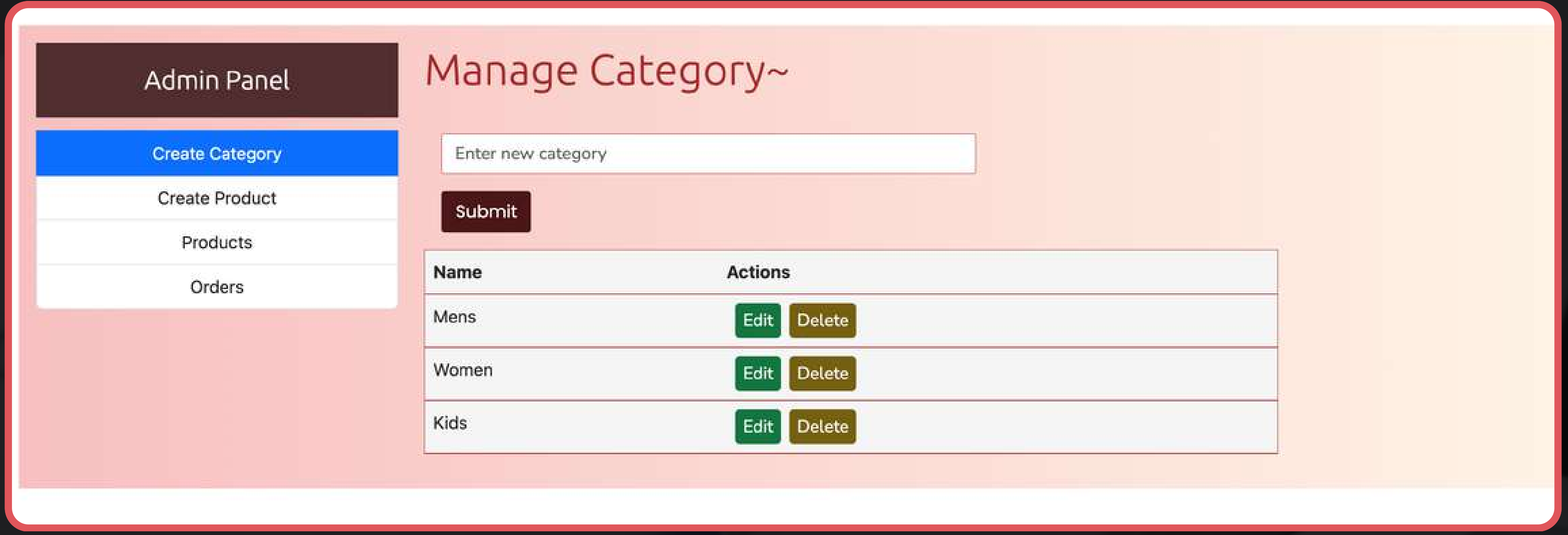
Product Desc: Boys Regular Fit Mid-Rise Ligh

Price : $14

Orders Page

STOPPERS'S SHOP

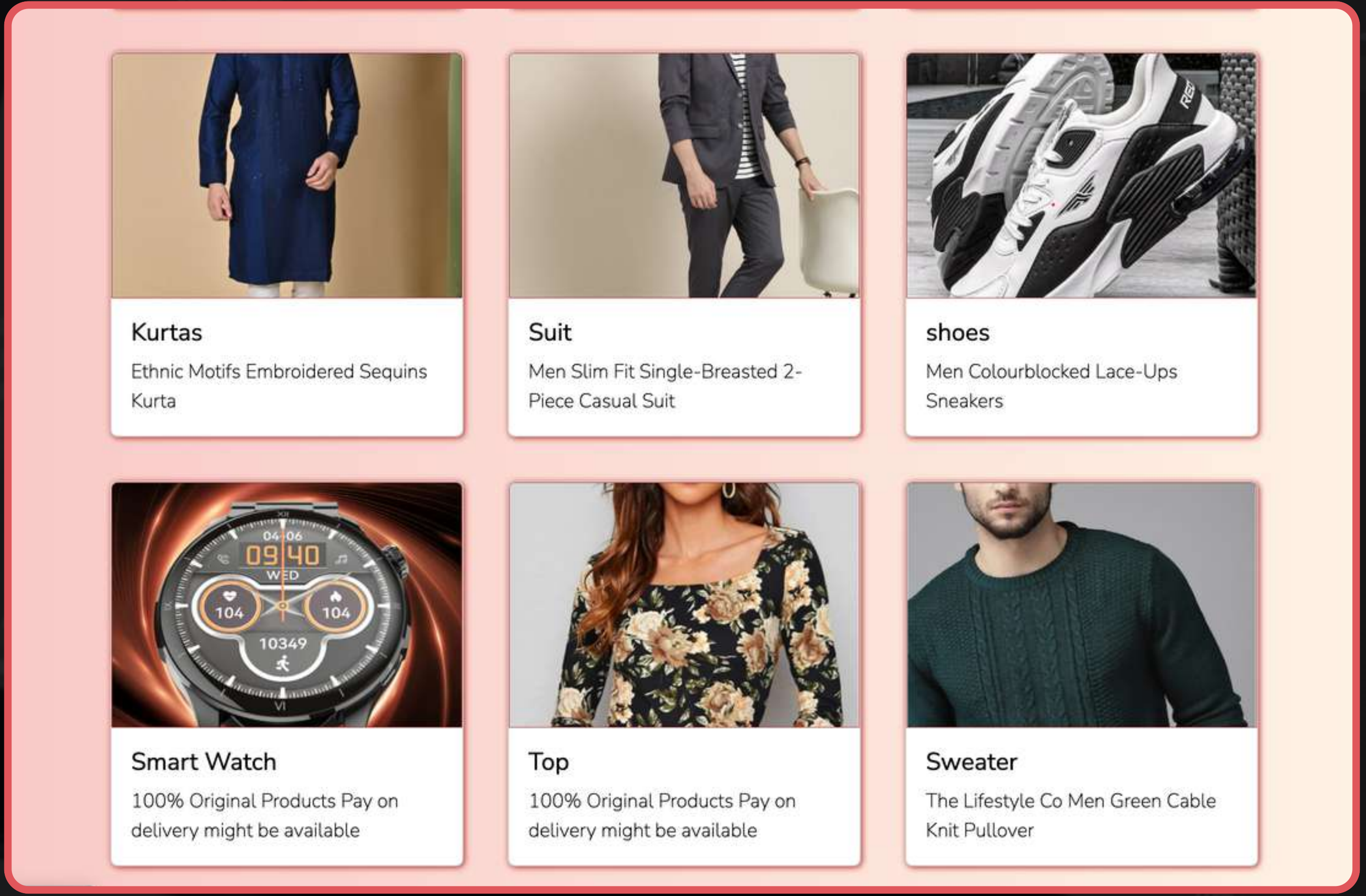
ADMIN ~ PAGES



Manage Category~

Name	Actions
Mens	Edit Delete
Women	Edit Delete
Kids	Edit Delete

CATEGORY CREATION



Kurtas
Ethnic Motifs Embroidered Sequins Kurta

Suit
Men Slim Fit Single-Breasted 2-Piece Casual Suit

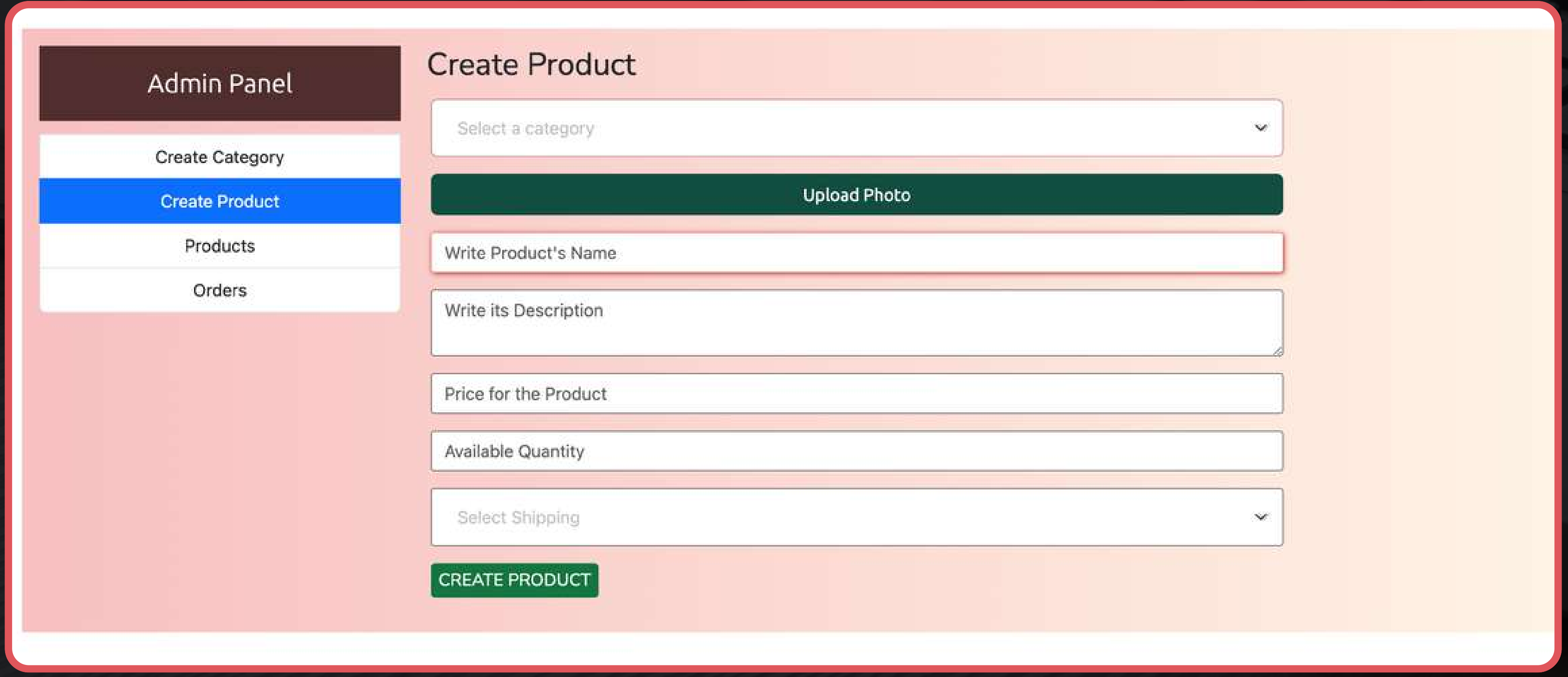
shoes
Men Colourblocked Lace-Ups Sneakers

Smart Watch
100% Original Products Pay on delivery might be available

Top
100% Original Products Pay on delivery might be available

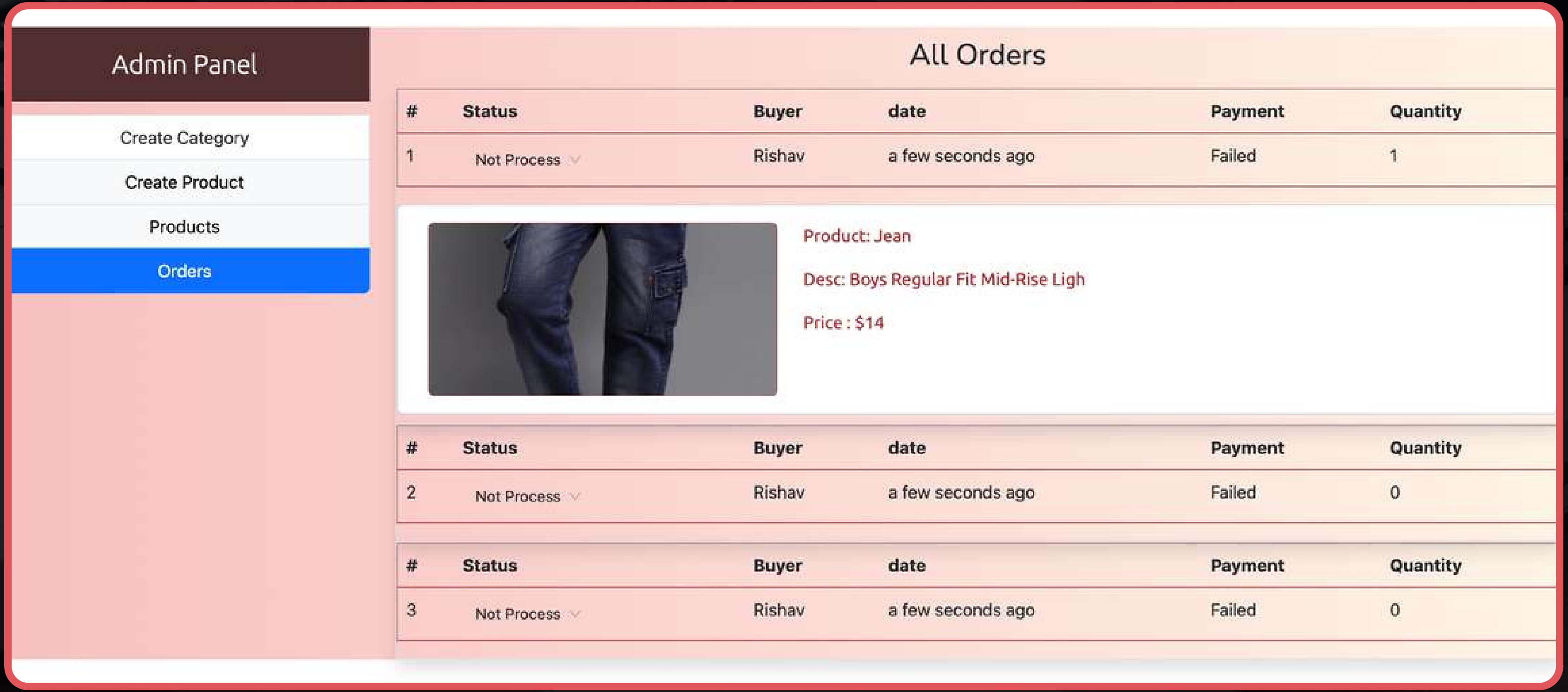
Sweater
The Lifestyle Co Men Green Cable Knit Pullover

ALL PRODUCTS CREATED



Create Product

PRODUCT CREATION



All Orders

#	Status	Buyer	date	Payment	Quantity
1	Not Process	Rishav	a few seconds ago	Failed	1

Product: Jean

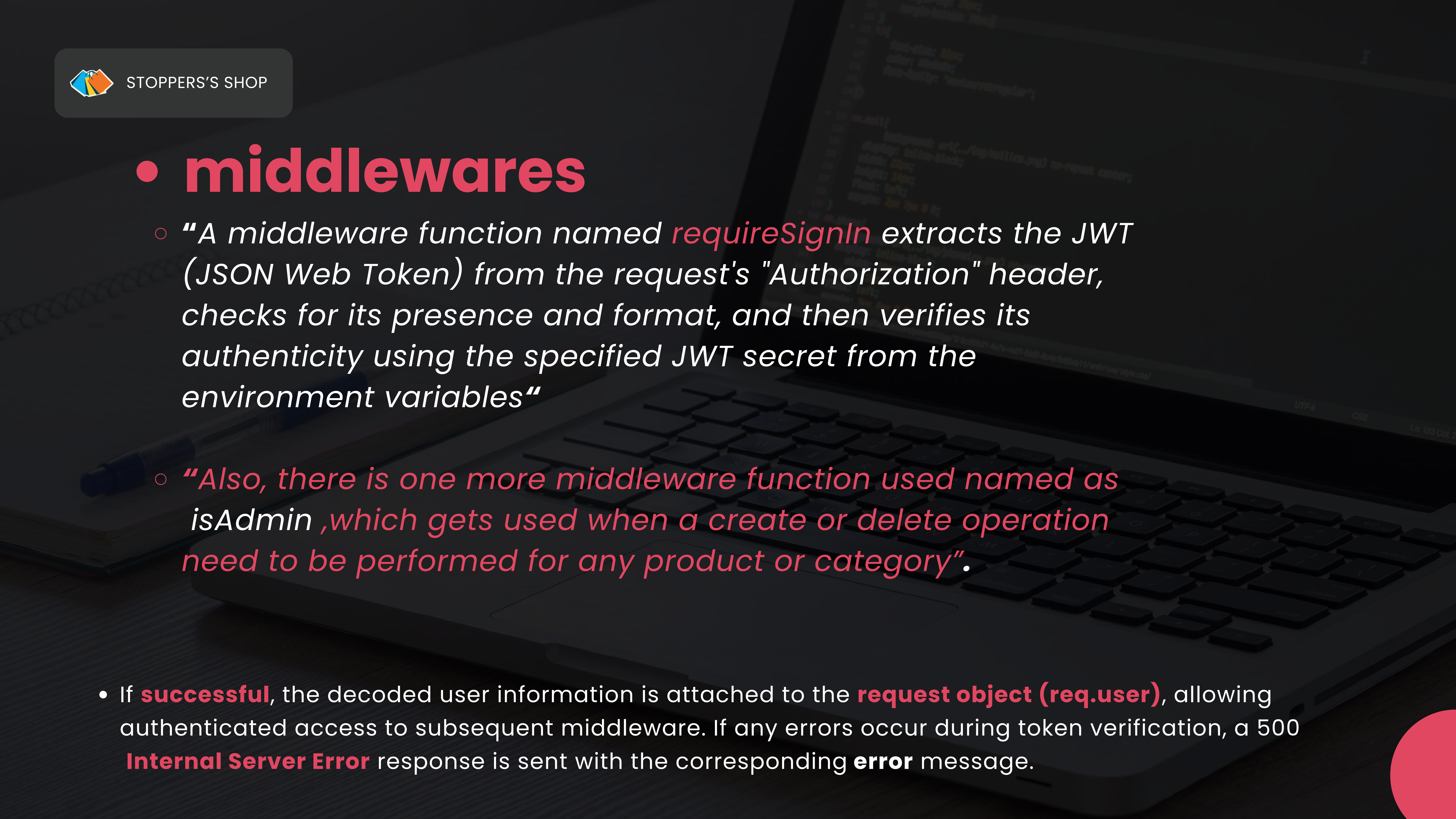
Desc: Boys Regular Fit Mid-Rise Ligh

Price : $14

#	Status	Buyer	date	Payment	Quantity
2	Not Process	Rishav	a few seconds ago	Failed	0

#	Status	Buyer	date	Payment	Quantity
3	Not Process	Rishav	a few seconds ago	Failed	0

PRODUCTS PURCHASED



STOPPERS'S SHOP

• middlewares

- *"A middleware function named requireSignIn extracts the JWT (JSON Web Token) from the request's "Authorization" header, checks for its presence and format, and then verifies its authenticity using the specified JWT secret from the environment variables"*

- *"Also, there is one more middleware function used named as* isAdmin *,which gets used when a create or delete operation need to be performed for any product or category".*

- If **successful**, the decoded user information is attached to the **request object (req.user)**, allowing authenticated access to subsequent middleware. If any errors occur during token verification, a 500 **Internal Server Error** response is sent with the corresponding **error** message.

controllers

- ***auth controller ~*** *performs operations like registering a user, saving the user etc. including functions for login , forgot password, profile update and orders details.*
- ***category controller ~*** *particularly created for handling the CRUD operations on category collections of the DB.Used by the admin to create,update and delete a category.(Also, used for getting all the categories available)*
- ***product controller ~*** *contains functions for getting the products, deleting, updating and creating products.Also, contains functions for the search & filter functionality along with the payment handlers.*

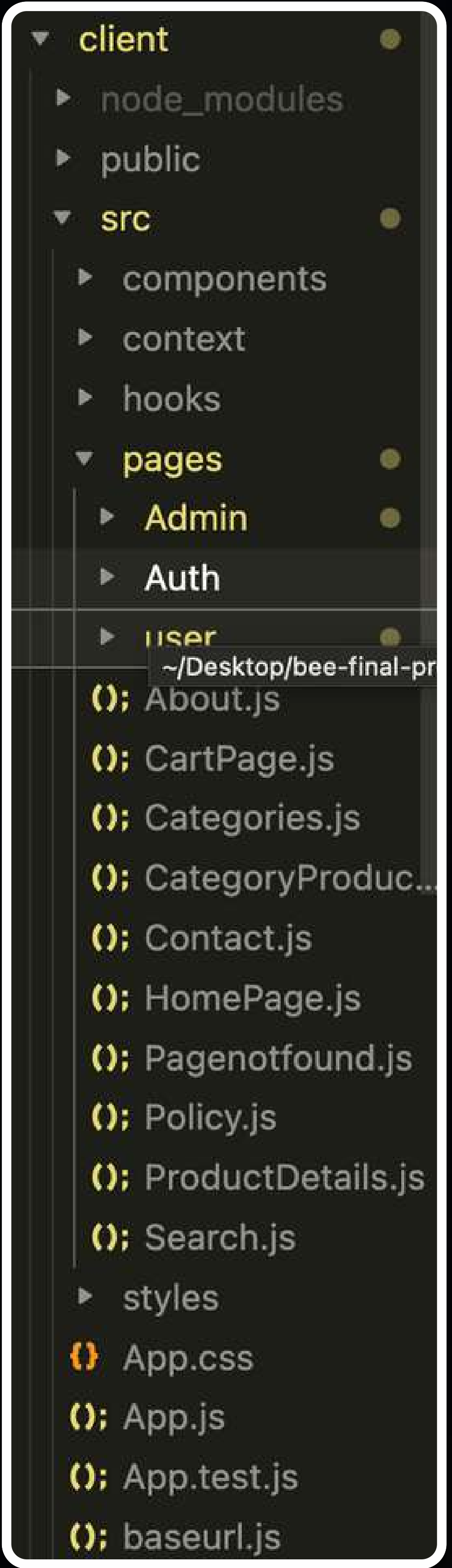
- ## routes
  - ***authRoutes ~*** *Express router defines a POST login route, utilizing the login controller function to handle user authentication requests along with profile ,orders and protected-admin routes.*

  - ***categoryRoutes ~*** *routes the requests related to categories after identification to the category controller.*

  - ***productRoutes ~*** *this route handles all the requests related to products like search ,filter etc. and forwards them to the product controllers.*

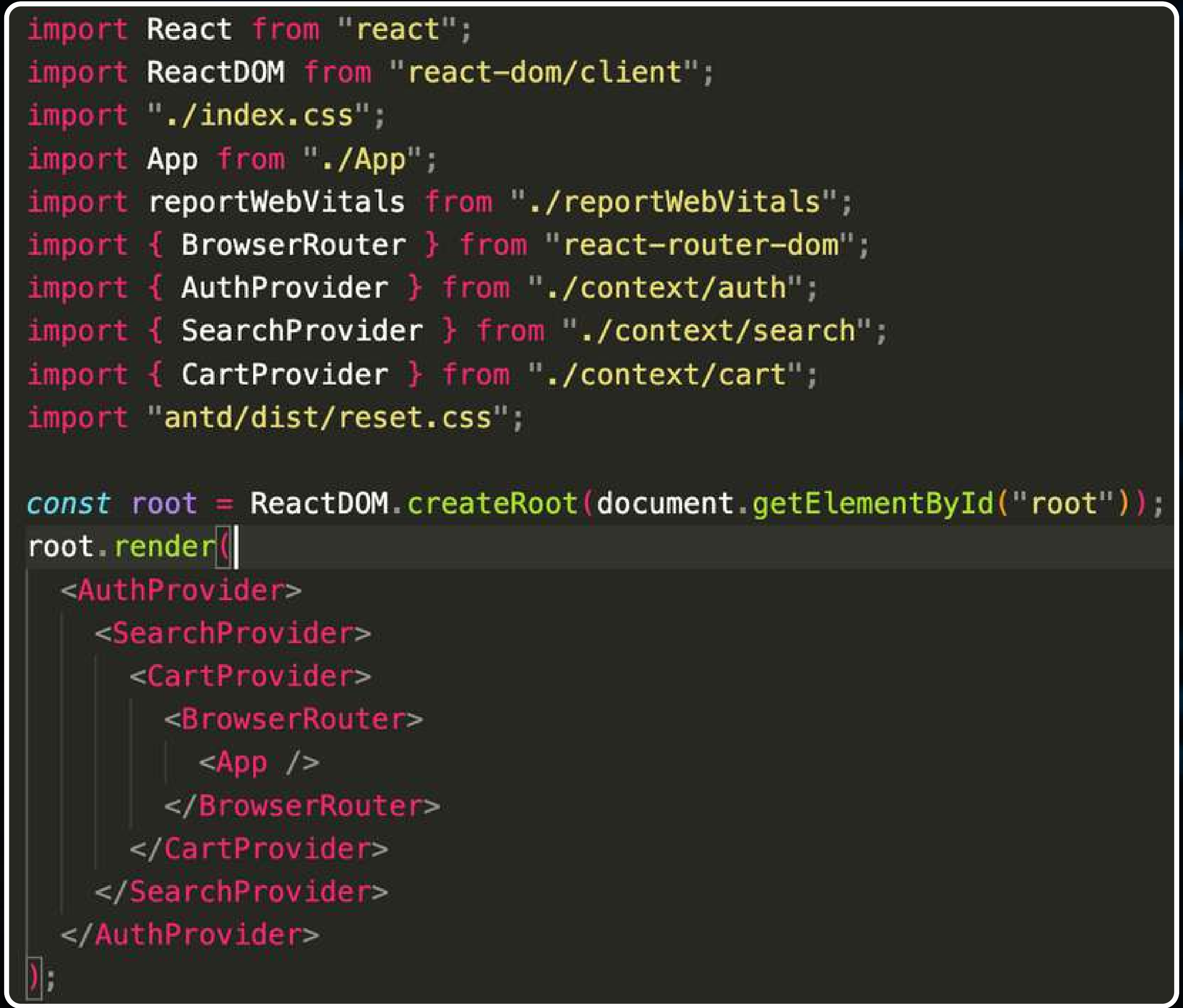
STOPPERS'S SHOP

CODE OVERVIEW

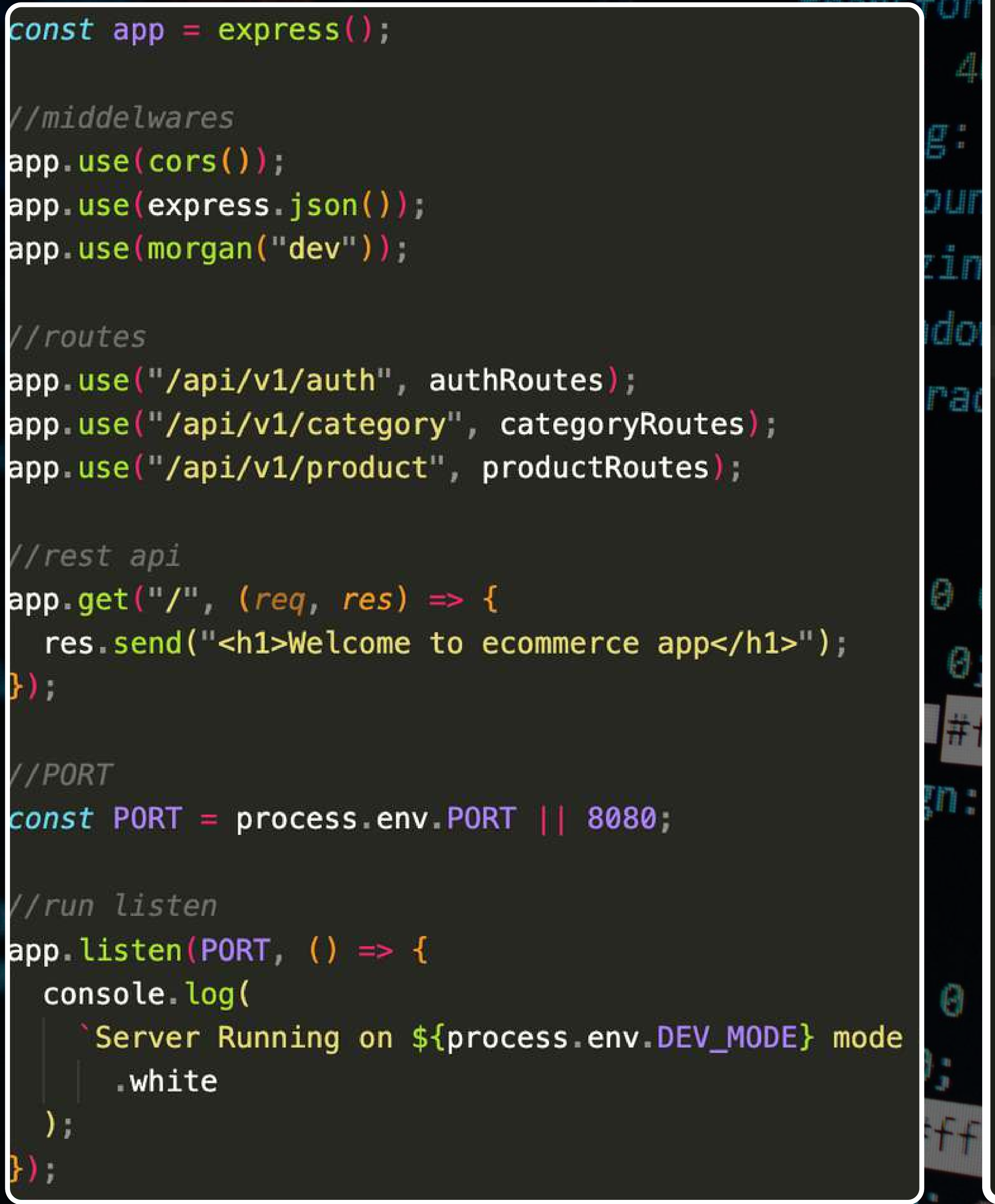


```
client
  node_modules
  public
  src
    components
    context
    hooks
    pages
      Admin
      Auth
      user
      About.js
      CartPage.js
      Categories.js
      CategoryProduc...
      Contact.js
      HomePage.js
      Pagenotfound.js
      Policy.js
      ProductDetails.js
      Search.js
    styles
    App.css
    App.js
    App.test.js
    baseurl.js
```

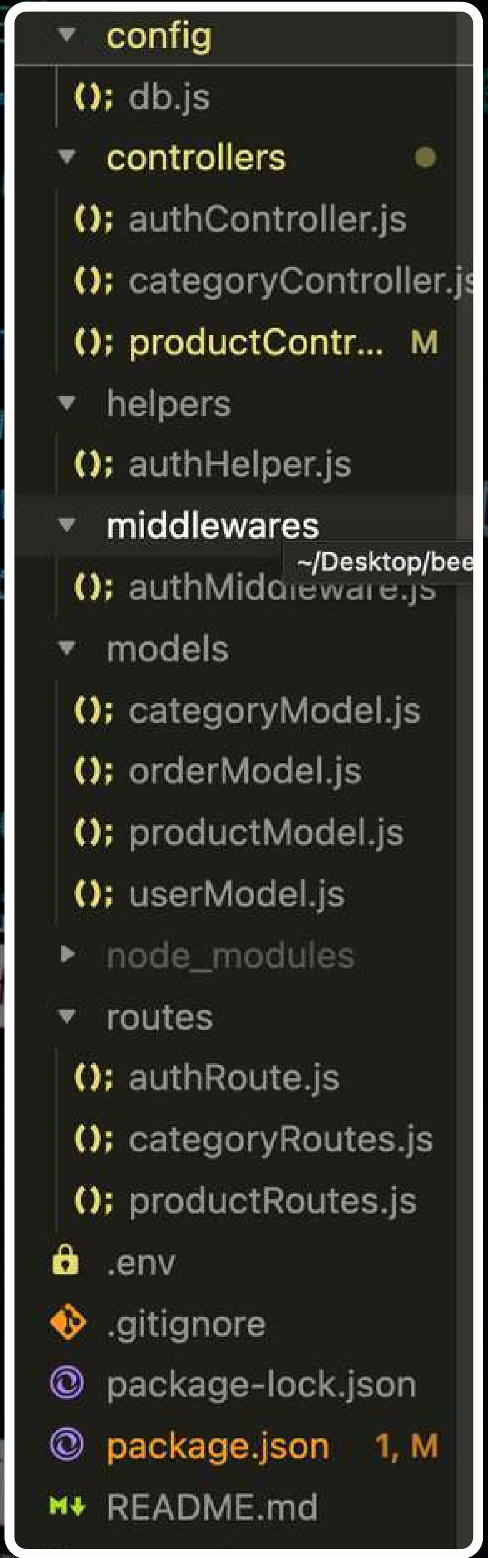
~/Desktop/bee-final-pr



```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import { BrowserRouter } from "react-router-dom";
import { AuthProvider } from "./context/auth";
import { SearchProvider } from "./context/search";
import { CartProvider } from "./context/cart";
import "antd/dist/reset.css";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <AuthProvider>
    <SearchProvider>
      <CartProvider>
        <BrowserRouter>
          <App />
        </BrowserRouter>
      </CartProvider>
    </SearchProvider>
  </AuthProvider>
);
```



```
const app = express();

//middelwares
app.use(cors());
app.use(express.json());
app.use(morgan("dev"));

//routes
app.use("/api/v1/auth", authRoutes);
app.use("/api/v1/category", categoryRoutes);
app.use("/api/v1/product", productRoutes);

//rest api
app.get("/", (req, res) => {
  res.send("<h1>Welcome to ecommerce app</h1>");
});

//PORT
const PORT = process.env.PORT || 8080;

//run listen
app.listen(PORT, () => {
  console.log(
    `Server Running on ${process.env.DEV_MODE} mode
      .white
  );
});
```



```
config
  db.js
controllers
  authController.js
  categoryController.js
  productContr...  M
helpers
  authHelper.js
middlewares
  authMiddleware.js
models
  categoryModel.js
  orderModel.js
  productModel.js
  userModel.js
node_modules
routes
  authRoute.js
  categoryRoutes.js
  productRoutes.js
.env
.gitignore
package-lock.json
package.json  1, M
README.md
```

~/Desktop/bee

RESOURCES & PLATFORMS USED

Resources:

- MDN Docs
- Stack Overflow
- Material UI Docs
- React Documentations

Platforms:

- Google Meet (meetings)
- VS Code (Editors)
- GitHub(Collaboration)
- Google Drive

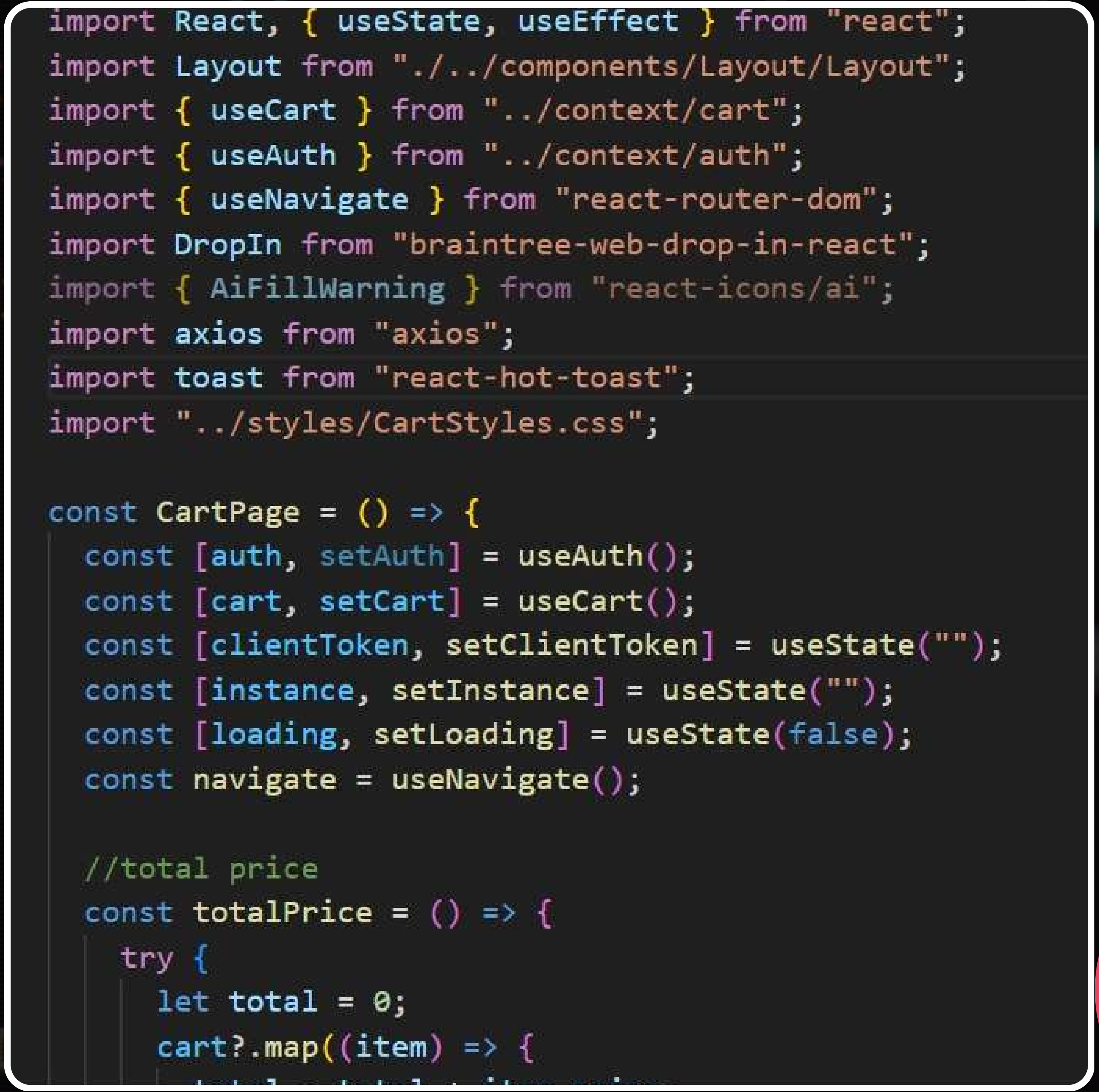
ROLES AND RESPONSIBILITIES ~ RIYA

- ***setting up the backend directory with initially required dependencies for the project.***

- ***complete admin panel pages along with their routes and controller configurations.***

- ***individual category pages along with routes and controllers .***

- ***cart page along with its complete functionality and payment gateway.***



```
import React, { useState, useEffect } from "react";
import Layout from "./../components/Layout/Layout";
import { useCart } from "../context/cart";
import { useAuth } from "../context/auth";
import { useNavigate } from "react-router-dom";
import DropIn from "braintree-web-drop-in-react";
import { AiFillWarning } from "react-icons/ai";
import axios from "axios";
import toast from "react-hot-toast";
import "../styles/CartStyles.css";

const CartPage = () => {
  const [auth, setAuth] = useAuth();
  const [cart, setCart] = useCart();
  const [clientToken, setClientToken] = useState("");
  const [instance, setInstance] = useState("");
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();

  //total price
  const totalPrice = () => {
    try {
      let total = 0;
      cart?.map((item) => {
```

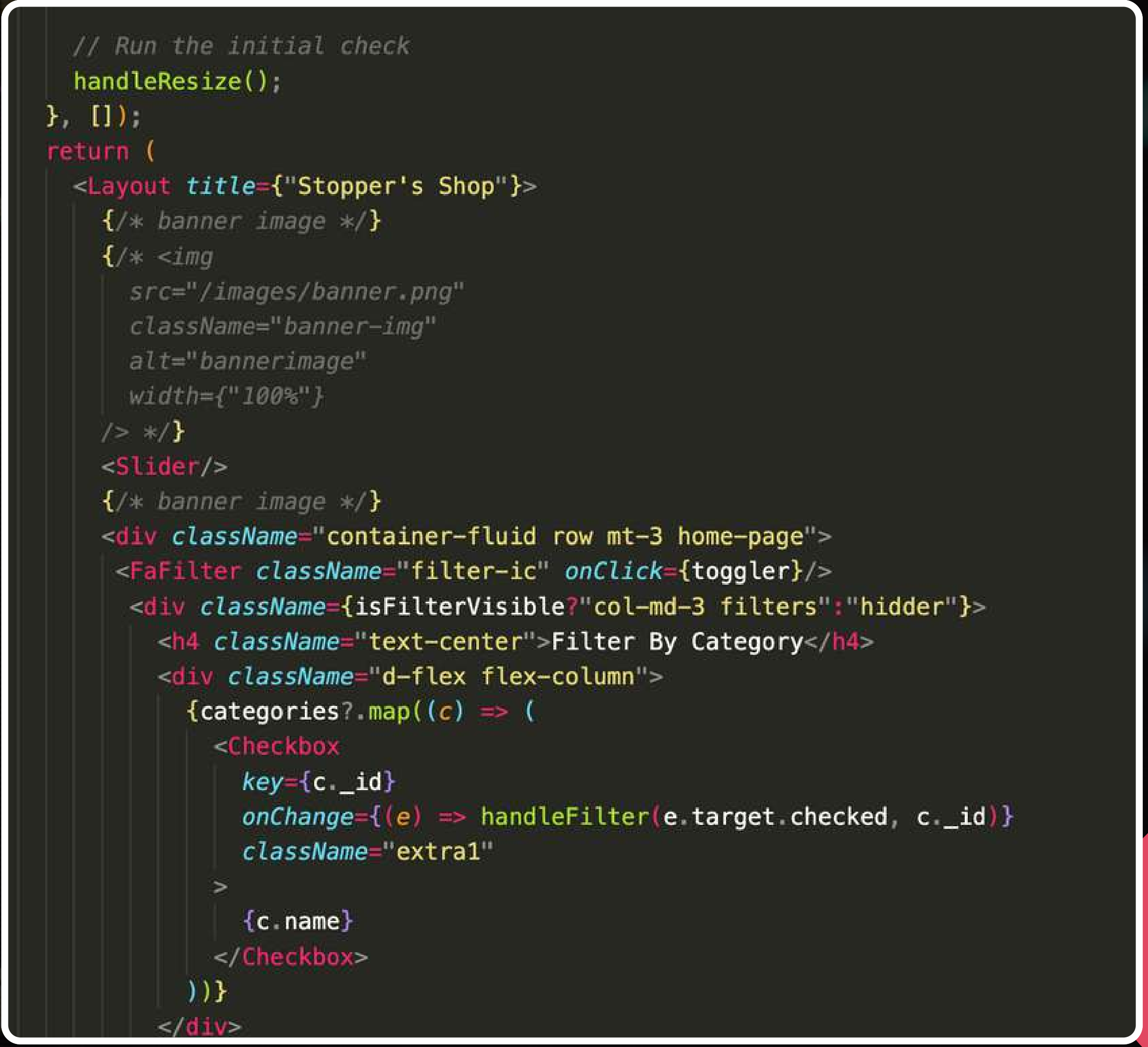
ROLES AND RESPONSIBILITIES ~ RISHAV

- ***setting up the github environment for the overall collaboration of the project .***

- ***authentication pages along with their respective routes and controllers.***

- ***homepage of stopper’s shop with the filter functionality and the product lists .***

- ***all-categories page showing multiple categories and the the categories listed by the admin .***



```
  // Run the initial check
  handleResize();
}, []);
return (
  <Layout title={"Stopper's Shop"}>
    {/* banner image */}
    {/* <img
      src="/images/banner.png"
      className="banner-img"
      alt="bannerimage"
      width={"100%"}
    /> */}
    <Slider/>
    {/* banner image */}
    <div className="container-fluid row mt-3 home-page">
     <FaFilter className="filter-ic" onClick={toggler}/>
      <div className={isFilterVisible?"col-md-3 filters":"hidder"}>
        <h4 className="text-center">Filter By Category</h4>
        <div className="d-flex flex-column">
          {categories?.map((c) => (
            <Checkbox
              key={c._id}
              onChange={(e) => handleFilter(e.target.checked, c._id)}
              className="extra1"
            >
              {c.name}
            </Checkbox>
          ))}
        </div>
```

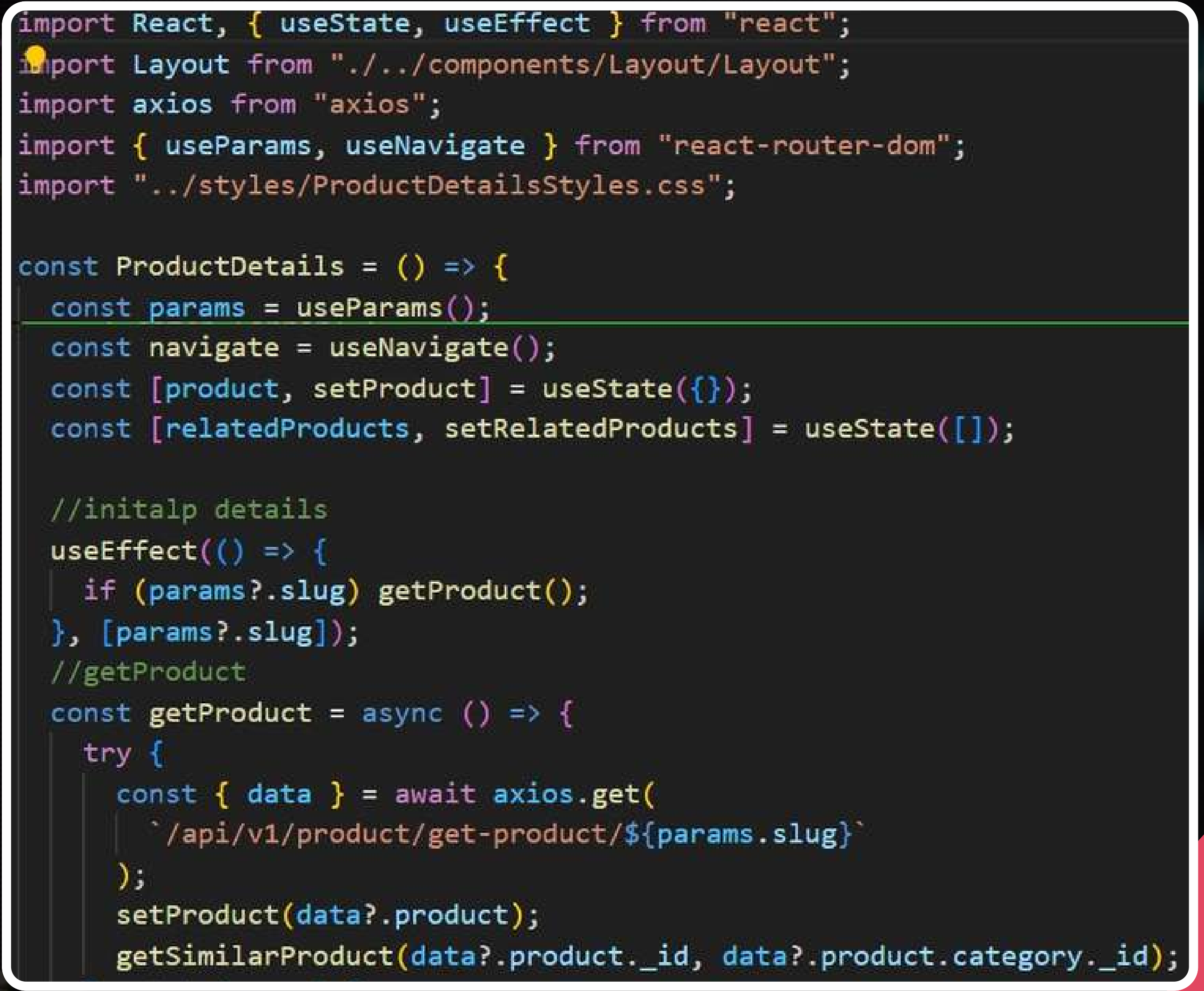
ROLES AND RESPONSIBILITIES ~ ANKIT

- *setting up the frontend directory with initially required dependencies for the project.*

- *user panel pages like profile and orders section along with their routes and controller functions.*

- *individual product details page that shows the details for each of the product when clicked.*

- *implementing the search functionality with a separate page including the filtered items according to searched keywords.*



```
import React, { useState, useEffect } from "react";
import Layout from "./../components/Layout/Layout";
import axios from "axios";
import { useParams, useNavigate } from "react-router-dom";
import "../styles/ProductDetailsStyles.css";

const ProductDetails = () => {
  const params = useParams();
  const navigate = useNavigate();
  const [product, setProduct] = useState({});
  const [relatedProducts, setRelatedProducts] = useState([]);

  //initalp details
  useEffect(() => {
    if (params?.slug) getProduct();
  }, [params?.slug]);
  //getProduct
  const getProduct = async () => {
    try {
      const { data } = await axios.get(
        `/api/v1/product/get-product/${params.slug}`
      );
      setProduct(data?.product);
      getSimilarProduct(data?.product._id, data?.product.category._id);
```

STOPPERS'S SHOP

THANK YOU

STOPPERSHOP.VERCEL.APP