

Module 6 (Ecommerce website).

Flow of creating the application - First we will build the Backend and then once we have all of the APIs ready we will create the front end.

Deploy Url => <https://frail-outerwear-cod.cyclic.app/>

Following are the Features included in the App.

1. Mongodb setup.
2. Models & hashing password.
3. Register ,login and forgot password.
4. Using React Helmet for SEO.
5. Admin And User Dashboard.
6. Category API.
7. Product API.
8. Create,update and delete Product.
9. Create ,update and delete Category.
10. Filter by category and Price.
11. Search filter single product page
12. Cart Page
13. User Profile
14. Payment Gateway.
15. User Orders.
16. Admin Orders

started with Part 1 - Backend

Part 1 – Backend

Tech Stack Used

- Node.js (main backend language)
- Express.js (framework based on nodejs to create REST APIs)
- Mongoose (to connect MongoDB with express.js)
- bcrypt (to encrypt password when stored in DB)
- Formidable (to upload images)

Statement

Create a backend app which will contain a set of REST APIs to store and get data from MongoDB which will help frontend interact and store user information. The app will have three entities and we will build REST API's around all three entities to perform all the required functionalities.

1. AuthRoute
2. CategoryRoute
3. ProductRoute

Details on Functionalities to build

1. Setup project and create a server on PORT

- Start by creating a Nodejs project and installing ExpressJs init.
- create an server.js file and listen to the express server on any port.

2. Connect app with MongoDB

- install mongoose and connect the app with either local MongoDB server or theonline MongoDB atlas .

console.log("connerction to mongoDb(conn.connection.host)"), when MongoDB is connected to indicate and confirmthat DB is connected successfully.

3. Creating User schema

- Before implementing user related APIs let's first create a User MongoDB schema to define how the data will look like.
- Below are the fields included in the User schema.

Field name	Data type	Additional attributes	Comments
Name	String	This field is required trim true	This field will store a user fullname Eg. John Doe
Email	String	This field is required and also the username should beunique.	This field will store a users emailid. Eg. john@email.com

Password	String	This field is required	<p>This field will store the user account password.</p> <p>Also the password stored in the DB should be encrypted.</p> <p>eg. actual password - john Encrypted</p> <p>password stored in DB - <code>\$2a\$10\$HaK9HroNq1429ckGePQFNeL9O1fYU7 Eb MFVDKHZC9/xT.7f4mjApm</code></p>
phone	String	This field is required	<p>This field will store the phone number of the user it must be required. phone no must be greater than 10.</p> <p>Eg. 7897744166</p>
Answer	String	This field is required	<p>This field will store user hobby when user forgot their password and renew password it is required.</p> <p>Eg. game, pet name, nick name etc</p>
Role	String	This field is required	<p>This field will store 0 and 1 value 0 for user and 1 for Admin</p>

Also use the { timestamp: true } feature of mongoose which will store createdAt and updatedAt dates of every document.

Eg of how a typical User document will look like

Admin user

```
_id: ObjectId('63f80920d108b94f5c2ea43a')
name: "admin"
email: "admin@admin.com"
password: "$2b$10$XyI5EZ8n/cfaWGW3Je3MROz9DK7.AJ5VHpp4IcI2hbcdb.DBtbr.
6"
phone: "7897744166"
address: "mevdan"
answer: "football"
role: 1
createdAt: 2023-02-24T00:47:28.922+00:00
updatedAt: 2023-02-24T00:47:28.922+00:00
__v: 0
```

User Register

```
_id: ObjectId('63f80920d108b94f5c2ea44r')
name: "user"
email: "user@gmail.com"
password: "$2b$10$XyI5EZ8n/cfaWGW3Je3MROz9DK7.AJ5VHpp4IcI2hbcdb.DBtbr.
6"
phone: "7897744166"
address: "mevdan"
answer: "football"
role: 0
createdAt: 2023-02-24T00:47:28.922+00:00
updatedAt: 2023-02-24T00:47:28.922+00:00
__v: 0
```

{ Creating Category schema.

Field name	Data type	Additional attributes	Comments
name	String	This field is required name should be unique.	This field will store a user fullname Eg. John Doe
slug	String	Lowercase : true	This field is use for including hyphen for between two words Eg. men collection -> men-collection

Creating Product Schema

- Product MongoDB schema to define how the data will look like.
- Below are the fields included in the Product schema.

Field name	Data type	Additional attributes	Comments
name	String	This field is required	<p>This field will content name of the product</p> <p>Eg. Shirt , Paint, Nakab , skirt .jeans?</p>
slug	string	This field is required	<p>This field will use for hyphen between two words for easily get single product with category wise also update product slug is use when get browser photo using formidabal npm package also use this</p> <p>Eg.</p> <p>Printed Suit -> Printed-Suit</p>
description	String	This field is required	This field is use for details description of the product
price	Number	This field is required	This field is use for price of the product
category	type :mongoose. ObjectId ref:category	This field is required	This field is required for product category ref to category Schema for name and slugify the category
qunatity	Number	This field is required	
photo	Data :buffer	Content type:String	This field is required for include a image their product

Shipping	Boolean		
----------	---------	--	--

use the { timestamp: true } feature of mongoose which will store createdAt and updatedAt dates of every document.

Creating Order Schema

- Order MongoDB schema to define how the mange or order will look like .
- Below are the fields included in the order schema.

Field name	Data type	Additional attributes	Comments
product	Mongoose. objectId Ref:product Model	This field is required	This field will use for fetching all product which user order by using cart
payment	Empty object {}		
buyer	Mongoose. ObjectId Ref:users		
status	string	Default: "not processing"	<code>enum: ["Not Process", "processing", "shipped", "deliver", "cancel"]</code>

Eg of how a typical Tweet document will look like

```
_id
:
ObjectId('63ffd5aa160b89fada9c8ff8')
products
:
Array
0
:
ObjectId('63fd442b56b2db50c432a1e7')
payment
:
Object
transaction
:
Object
success
:
true
buyer
:
ObjectId('63ff806ef3e06fb0c4c6fa48')
status
:
"processing"
createdAt
:
2023-03-01T22:46:02.477+00:00
updatedAt
:
2023-03-02T06:34:15.748+00:00
__v
:
0
```

Server.js

```
const express = require("express")
const colors = require("colors")
const dotenv = require('dotenv')
const morgan = require('morgan')
const cors = require('cors')
const connectDB = require("./config/db")
const authRoutes = require("./routes/authRoutes.js")
const categoryRoutes = require("./routes/categoryRoutes.js")
const productRoutes = require("./routes/productRoutes")

//configure env
dotenv.config()

//database config
connectDB()

//test object
const app = express()

//middlewares
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("welcome")
})

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

```
app.listen(PORT, ()=>{
  console.log(`server running on ${process.env.DEV_MODE} mode on port ${PORT}`.bgCyan.white)
})
```

MongoDb Connection(config/db.js)

```
const mongoose = require("mongoose")
const colors = require("colors")
const connectDB = async()=>{
  try {
    const conn = await mongoose.connect("mongodb://localhost:27017/ecommerce")
    console.log(`connected to mongodb Database ${conn.connection.host}`.bgWhite.white)
  } catch (error) {
    console.log(`Error in Mongodb ${error}`.bgRed.white)
  }
}
module.exports = connectDB;
```

UserSchema(model/userModel.js)

```
const mongoose = require("mongoose")

const userSchema = new mongoose.Schema({
  name: {
    type:String,
    required:true,
    trim:true
  },
  email: {
    type:String,
    required:true,
    unique:true
  },
  password: {
    type:String,
    required:true,
  },
  phone: {
    type:String,
    required:true
  },
  address: {
    type:String,
    required:true
  },
  answer:{
    type:String,
    required:true
  },
  role:{
    type:Number,
    default:0
  }
}, {timestamps:true})

module.exports= mongoose.model("users" , userSchema)
```

Auth Routes

```
const express = require("express")
const {registerController,
      loginController,
      testController,
      forgotPasswordController,
      updateProfileController,
      getOrdersController,
      getAllOrdersController,
      orderStatusController,
      } =require("../controllers/authController")
const { requireSignin, isAdmin } = require("../middleware/authMiddleware")
```

```

//router Object
const router = express.Router()

//routing
//REGISTER || METHOD POST
router.post("/register", registerController)

//LOGIN || METHOD POST
router.post("/login", loginController)

//FORGOT PASSWORD
router.post('/forgot-password', forgotPasswordController)

//getall users route
router.get('/get-all-users', requireSignin, isAdmin, testController)

//protected user route auth
router.get('/user-auth', requireSignin, (req, res) => {
    res.status(200).send({ok: true})
})

//protected admin route auth
router.get('/admin-auth', requireSignin, isAdmin, (req, res) => {
    res.status(200).send({ok: true})
})

//updateController
router.put('/profile', requireSignin, updateProfileController)

//user orders
router.get('/orders', requireSignin, getOrdersController)

//admin orders
router.get('/all-orders', requireSignin, isAdmin, getAllOrdersController)

//order Status update
router.put('/order-status/:orderId', requireSignin, isAdmin, orderStatusController)

module.exports = router;

```

Auth Controller

```

const { hashPassword, comparePassword } = require("../helpers/authHelper");
const userModel = require("../models/userModel")
const orderModel = require("../models/orderModel")
const JWT = require("jsonwebtoken")

//POST REGISTER

const registerController = async (req, res) => {
    try {
        const { name, email, password, phone, address, answer } = req.body
        //validations
        if (!name || !email || !password || !phone || !address || !answer) {
            return res.status(400).json({ message: "one or more mandatory fields are empty" })
        }
        //check user
        const existingUser = await userModel.findOne({ email: email })
        //existing user
        if (existingUser) {
            return res.status(200).send({
                success: false,
                message: 'Already Register Please Login'
            })
        }
    }
}

```

```

//register user
const hashedPassword = await hashPassword(password)
//save
const user = await new userModel({name, email, phone, address, answer, password:hashedPassword}).save()

res.status(201).send({
    success:true,
    message:"User Registered Success",
    user
})

} catch (error) {
    console.log(error)
    res.status(500).send({
        success:false,
        message:"Error in Registration",
        error
    })
}
};

//POST LOGIN

const loginController = async(req,res)=>{
try {
    const {email,password}=req.body
    //validation
    if(!email || !password){
        return res.status(404).send({
            success:false,
            message:"invalid email or password",
        })
    }
    //check user
    const user = await userModel.findOne({email})
    if(!user){
        return res.status(404).send({
            success:false,
            message:"Email not registered"
        })
    }
    const match = await comparePassword(password,user.password)
    if(!match){
        return res.status(200).send({
            success:false,
            message:"invalid password"
        })
    }
    //token
    const token = await JWT.sign({_id:user._id}, process.env.JWT_SECRET,{expiresIn:"7d"})
    res.status(200).send({
        success:true,
        message:"Logged in successfully",
        user:{
            name:user.name,
            email:user.email,
            phone:user.phone,
            address:user.address,
            role:user.role,
        },
        token,
    })
}

} catch (error) {
    console.log(error)
    res.status(500).send({
        success:false,
        message:"Error in Login",
        error
    })
}

```

```

        })
    }

//forgotPasswordController
const forgotPasswordController= async (req,res)=>{
try {
    const {email,newPassword,answer,}=req.body
    if(!email){
        res.status(400).send({
            success:false,
            message:"email is required"
        })
    }
    if(!answer){
        res.status(400).send({
            success:false,
            message:"answer is required"
        })
    }
    if(!newPassword){
        res.status(400).send({
            success:false,
            message:"New password is required"
        })
    }
    //check
    const user = await userModel.findOne({email,answer})
    //validation
    if(!user){
        return res.status(404).send({
            success:false,
            message:"wrong email or answer"
        })
    }
    const hashed = await hashPassword(newPassword)
    await userModel.findByIdAndUpdate(user._id,{password:hashed})
    res.status(200).send({
        success:true,
        message:"Password reset successfully"
    })
} catch (error) {
    console.log(error)
    res.status(500).send({
        success:false,
        message:"Something went wrong",
        error,
    })
}
}

//test controller

const testController = async(req,res)=>{
try {
    const user = await userModel.find({})
    res.json(user)
} catch (error) {
    console.log(error)
    res.status(500).send({
        success:false,
        message:"Something went wrong",
        error,
    })
}
}

```

```

//update User Profile
const updateProfileController=async(req,res)=>{
    try {
        const {name,email,password,phone,address} = req.body
        const user = await userModel.findById(req.user._id);
        //password
        if(password && password.length < 6){
            return res.json({error:"password is required and 6 character long"})
        }
        //hashedpassword
        const hashedPassword = password ? await hashPassword(password) : undefined;
        const updatedUser = await userModel.findByIdAndUpdate(req.user._id,{
            name:name || user.name,
            password:hashedPassword || user.password,
            phone:phone|| user.phone,
            address:address || user.address
        },{name:true})
        res.status(200).send({
            success:true,
            message:"profile updated Successfully",
            updatedUser,
        })
    } catch (error) {
        console.log(error)
        res.status(500).send({
            success:false,
            message:"Error while update user profile",
            error,
        })
    }
}

//user orders
const getOrdersController = async(req,res)=>{
    try {
        const orders = await orderModel.find({buyer:req.user._id})
        .populate("products", "-photo")
        .populate("buyer" , "name")
        res.json(orders)
    } catch (error) {
        console.log(error)
        res.status(500).send({
            success:false,
            message:"Error while user order",
            error,
        })
    }
}
}

//all-orders admin

const getAllOrdersController = async(req,res)=>{
    try{
        const orders = await orderModel.find({})
        .populate("products", "-photo")
        .populate("buyer" , "name")
        .sort({createdAt: "-1"})
        res.json(orders)
    }catch(error){
        console.log(error)
        res.status(500).send({
            success:false,
            message:"Error while occurring all orders",
            error,
        })
    }
}

//orderStatus Controller

```

```

const orderStatusController = async(req,res)=>{
try {
    const { orderId} = req.params;
    const {status} = req.body;
    const orders = await orderModel.findByIdAndUpdate(orderId,{status},{new:true});
    res.json((orders))
} catch (error) {
    console.log(error)
    res.status(500).send({
        success:false,
        message:"Error while occurring order Status",
        error,
    })
}
}

module.exports={
    registerController,
    loginController,
    testController,
    forgotPasswordController,
    updateProfileController,
    getOrdersController,
    getAllOrdersController,
    orderStatusController,
}

```

CategorySchema(CategoryModel.js)

```

const mongoose = require("mongoose")

const categorySchema = new mongoose.Schema({
    name:{
        type:String,
        required:true,
        unique:true
    },
    slug:{
        type:String,
        lowercase:true
    }
})

module.exports= mongoose.model('Category', categorySchema)

```

CategoryRoutes.js

```

const express = require("express")
const { createCategoryController ,updateCategoryController,categoryController ,singleCategoryController, deleteCategoryController}=
require("../controllers/categoryController")
const { isAdmin, requireSignin } = require("../middleware/authMiddleware")

const router = express.Router()

//routes
//create Category
router.post('/create-category' , requireSignin, isAdmin, createCategoryController)

//update category
router.put('/update-category/:id' ,requireSignin,isAdmin,updateCategoryController)

//getAll category
router.get("/get-category" , categoryController)

//single category
router.get('single-category/:slug' , singleCategoryController)

//delete category
router.delete('/delete-category/:id' , requireSignin, isAdmin, deleteCategoryController)

```

CategoryController.js

```

const { default: slugify } = require('slugify')
const categoryModel = require('../models/categoryModel')

//create category controller
const createCategoryController = async (req,res)=>{
  try {
    const {name}=req.body
    if(!name){
      res.send(401).send({
        message:'name is required'
      })
    }
    //check category
    const existingCategory = await categoryModel.findOne({name})
    //existing category
    if(existingCategory){
      return res.status(200).send({
        success:true,
        message:"Category already Exist"
      })
    }
    //save
    const category = await new categoryModel({name,slug:slugify(name)}).save()
    res.status(200).send({
      success:true,
      message:'new category created',
      category
    })
  } catch (error) {
    console.log(error)
    res.status(500).send({
      success:false,
      message:"Error in Category",
      error
    })
  }
}

//update category controller
const updateCategoryController = async(req,res)=>{
  try {
    const {name} = req.body
    const {id}= req.params
    const category = await categoryModel.findByIdAndUpdate(id,{name,slug:slugify(name)},{new:true})
    res.status(200).send({
      success:true,
      message:"category updated successfully",
      category
    })
  } catch (error) {
    console.log(error)
    res.status(500).send({
      success:false,
      message:"Error while updating category",
      error,
    })
  }
}

//get all category
const categoryController = async (req,res)=>{
  try {
    const category = await categoryModel.find({})
    res.status(200).send({
      success:true,
      message:" All category List",
      category
    })
  } catch (error) {
    console.log(error)
    res.status(500).send({
      success:false,
    })
  }
}

```

```

        message:"Error while getting all category",
        error,
    })
}

//get Single category controllers
const singleCategoryController = async (req,res) =>{
    try {

        const category = await categoryModel.findOne({slug:req.params.slug})
        res.status(200).send({
            success:true,
            message:"Get Single Category",
            category
        })
    } catch (error) {
        console.log(error)
        res.status(500).send({
            success:false,
            message:"Error while getting single category",
            error
        })
    }
}
//delete Category
const deleteCategoryController =async (req,res)=>{
    try {
        const {id}=req.params
        await categoryModel.findByIdAndDelete(id)
        res.status(200).send({
            success:true,
            message:"Delete category successfully"
        })
    } catch (error) {
        console.log(error)
        res.status(500).send({
            success:false,
            message:"Error while Deleting Category",
            error
        })
    }
}

module.exports={
    createCategoryController,
    updateCategoryController,
    categoryController,
    singleCategoryController,
    deleteCategoryController
}

```

ProductModel.js

```

const mongoose = require("mongoose")

const productSchema = new mongoose.Schema({
    name:{
        type:String,
        required:true
    },
    slug:{
        type:String,
        required:true
    },
    description:{
        type:String,
        required:true
    },
    price:{
        type:Number,
        required:true
    },
    category:{
        type:mongoose.ObjectId,
        ref:'Category',
        required:true
    }
})

```

```

    },
    quantity:{
      type:Number,
      required:true
    },
    photo:{
      data:Buffer,
      contentType:String,
    },
    shipping:{
      type:Boolean,
    }
  },{timestamps:true})

module.exports = mongoose.model("Products", productSchema)

```

ProductRoute.js

```

const express = require("express");
const {
  createProductController,
  getProductController,
  getSingleProductController,
  productPhotoController,
  deleteProductController,
  updateProductController,
  filterProductController,
  productCountController,
  productListController,
  searchProductController,
  relatedProductController,
  productCategoryController,
  braintreeTokenController,
  braintreePaymentController,
} = require("../controllers/productController");
const { isAdmin, requireSignin } = require("../middleware/authMiddleware");
const formidable = require("express-formidable");

const router = express.Router();

//routes

//create product
router.post(
  "/create-product",
  requireSignin,
  isAdmin,
  formidable(),
  createProductController
);

//get product
router.get("/get-product", getProductController);

//single product
router.get("/get-product/:slug", getSingleProductController);

//get photo based on product id
router.get("/product-photo/:pid", productPhotoController);

//update product
router.put(
  "/update-product/:pid",
  requireSignin,
  isAdmin,
  formidable(),
  updateProductController
);

//delete product
router.delete("/delete-product/:pid", deleteProductController);

//filter product
router.post("/product-filter", filterProductController);

```

```

//product count
router.get("/product-count", productCountController);

//product per page
router.get('/product-list/:page', productListController)

//search product
router.get('/search/:keyword', searchProductController)

//similer product
router.get('/related-product/:pid/:cid', relatedProductController)

//category wise product
router.get('/product-category/:slug', productCategoryController)

//payment route
//token
router.get('/braintree/token' , braintreeTokenController)

//payments
router.post('/braintree/payment', requireSignin,braintreePaymentController)

module.exports = router;

```

ProductController.js

```

const { default: slugify } = require("slugify");
const productModel = require("../models/productModel");
const categoryModel = require("../models/categoryModel");
const fs = require("fs");
const braintree = require("braintree");
const orderModel = require("../models/orderModel");
const dotenv = require("dotenv");
dotenv.config();

//paymentgateway
var gateway = new braintree.BraintreeGateway({
  environment: braintree.Environment.Sandbox,
  merchantId: process.env.BRAINTREE_MERCHANT_ID,
  publicKey: process.env.BRAINTREE_PUBLIC_KEY,
  privateKey: process.env.BRAINTREE_PRIVATE_KEY,
});

const createProductController = async (req, res) => {
  try {
    const { name, slug, description, price, category, quantity, shipping } =
      req.fields;
    const { photo } = req.files;
    //validation
    switch (true) {
      case !name:
        return res.status(500).send({ error: "Name is required" });
      case !description:
        return res.status(500).send({ error: "Description is required" });
      case !price:
        return res.status(500).send({ error: "Price is required" });
      case !category:
        return res.status(500).send({ error: "Category is required" });
      case !quantity:
        return res.status(500).send({ error: "Quantity is required" });
      case photo && photo.size > 1000000:
        return res
          .status(500)
          .send({ error: "photo is required and should be less then 1mb" });
    }
    const products = new productModel({ ...req.fields, slug: slugify(name) });
    if (photo) {
      products.photo.data = fs.readFileSync(photo.path);
      products.photo.contentType = photo.type;
    }
    await products.save();
    res.status(200).send({
      success: true,
    });
  } catch (err) {
    console.log(err);
    res.status(500).send({
      error: "Internal Server Error",
    });
  }
};

const updateProductController = async (req, res) => {
  try {
    const { id } = req.params;
    const { name, slug, description, price, category, quantity, shipping } =
      req.fields;
    const { photo } = req.files;
    const products = await productModel.findById(id);
    if (!products) {
      return res.status(404).send({ error: "Product not found" });
    }
    products.name = name;
    products.slug = slug;
    products.description = description;
    products.price = price;
    products.category = category;
    products.quantity = quantity;
    products.shipping = shipping;
    if (photo) {
      products.photo.data = fs.readFileSync(photo.path);
      products.photo.contentType = photo.type;
    }
    await products.save();
    res.status(200).send({
      success: true,
    });
  } catch (err) {
    console.log(err);
    res.status(500).send({
      error: "Internal Server Error",
    });
  }
};

const deleteProductController = async (req, res) => {
  try {
    const { id } = req.params;
    const products = await productModel.findById(id);
    if (!products) {
      return res.status(404).send({ error: "Product not found" });
    }
    await products.remove();
    res.status(200).send({
      success: true,
    });
  } catch (err) {
    console.log(err);
    res.status(500).send({
      error: "Internal Server Error",
    });
  }
};

const getAllProductsController = async (req, res) => {
  try {
    const products = await productModel.find();
    res.status(200).send({
      products,
    });
  } catch (err) {
    console.log(err);
    res.status(500).send({
      error: "Internal Server Error",
    });
  }
};

const getSingleProductController = async (req, res) => {
  try {
    const { id } = req.params;
    const products = await productModel.findById(id);
    if (!products) {
      return res.status(404).send({ error: "Product not found" });
    }
    res.status(200).send({
      products,
    });
  } catch (err) {
    console.log(err);
    res.status(500).send({
      error: "Internal Server Error",
    });
  }
};

```

```

        message: "Product created successfull",
        products,
    });
} catch (error) {
    console.log(error);
    res.status(500).send({
        success: false,
        message: "Error while creating product",
        error,
    });
}
};

//get all controller

const getProductsController = async (req, res) => {
    try {
        const products = await productModel
            .find({})
            .populate("category")
            .select("-photo")
            .limit(12)
            .sort({ createdAt: -1 });
        res.status(200).send({
            success: true,
            message: "All Products",
            products,
            totalcount: products.length,
        });
    } catch (error) {
        console.log(error);
        res.status(500).send({
            success: false,
            message: "Error while getting Product",
            error,
        });
    }
};

//get single product

const getSingleProductController = async (req, res) => {
    try {
        const product = await productModel
            .findOne({ slug: req.params.slug })
            .select("-photo")
            .populate("category");
        res.status(200).send({
            success: true,
            message: "Single product fetch",
            product,
        });
    } catch (error) {
        console.log(error);
        res.status(500).send({
            success: false,
            message: "Error while getting single product",
            error,
        });
    }
};

//get photo
const productPhotoController = async (req, res) => {
    try {
        const product = await productModel.findById(req.params.pid).select("photo");
        if (product.photo.data) {
            res.set("Content-type", product.photo.contentType);
            return res.status(200).send(product.photo.data);
        }
    } catch (error) {
        console.log(error);
        res.status(500).send({
            success: false,
            message: "Error while getting photo",
            error,
        });
    }
};

//update product

const updateProductController = async (req, res) => {

```

```

try {
    const { name, slug, description, price, category, quantity, shipping } =
        req.fields;
    const { photo } = req.files;
    //validation
    switch (true) {
        case !name:
            return res.status(500).send({ error: "Name is required" });
        case !description:
            return res.status(500).send({ error: "Description is required" });
        case !price:
            return res.status(500).send({ error: "Price is required" });
        case !category:
            return res.status(500).send({ error: "Category is required" });
        case !quantity:
            return res.status(500).send({ error: "Quantity is required" });
        case photo && photo.size > 1000000:
            return res
                .status(500)
                .send({ error: "photo is required and should be less than 1mb" });
    }
}

const products = await productModel.findByIdAndUpdate(req.params.pid,
    {...req.fields, slug:slugify(name)}, {new:true});
if (photo) {
    products.photo.data = fs.readFileSync(photo.path);
    products.photo.contentType = photo.type;
}
await products.save();
res.status(200).send({
    success: true,
    message: "Product updated successfull",
    products,
});
} catch (error) {
    console.log(error);
    res.status(500).send({
        success: false,
        message: "Error while update product",
        error,
    });
}
};

//delete product

const deleteProductController = async (req, res) => {
    try {
        await productModel.findByIdAndDelete(req.params.pid).select("-photo");
        res.status(200).send({
            success: true,
            message: "product deleted Successfull",
        });
    } catch (error) {
        console.log(error);
        res.status(500).send({
            success: false,
            message: "Error while deleting product",
            error,
        });
    }
};

//product-filter controller
const filterProductController = async (req, res) => {
    try {
        const { checked, radio } = req.body
        let args = {}
        if (checked.length > 0) args.category = checked
        if (radio.length) args.price = {$gte: radio[0], $lte: radio[1]}
        const products = await productModel.find(args)
        res.status(200).send({
            success: true,
            message: "products save successfull",
            products,
        })
    } catch (error) {
        console.log(error);
        res.status(500).send({
            success: false,
            message: "Something went wrong while filter",
            error,
        });
    }
};

```

```

        })
    }

//product count controller
const productCountController = async(req,res)=>{
    try {
        const total = await productModel.find({}).estimatedDocumentCount()
        res.status(200).send({
            success:true,
            message:"product count successfull",
            total,
        })
    } catch (error) {
        console.log(error),
        res.status(500).send({
            success:false,
            message:"Error while getting product count",
            error,
        })
    }
}

//product list base on page
const productListController = async(req,res)=>{
    try {
        const perPage = 3
        const page = req.params.page ? req.params.page : 1
        const products =await productModel.find({})
        .select("-photo")
        .skip((page-1) * perPage)
        .limit(perPage)
        .sort({createdAt: -1})
        res.status(200).send({
            success:true,
            message:"product list getting successfull",
            products
        })
    } catch (error) {
        console.log(error),
        res.status(500).send({
            success:false,
            message:"Error while getting product list",
            error,
        })
    }
}

//search product controller
const searchProductController= async( req,res)=>{
try {
    const {keyword} = req.params;
    const results = await productModel.find({
        $or:[
            {name:{$regex:keyword, $options:"i"}},
            {description:{$regex:keyword, $options:"i"}}
        ]
    }).select("-photo");
    res.json(results)
} catch (error) {
    console.log(error);
    res.status(500).send({
        success:false,
        message:"Error while getting search product",
        error
    })
}
}

//similer related product controller
const relatedProductController = async(req,res)=>{
try {
    const {pid,cid}= req.params;
    const products = await productModel.find({
        category:cid,
        _id:{$ne:pid}
    }).select("-photo").limit(3).populate("category")
    res.status(200).send({
        success:true,
        products
    })
}
}

```

```

    } catch (error) {
      console.log(error)
      res.status(500).send({
        success:false,
        message:"error while getting similer product",
        error
      })
    }
  }

//category wise controller
const productCategoryController = async(req,res)=>{
try {
  const category = await categoryModel.findOne({slug:req.params.slug})
  const products = await productModel.find({category}).populate('category')
  res.status(200).send({
    success:true,
    category,products
  })
} catch (error) {
  console.log(error)
  res.status(400).send({
    success:false,
    message:"Error while getting product"
    ,error
  })
}
}

//payment gateway api
//token
const braintreeTokenController =async(req,res)=>{
  try {
    gateway.clientToken.generate({},function(err,response){
      if(err){
        res.status(500).send(err)
      }else{
        res.send(response)
      }
    })
  } catch (error) {
    console.log(error)
  }
}

//payment
const braintreePaymentController =async(req,res)=>{
  try {
    const{cart,nonce}=req.body;
    let total = 0
    cart.map( (i) =>{total += i.price})
    let newTransection = gateway.transaction.sale({
      amount:total,
      paymentMethodNonce:nonce,
      options:{
        submitForSettlement:true
      }
    },
    function(error,result){
      if(result){
        const order =new orderModel({
          products:cart,
          payment:result,
          buyer:req.user._id
        }).save()
        res.json({ok:true})
      }else{
        res.status(500).send(error)
      }
    )
  } catch (error) {
    console.log(error)
  }
}

module.exports = {
  createProductController,
  getProductController,
  getSingleProductController,
  productPhotoController,
}

```

```

updateProductController,
deleteProductController,
filterProductController,
productCountController,
productListController,
searchProductController,
relatedProductController,
productCategoryController,
braintreeTokenController,
braintreePaymentController,
};

}

```

orderModel.js

```

const mongoose = require("mongoose")

const orderSchema = new mongoose.Schema({
  products:[{
    type : mongoose.ObjectId,
    ref:"Products"
  },
],
payment:{},
buyer:{ 
  type:mongoose.ObjectId,
  ref:"users"
},
status:{ 
  type:String,
  default:"Not Process",
  enum:["Not Process",'processing',"shipped","deliver", "cancle"]
}
}, {timestamps:true})

module.exports= mongoose.model('Order', orderSchema)

```

Helper.js(bcrypt password)

```

const bcrypt = require("bcrypt")

//bcrypt hashPassword
const hashPassword = async(password)=>{
  try {
    const saltRound = 10;
    const hashedPassword = await bcrypt.hash(password,saltRound)
    return hashedPassword
  } catch (error) {
    console.log((error))
  }
}

//bcrypt comparePassword
const comparePassword = async(password,hashPassword)=>{
  return bcrypt.compare(password,hashPassword)
}

module.exports = {hashPassword,comparePassword}

```

middleware(Authmiddleware)

```

const JWT = require("jsonwebtoken")
const userModel = require("../models/userModel")

//Protected routes token base
const requireSignin = async(req,res,next)=>{
  try {
    const decode = JWT.verify(req.headers.authorization,process.env.JWT_SECRET)
    req.user=decode
    next()
  } catch (error) {
    console.log(error)
  }
}

```

```

}

//admin access

const isAdmin = async (req,res,next)=>{
try {
  const user = await userModel.findById(req.user._id)
  if(user.role !==1){
    return res.status(401).send({
      success:false,
      message:"UnAuthorized Access"
    })
  }else{
    next();
  }
} catch (error) {
  console.log(error)
}
}

module.exports = {requireSignin, isAdmin }

```

.env file

```

PORT =8080
DEV_MODE = development
MONGO_URL = mongodb+srv://ecommerce:1234@cluster1.wxbg.mongodb.net/ecommerce
JWT_SECRET = aghshhajkjskhxjbzasjkjhjhx
BRAINTREE_MERCHANT_ID = z8njmz2d82m4
BRAINTREE_PUBLIC_KEY= v2rt7yhxq7r77
BRAINTREE_PRIVATE_KEY =7edd02a5fae25dc9ba4e3fb9a26962

```

Package.json

```

{
  "name": "ecommerce_backend",
  "version": "1.0.0",
  "description": "ecommerce rest api",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "server": "nodemon server.js",
    "client": "npm start --prefix ../ecommerce_frontend",
    "dev": "concurrently \"npm run server\" \"npm run client\""
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcrypt": "^5.1.0",
    "braintree": "^3.13.0",
    "color": "^4.2.3",
    "colors": "^1.4.0",
    "concurrently": "^7.6.0",
    "cors": "^2.8.5",
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "express-formidable": "^1.2.0",
    "jsonwebtoken": "^9.0.0",
    "mongoose": "^6.9.1",
    "morgan": "^1.10.0",
    "nodemon": "^2.0.20",
    "slugify": "^1.6.5",
    "stripe": "^11.12.0"
  }
}

```

Frontend Index.js

```
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 'antd/dist/reset.css';
import { SearchProvider } from './context/search';
import { CartProvider } from './context/cart';

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

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

App.js

```
import {Routes,Route} from "react-router-dom"
import About from "./pages/About";
import Contact from "./pages/Contact";
import HomePage from "./pages/HomePage";
import PagenotFound from "./pages/PagenotFound";
import Policy from "./pages/Policy";
import Register from "./pages/Auth/Register"
import Login from "./pages/Auth/Login";
import DashBoard from "./pages/user/DashBoard";
import PrivateRoute from "./component/Routes/Private";
import ForgotPassword from "./pages/Auth/ForgotPassword";
import AdminRoute from "./component/Routes/AdminRoutes";
import AdminDashboard from "./pages/Admin/AdminDashboard";
import CreateCategory from "./pages/Admin/CreateCategory";
import CreateProduct from "./pages/Admin/CreateProduct";
import User from "./pages/Admin/User";
import Profile from "./pages/user/Profile";
import Orders from "./pages/user/Orders";
import Products from "./pages/Admin/Products";
import UpdateProduct from "./pages/Admin/UpdateProduct";
import Search from "./pages/Search";
import ProductDetails from "./pages/ProductDetails";
import Categories from "./pages/Categories";
import CategoryProduct from "./pages/CategoryProduct";
import CartPage from "./pages/CartPage";
import AdminOrders from "./pages/Admin/AdminOrders";

function App() {
  return (
    <>
    <Routes>
      <Route path='/' element={<HomePage/>}/>
      <Route path='/product/:slug' element={<ProductDetails/>}/>
      <Route path='/categories' element={<Categories/>}/>
      <Route path='/cart' element={<CartPage/>}/>
      <Route path='/category/:slug' element={<CategoryProduct/>}/>
    </Routes>
  )
}

export default App;
```

```

<Route path='/search' element={<Search/>}/>
<Route path='/dashboard' element={<PrivateRoute/>}>
<Route path='user' element={<DashBoard/>}/>
<Route path='user/profile' element={<Profile/>}/>
<Route path='user/orders' element={<Orders/>}/>
</Route>
<Route path='/dashboard' element = {<AdminRoute/>}>
<Route path='admin' element={<AdminDashboard/>}/>
<Route path='admin/create-category' element={<CreateCategory/>}/>
<Route path='admin/create-product' element={<CreateProduct/>}/>
<Route path='admin/product/:slug' element={<UpdateProduct/>}/>
<Route path='admin/products' element={<Products/>}/>
<Route path='admin/users' element={<User/>}/>
<Route path='admin/orders' element={<AdminOrders/>}/>
</Route>
<Route path='/register' element={<Register/>}/>
<Route path='/forgot-password' element={<ForgotPassword/>}/>
<Route path='/login' element={<Login/>}/>
<Route path='/about' element={<About/>}/>
<Route path='/contact' element={<Contact/>}/>
<Route path='/policy' element={<Policy/>}/>
<Route path='*' element={<PagenotFound/>}/>
</Routes>
</>
);
};

}

export default App;

```

Component/Layout/Header.js =>(in this component created Navigation bar

```

import React from 'react'
import { NavLink ,Link} from 'react-router-dom'
import {GiShoppingBag} from "react-icons/gi"
import { useAuth } from '../../../../../context/auth'
import toast from "react-hot-toast"
import SearchInput from '../Form/SearchInput'
import useCategory from '../../../../../hooks/useCategory'
import { useCart } from '../../../../../context/cart'
import{Badge} from "antd"

const Header = () => {
  const [auth, setAuth] = useAuth()
  const [cart] = useCart()
  const categories = useCategory()
  const handleLogout = ()=>{
    setAuth({
      ...auth, user: null, token: ''
    })
    localStorage.removeItem('auth')
    toast.success("Successfully Logout")
  }
  return (
    <>
      <nav className="navbar navbar-expand-lg bg-body-tertiary">
        <div className="container-fluid">
          <Link to="/" className="navbar-brand">
            <GiShoppingBag/>
            ECOMMERCE APP</Link>
          <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span className="navbar-toggler-icon"></span>
          </button>
          <div className="collapse navbar-collapse" id="navbarSupportedContent">
            <ul className="navbar-nav ms-auto mt-2 mb-2 mb-lg-0">
              <SearchInput/>
              <li className="nav-item">
                <NavLink to='/' className="nav-link">Home</NavLink>
              </li>
              <li className="nav-item dropdown">
                <Link className="nav-link dropdown-toggle" to="/categories" data-bs-toggle="dropdown" >
                  Category
                </Link>
              </li>
            </ul>
          </div>
        </div>
      </nav>
    </>
  )
}


```

```


- <Link className="dropdown-item" to="/categories">All Categories</Link></li>
  {categories?.map((citem)=>(
    <li key={citem._id}><Link className="dropdown-item" to={`/category/${citem.slug}`}>{citem.name}</Link></li>
  )))
  </ul>
</li>

{!auth.user ? (<>
  <li className="nav-item">
    <NavLink to='/register' className="nav-link" >REGISTER</NavLink>
  </li>
  <li className="nav-item">
    <NavLink to='/login' className="nav-link" >LOGIN</NavLink>
  </li>
  </>) : (<>

  <ul className="navbar-nav">
    <li className="nav-item dropdown">
      <NavLink className="nav-link dropdown-toggle" to="" id="navbarDarkDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
        {auth?.user?.name}
      </NavLink>
      <ul className="dropdown-menu dropdown-menu-dark" aria-labelledby="navbarDarkDropdownMenuLink">
        <li className="nav-item"><NavLink className="dropdown-item" to = {`/dashboard/${auth?.user?.role === 1 ? 'admin' : 'user'}`}>DashBoard</NavLink></li>
        <li className="nav-item">
          <NavLink onClick={handleLogout} to='/login' className="dropdown-item">LOGOUT</NavLink>
        </li>
      </ul>
    </li>
  </ul>
</>)
  </>
)
}

export default Header

```

component/Layout/footer =>(in this component
created footer bar)

```

import React from 'react'
import { Link } from 'react-router-dom'

const Footer = () => {
  return (
    <div className='footer'>
      <h4 className='text-center'>All Right Reserve &copy; 2023</h4>
      <p className='text-center mt-3'>
        <Link to="/about">About</Link>
        |
        <Link to="/contact">Contact</Link>
        |
        <Link to="/policy">Policy</Link>
      </p>
    </div>
  )
}

export default Footer

```

component/Layout/Layout.js =>(call header and footer component inside Layout component and also use here react-helmet for SEO call this component all page for dynamic using header main and footer

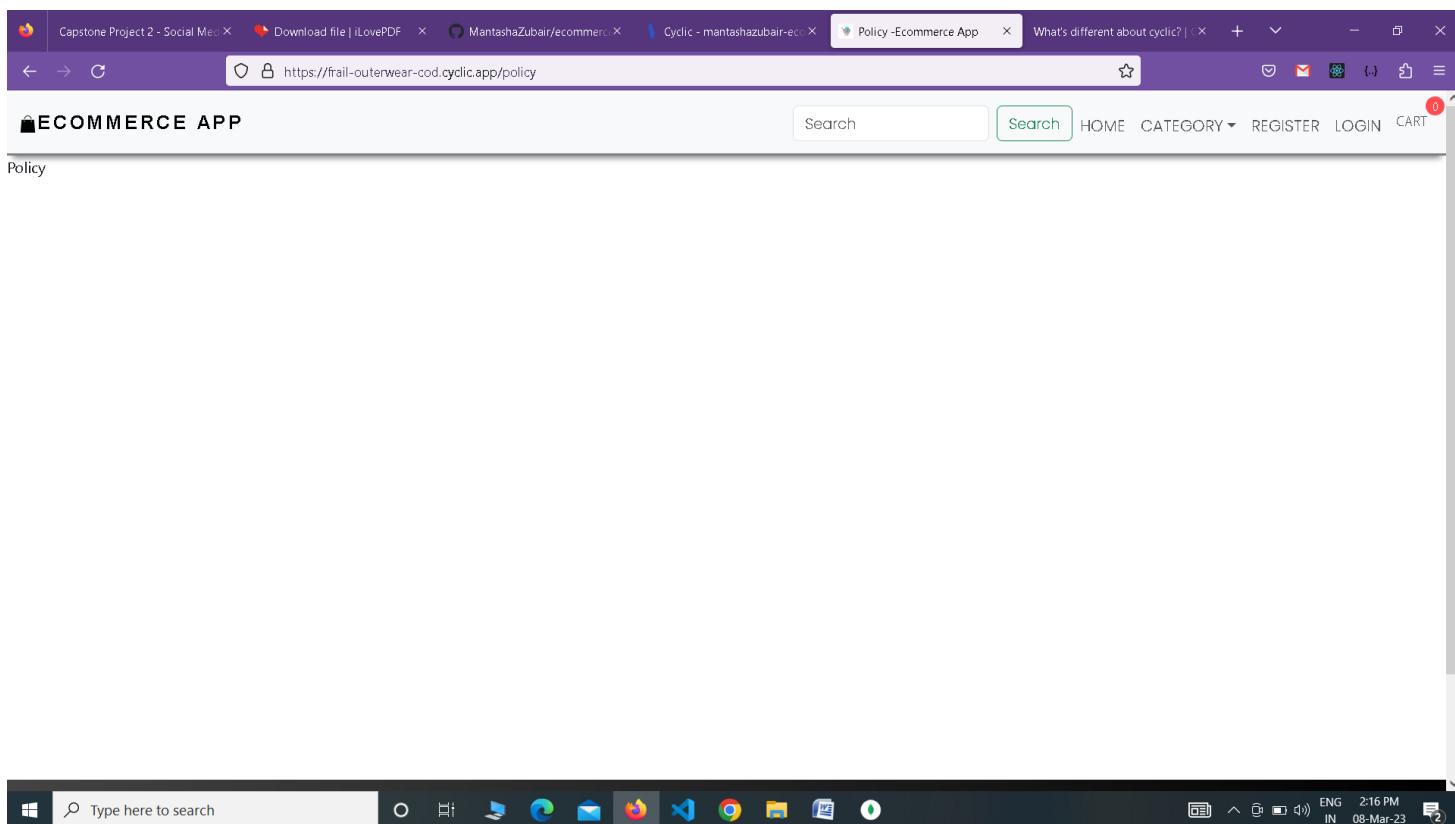
```
import React from 'react'
import Footer from './Footer'
import Header from './Header'
import {Helmet} from 'react-helmet'
import { Toaster } from 'react-hot-toast';

const Layout = (props) => {
  return (
    <div>
      <Helmet>
        <meta charSet="utf-8" />
        <meta name="description" content={props.description} />
        <meta name="keywords" content={props.keywords} />
        <meta name="author" content={props.author} />
        <title>{props.title}</title>
      </Helmet>
      <Header/>
      <main style={{minHeight:"82vh"}}>
        <Toaster/>
        {props.children}
      </main>
      <Footer/>
    </div>
  )
}

Layout.defaultProps ={
  title:'Ecommerce app -shop now',
  description:"mern-stack project",
  keywords:'mern , react,node,mongodb',
  author:"mantasha"
}

export default Layout
```

output



pages/Auth/Register.js =>{

in this component create a register form where use multiple State and post it api/v1/auth/register when form is submit then and pass all state

```
import React, { useState } from "react";
import Layout from "../../component/Layout/Layout";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import toast from "react-hot-toast";
import "./Auth.css";

const Register = () => {
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [phone, setPhone] = useState("");
  const [address, setAddress] = useState("");
  const [answer, setAnswer] = useState("");
  const navigate = useNavigate();

  //Form submit handler
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const res = await axios.post(
        `${process.env.REACT_APP_API}/api/v1/auth/register`,
        { name, email, password, phone, address, answer }
      );
      if (res.data.success) {
        toast.success(res.data.message);
        navigate("/login");
      } else {
        toast.error(res.data.message);
      }
    } catch (error) {
      console.log(error);
      toast.error("Something Went wrong");
    }
  };
}

return (
  <Layout title="Register - E-commerce App">
```

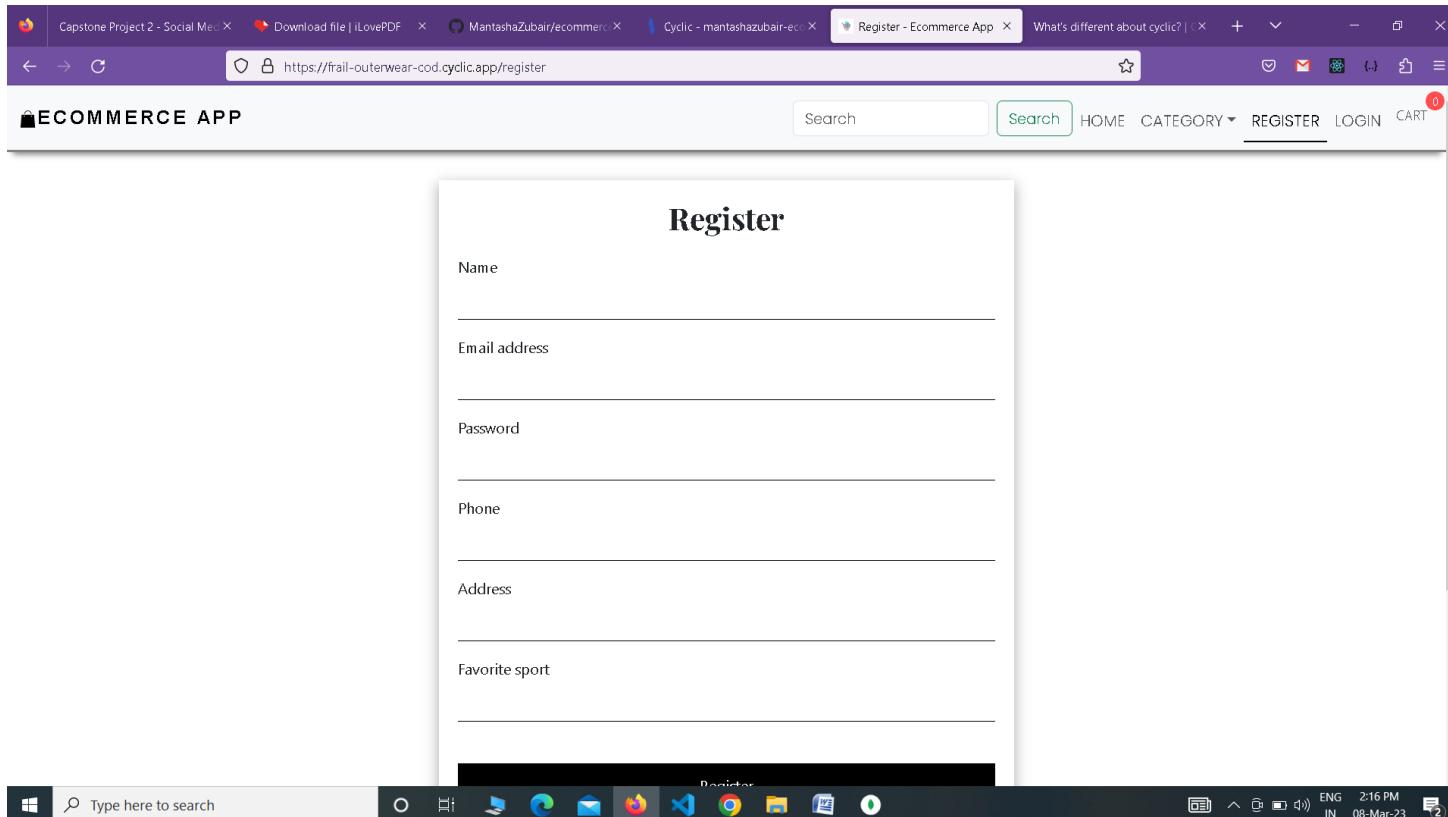
```

<div className="form-container">
  <form onSubmit={handleSubmit}>
    <h4 className="title">Register</h4>
    <div className="mb-3">
      <label htmlFor="name" className="form-label">
        Name
      </label>
      <input
        type="text"
        name="name"
        className="form-control"
        value={name}
        onChange={(e) => setName(e.target.value)}
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="exampleInputEmail1" className="form-label">
        Email address
      </label>
      <input
        type="email"
        name="email"
        className="form-control"
        value={email}
        onChange={(e) => setEmail(e.target.value)}
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="exampleInputPassword1" className="form-label">
        Password
      </label>
      <input
        type="password"
        name="password"
        className="form-control"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="phone" className="form-label">
        Phone
      </label>
      <input
        type="text"
        name="phone"
        className="form-control"
        value={phone}
        onChange={(e) => setPhone(e.target.value)}
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="address" className="form-label">
        Address
      </label>
      <input
        type="text"
        name="address"
        className="form-control"
        value={address}
        onChange={(e) => setAddress(e.target.value)}
        required
      />
    </div>
    <div className="mb-3">
      <label htmlFor="address" className="form-label">
        Favorite sport
      </label>
      <input
        type="text"
        name="answer"
        className="form-control"
        value={answer}
        onChange={(e) => setAnswer(e.target.value)}
        required
      />
    </div>
  </div>

```

```
        <button type="submit" className="btn btn-primary">
          Register
        </button>
      </div>
    </form>
  </div>
</Layout>
);
};

export default Register;
```



Pages/Auth/Login.js=> in this page use context api for checking user and token if is it valid user or admin then access the dashboard also here use some protected route or user and admin dashboard no one access user and admin dashboard when they are not login their valid email and password

```
import React, { useState } from "react";
import Layout from "../../component/Layout/Layout";
import { useLocation, useNavigate } from "react-router-dom";
import axios from "axios"
import toast from 'react-hot-toast';
import "./Auth.css";
import { useAuth } from "../../context/auth";

const Login = () => {

const [email, setEmail] = useState("");
const [password, setPassword] = useState("");
const [auth, setAuth] = useAuth()

const navigate = useNavigate()
const location = useLocation()

//Form submit handler
const handleSubmit = async(e)=>{
  e.preventDefault()
  try {
    const res = await axios.post(
      `${process.env.REACT_APP_API}/api/v1/auth/login`,
      {email,password}
    )
    if(res.data.success){
      toast.success(res.data.message)
      setAuth({
        ...auth,
        user:res.data.user,
        token:res.data.token,
      })
      localStorage.setItem('auth',JSON.stringify(res.data))
      navigate(location.state || "/")
    }else{
      toast.error(res.data.message)
    }
  } catch (error) {
    console.log(error)
    toast.error("Something Went wrong")
  }
}

return (
  <Layout title="Login - Ecommerce App">
    <div className="form-container ">
      <form onSubmit={handleSubmit}>
        <h4 className="title">
          Login
        </h4>
        <div className="mb-3">
          <label htmlFor="exampleInputEmail1" className="form-label">
            Email address
          </label>
          <input
            type="email"
            name="email"
            className="form-control"
            value={email}
            onChange={(e)=>setEmail(e.target.value)}
            required
          />
        </div>
        <div className="mb-3">
          <label
            htmlFor="exampleInputPassword1"
            className="form-label"
          >
            Password
          </label>
          <input
            type="password"
          />
        </div>
      </form>
    </div>
  </Layout>
)
```

```

        name="password"
        className="form-control"
        value={password}
        onChange={(e)=>setPassword(e.target.value)}
        required
      />
    </div>

    <div className="mb-md-5 mb-2 mt-4 d-grid">
      <button type="button" className="btn btn-primary mb-4" onClick={()=>navigate('/forgot-password')}>
        Forgot Password
      </button>
      <button type="submit" className="btn btn-primary">
        Login
      </button>
    </div>

    </form>
  </div>
</Layout>
);
};

export default Login;

```

pages/Auth/ForgotPassword.js

```

import React, { useState } from "react";
import Layout from "../../component/Layout/Layout";
import { useNavigate } from "react-router-dom";
import axios from "axios";
import toast from 'react-hot-toast';
import "./Auth.css";

const ForgotPassword = () => {
  const [email, setEmail] = useState("");
  const [newPassword, setNewPassword] = useState("");
  const [answer, setAnswer] = useState("");

  const navigate = useNavigate();

  //Form submit handler
  const handleSubmit = async(e)=>{
    e.preventDefault()
    try {
      const res = await axios.post(
        `${process.env.REACT_APP_API}/api/v1/auth/forgot-password` ,
        {email,newPassword,answer})
      if (res.data.success) {
        toast.success(res.data.message);
        navigate("/login");
      }else {
        toast.error(res.data.message);
      }
    } catch (error) {
      console.log(error)
      toast.error("Something Went wrong")
    }
  }

  return (
    <Layout title="Forgot Password - Ecommerce App" >
      <div className="form-container">
        <form onSubmit={handleSubmit}>
          <h4 className="title">
            Forgot Password
          </h4>
          <div className="mb-3">
            <label htmlFor="exampleInputEmail1" className="form-label">
              Email address
            </label>
            <input
              type="email"
              name="email"
              className="form-control"
              value={email}
              onChange={(e)=>setEmail(e.target.value)}
              required
            />
          </div>
          <div className="mb-3">
            <label htmlFor="exampleInputEmail1" className="form-label">

```

```

        Enter your Favorite Sport
    </label>
    <input
        type="text"
        name="answer"
        className="form-control"
        value={answer}
        onChange={(e)=>setAnswer(e.target.value)}
        required
    />
</div>
<div className="mb-3">
    <label
        htmlFor="exampleInputPassword1"
        className="form-label"
    >
        New Password
    </label>
    <input
        type="password"
        name="newPassword"
        className="form-control"
        value={newPassword}
        onChange={(e)=>setNewPassword(e.target.value)}
        required
    />
</div>

<div className="mb-md-5 mb-2 mt-4 d-grid">
    <button type="submit" className="btn btn-primary">
        Reset
    </button>
</div>
</form>
</div>
</Layout>
)
}

export default ForgotPassword

```

Context /auth.js

```

import axios from "axios";
import { useState,useContext,createContext, useEffect } from "react";

const AuthContext =createContext()

const AuthProvider =({children})=>{
    const [auth,setAuth] = useState({
        user:null,
        token:""
    })
    //default axios
    axios.defaults.headers.common['Authorization']= auth?.token

    useEffect(()=>{
        const data= localStorage.getItem('auth')
        if(data){
            const parseData= JSON.parse(data)
            setAuth({
                ...auth,
                user:parseData.user,
                token:parseData.token
            })
        }
        //eslint-disable-next-line
    },[])
    return (

```

```

        <AuthProvider value={[auth, setAuth]}>
        {children}
        </AuthProvider>
    )
}

//custom hook

const useAuth = ()=> useContext(AuthContext)
export {useAuth, AuthProvider}

```

component/Routes/PrivateRoute.js (userRoute) => import outlet from react-router-dom for created nested route in this component create one state ok and setOk initial value is false inside useEffect create arrow function and get user-auth and check registerSingnIn if ok then return nested outlet other wise redirecete to the Spinner (Login page)

```

import axios from "axios";
import { useState, useEffect } from "react";
import { Outlet } from "react-router-dom";
import { useAuth } from "../../context/auth";
import Spinner from "./Spinner";

export default function PrivateRoute() {
    const [ok, setOk] = useState(false)
    const [auth, setAuth] = useAuth()

    useEffect(()=>{
        const authCheck = async ()=>{
            const res = await axios.get(`process.env.REACT_APP_API}/api/v1/auth/user-auth`)
            if(res.data.ok){
                setOk(true)
            }else{
                setOk(false)
            }
        }
        if(auth?.token) authCheck();
    },[auth?.token])

    return ok ? <Outlet/> : <Spinner/>
}

```

Component/Spinner

```

import React, {useState, useEffect} from 'react'
import { useLocation, useNavigate } from 'react-router-dom'

const Spinner = ({path ='login'}) => {
    const [count, setCount] = useState(3)
    const navigate = useNavigate()
    const location = useLocation()

    useEffect(()=>{
        const interval = setInterval(()=>{
            setCount((preValue)=> --preValue)
        }, 1000)
        count === 0 && navigate(`/${path}`, {
            state: location.pathname,
        })
        return ()=> clearInterval(interval)
    }, [count, navigate, location, path])
    return (
        <>
        <div className="d-flex flex-column justify-content-center align-items-center" style={{height:"100vh"}}>
        <h1 className='text-center'>redirecting to you in {count} second</h1>
        <div className="spinner-border" role="status">
            <span className="visually-hidden">Loading...</span>
        </div>
        </div>
        </>
    )
}

export default Spinner

```

AuthRoute.js=>same as userRoute but in this arrow function check if role==1 then redirect to the admin route

```
import axios from "axios";
import { useState, useEffect } from "react";
import { Outlet } from "react-router-dom";
import { useAuth } from "../../context/auth";
import Spinner from "./Spinner";

export default function AdminRoute() {
  const [ok, setOk] = useState(false);
  const [auth] = useAuth();

  useEffect(() => {
    const authCheck = async () => {
      const res = await axios.get(`process.env.REACT_APP_API}/api/v1/auth/admin-auth`);
      if (res.data.ok) {
        setOk(true);
      } else {
        setOk(false);
      }
    }
    if (auth?.token) authCheck();
  }, [auth?.token]);
}

return ok ? <Outlet/> : <Spinner path=" " />
}
```

AdminMenu

```
import React from 'react';
import { NavLink } from 'react-router-dom';
const AdminMenu = () => {
  return (
    <>
      <div className="text-center">
        <div className="list-group">
          <h4>Admin Panel</h4>
          <NavLink to="/dashboard/admin/create-category" className="list-group-item list-group-item-action">Create Category</NavLink>
          <NavLink to="/dashboard/admin/create-product" className="list-group-item list-group-item-action">Create Product</NavLink>
          <NavLink to="/dashboard/admin/products" className="list-group-item list-group-item-action">Products</NavLink>
          <NavLink to="/dashboard/admin/users" className="list-group-item list-group-item-action">Users</NavLink>
          <NavLink to="/dashboard/admin/orders" className="list-group-item list-group-item-action">Orders</NavLink>
        </div>
      </div>
    </>
  )
}

export default AdminMenu
```

pages/Admin/AdminDashboard.js

```
import React from 'react';
import AdminMenu from '../../component/Layout/AdminMenu';
import { useAuth } from '../../context/auth';
import Layout from '../../component/Layout/Layout';

const AdminDashboard = () => {
  const [auth] = useAuth();
  return (
    <Layout>
      <div className='container'>
        <div className='row m-3 p-3'>
          <div className='col-md-4 mb-4'>
            <AdminMenu/>
          </div>
          <div className='col-md-8'>
            <div className='card p-3 text-success'>
              <h3> Name :{auth?.user?.name}</h3>
              <h3> Email :{auth?.user?.email}</h3>
              <h3> Phone :{auth?.user?.phone}</h3>
            </div>
          </div>
        </div>
      </div>
    
```

```

        </div>
      </Layout>
    )
}

export default AdminDashboard

```

Ecommerce APP

Search

HOME CATEGORY ADMIN CART

Name :admin

Email :admin@gmail.com

Phone :7897744166

Admin Panel

- Create Category
- Create Product
- Products
- Users
- Orders

pages/Admin/CreateCategory.js

```

import React, {useEffect, useState} from 'react'
import AdminMenu from '../../../../../component/Layout/AdminMenu'
import Layout from '../../../../../component/Layout/Layout'
import toast from 'react-hot-toast';
import axios from "axios"
import CategoryForm from '../../../../../component/Form/CategoryForm';

import {Modal} from "antd"
const CreateCategory = () => {
  const [categories, setCategories]= useState([])
  const [name, setName]= useState('')
  const [visible, setVisible]= useState(false)
  const [selected, setSelected] = useState(null)
  const [updatedName, setUpdatedName]= useState("")

//handle.Form
const handleSubmit = async(e)=>{
  e.preventDefault();
  try {
    const {data} = await axios.post(`process.env.REACT_APP_API}/api/v1/category/create-category` ,{name})
    if(data?.success){
      toast.success(`${name} is created`)
      getAllCategories();
      setName("")
    }else{
      toast.error(data.message)
    }
  } catch (error) {
    console.log(error)
    toast.error("Something went wrong in input Form")
  }
}

```

```

//get all categories
const getAllCategories =async ()=>{
  try {
    const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/category/get-category`)
    if(data?.success){
      setCategories(data?.category);
    }
  } catch (error) {
    console.log(error)
    toast.error('some thing went wrong in getting category')
  }
};

useEffect(()=>{
  getAllCategories()
},[])

//update Category

const handleUpdate = async(e)=>{
  e.preventDefault()
  try {
    const {data}= await axios.put(` ${process.env.REACT_APP_API}/api/v1/category/update-category/${selected._id}`,{name:updatedName})
    if(data.success){
      toast.success(`${updatedName} is updated`)
      setSelected(null)
      setUpdatedName("")
      setVisible(false)
      getAllCategories()
    }else{
      toast.error(data.message)
    }
    console.log(e)
  } catch (error) {
    toast.error("Someting went wrong")
    console.log(error)
  }
}

//delete Category

const handleDelete = async(pid)=>{
  try {
    const {data}= await axios.delete(` ${process.env.REACT_APP_API}/api/v1/category/delete-category/${pid}`)
    if(data.success){
      toast.success(`deleted successfully`)

      getAllCategories()
    }else{
      toast.error(data.message)
    }
  } catch (error) {
    toast.error("Someting went wrong")
    console.log(error)
  }
}

return (
<Layout title = "Dashboard - Create Category">
  <div className='container mt-5'>
    <div className='row'>
      <div className='col-md-3'>
        <AdminMenu/>
      </div>
      <div className='col-md-9'>
        <h1 className='text-center'>Manage Category</h1>
        <div className='p-3 w-50'>
          <CategoryForm handleSubmit={handleSubmit} value={name} setValue={setName}/>
        </div>
        <div className='w-100'>
          <table className="table">
            <thead>
              <tr>
                <th scope="col">Name</th>
                <th scope="col">Action</th>
              </tr>
            </thead>
            <tbody>
              {categories.map((citem)=>
                <>
                  <tr >
                    <td>
                      {citem.name}

```

```

        </td>
        <td>
            <button className='ms-md-2 btn btn-primary' onClick={()=>{setVisible(true) ; setUpdatedName(citem.name); setSelected(citem)}}>Edit</button>
            <button className='ms-md-2 ms-1 btn btn-danger' onClick={()=>{handleDelete(citem._id)}}>Delete</button>
        </td>
    </tr>
</tbody>
</table>

</div>
<Modal onCancel= {()=>setVisible(false)} footer={null} visible = {visible}>
<CategoryForm value={updatedName} setValue={setUpdatedName} handleSubmit={handleUpdate}/>
</Modal>
</div>
</div>
</div>
</Layout>
)
}

export default CreateCategory

```

Name	Action
women collection	<button>Edit</button> <button>Delete</button>
men collection	<button>Edit</button> <button>Delete</button>
kids collection	<button>Edit</button> <button>Delete</button>

Pages/Admin/CreateProduct.js

```

import axios from 'axios'
import React, {useEffect, useState} from 'react'
import { toast } from 'react-hot-toast'
import AdminMenu from '../../../../../component/Layout/AdminMenu'
import Layout from '../../../../../component/Layout/Layout'
import { useNavigate } from 'react-router-dom'
import{Select} from 'antd'
const {Option} = Select

const CreateProduct = () => {
  const navigate = useNavigate()
  const [categories, setCategories]= useState([])
  const[photo,setphoto]=useState("")
  const[name,setName]=useState("")
  const[description,setDescription]=useState("")
  const[price,setPrice]=useState("")
  const[category,setCategory]=useState("")
  const[quantity,setQuantity]=useState("")
  const[shipping,setShipping]=useState("")

```

```

//get all categories
const getAllCategories =async()=>{
  try {
    const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/category/get-category` )
    if(data?.success){
      setCategories(data?.category);
    }
  } catch (error) {
    console.log(error)
    toast.error('some thing went wrong in getting category')
  }
};

useEffect(()=>{
  getAllCategories()
},[])

//create product function
const handleCreate = async(e)=>{
e.preventDefault()
try {
  const productData = new FormData();
  productData.append("name",name)
  productData.append("description",description)
  productData.append("price",price)
  productData.append("quantity",quantity)
  productData.append("photo",photo)
  productData.append("category",category)
  const {data}= await axios.post(` ${process.env.REACT_APP_API}/api/v1/product/create-product` ,
  productData
)
  if(data?.success){
    toast.success('product created successfully')
    navigate('/dashboard/admin/products')
  }else{
    toast.error(data?.message)
  }
} catch (error) {
  console.log(error)
  toast.error("Something went wrong")
}
}

return (
<Layout title="Dashboard - Create Product">
  <div className='container mt-3 p-3 '>
    <div className='row'>
      <div className='col-md-3'>
        <AdminMenu/>
      </div>
      <div className='col-md-9 '>
        <h1 className='mt-4 text-center fw-bold'>Create Product</h1>
        <div className='m-1'>
          <Select bordered ={false} placeholder= "Select a category"
          size='large'
          showSearch className='from-select mb-3 w-75' onChange={(value)=>{setCategory(value)}}>
            {categories?.map(citem =>(
              <Option key={citem._id} value={citem._id}>{citem.name}</Option>
            )))
          </Select>
        <div className="mb-3 w-75">
          <label className='btn btn-outline-secondary'>
            {photo ? photo.name :"upload photo"}
            <input type="file" name='photo' accept='image/*' onChange={(e)=>setphoto(e.target.files[0])} hidden />
          </label>
        </div>
        <div className="mb-3">
          {photo && (
            <div className='text-center'>
              <img src={URL.createObjectURL(photo)} alt= "product photo" height={"200px"}>
              className="img img-responsive"/>
            </div>
          )}
        </div>
        <div className='mb-3'>
          <input type="text"
          value={name}
          placeholder="write a name"
          className='form-control'
          onChange={(e)=> setName(e.target.value)}
          />
        </div>
      </div>
    </div>
  </div>

```

```

        <div className='mb-3'>
          <textarea type="text"
            value={description}
            placeholder="write a description"
            className='form-control'
            onChange={(e)=> setDescription(e.target.value)}
          />
        </div>
        <div className='mb-3'>
          <input type="Number"
            value={price}
            placeholder="write a price"
            className='form-control'
            onChange={(e)=> setPrice(e.target.value)}
          />
        </div>
        <div className='mb-3'>
          <input type="Number"
            value={quantity}
            placeholder="write a quantity"
            className='form-control'
            onChange={(e)=> setQuantity(e.target.value)}
          />
        </div>
        <div className='mb-3'>
          <Select
            bordered={false}
            placeholder="Select shipping"
            size='large'
            className='form-select mb-3'
            onChange={(value)=>{setShipping(value)}}
          >
            <Option value = "0">No</Option>
            <Option value = "1">Yes</Option>
          </Select>
        </div>
        <div className='mb-3'>
          <button className='btn btn-primary' onClick={handleCreate}>CREATE PRODUCT</button>
        </div>
      </div>
    </div>
  </Layout>
)
}

export default CreateProduct

```

The screenshot shows a Microsoft Edge browser window with the URL <https://frail-outerwear-cod.cyclic.app/dashboard/admin/create-product>. The page title is "Create Product". On the left, there's a sidebar with "Admin Panel" and a menu with "Create Category", "Create Product" (which is highlighted in blue), "Products", "Users", and "Orders". The main content area has a heading "kids collection" and a file input field containing "img3.jpg". Below that is a preview image of a brown jacket. The product details form includes fields for "name" (jacket), "description" (pink kids jacket), "price" (3000), and "quantity" (1). The bottom of the screen shows the Windows taskbar with various pinned icons.

Pages/Admin/Product.js

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { toast } from 'react-hot-toast'
import AdminMenu from '../../../../../component/Layout/AdminMenu'
import Layout from '../../../../../component/Layout/Layout'
import {Link} from 'react-router-dom'
import "../../style/HomePage.css"

const Products = () => {
  const [products, setProduct] = useState([])

  //get all products

  const getAllProducts = async()=>{
    try {
      const {data} = await axios.get(`process.env.REACT_APP_API}/api/v1/product/get-product`)
      setProduct(data.products)
    } catch (error) {
      console.log(error)
      toast.error("something went wrong")
    }
  }

  //lifecycle method

  useEffect(()=>{
    getAllProducts()
  },[])

  return (
    <Layout>
      <div className="container">
        <div className='row '>
          <div className='col-lg-3'>
            <AdminMenu/>
          </div>
          <div className='col-lg-9 mt-3  home-page'>
            <h1 className='mx-auto'>All Products Lists</h1>
            <div className='d-flex flex-wrap mt-5 '>
              {products?.map((p)=>(
                <Link key={p._id} to={`/dashboard/admin/product/${p.slug}`}> className="product-Link">
                  <div className="card mx-auto ms-md-2 mb-4" >
                    <img src={ `${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}` } className="card-img-top" alt={p.name} />
                    <div className="card-body" >
                      <h5 className="card-title">{p.name}</h5>
                      <p className="card-text">{p.description}</p>
                </Link>
              ))
            </div>
          </div>
        </div>
      </Layout>
    )
}

```

```

        <p className="card-text">{p.price}</p>
    </div>
    </div>
    </div>
</Layout>
)
}

export default Products

```

The screenshot shows the Admin Panel of an E-commerce application. On the left, there is a sidebar with navigation links: Create Category, Create Product, Products (which is highlighted in blue), Users, and Orders. The main area is titled "All Products Lists" and displays three product items:

- Jacket**
pink kids jacket
3000
- Women Jeans**
women black printed trending jeans
8000
- Short Top**
women pink short tending top
500

The application is running on a local server at <https://frail-outerwear-cod.cyclic.app/dashboard/admin/products>. The browser interface includes a search bar, a navigation bar with links for HOME, CATEGORY, ADMIN, and CART, and a system tray at the bottom showing the date and time.

Pages/Admin/UpdateProduct

```

import axios from 'axios'
import React, {useEffect, useState} from 'react'
import { toast } from 'react-hot-toast'
import AdminMenu from '../../../../../component/Layout/AdminMenu'
import Layout from '../../../../../component/Layout/Layout'
import { useNavigate, useParams } from 'react-router-dom'
import { Select } from 'antd'
const { Option } = Select

const UpdateProduct = () => {
    const navigate = useNavigate()
    const params = useParams()
    const [categories, setCategories] = useState([])
    const [photo, setPhoto] = useState("")
    const [name, setName] = useState("")
    const [description, setDescription] = useState("")
    const [price, setPrice] = useState("")
    const [category, setCategory] = useState("")
    const [quantity, setQuantity] = useState("")
    const [shipping, setShipping] = useState("")
    const [id, setId] = useState("")

    //get single product
    const getSingleProduct = async () => {
        try {
            const { data } = await axios.get(`process.env.REACT_APP_API}/api/v1/product/get-product/${params.slug}`)
            setId(data.product._id)
            setName(data.product.name)
            setDescription(data.product.description)
            setPrice(data.product.price)
            setQuantity(data.product.quantity)
        } catch (error) {
            console.error(error)
        }
    }

    useEffect(() => {
        getSingleProduct()
    }, [])
}

```

```

        setCategory(data.product.category._id)
        setShipping(data.product.shipping)

    }catch(error){
        console.log(error)
        toast.error("something went wrong")
    }
}

useEffect(()=>{
    getSingleProduct();
    //eslint disable next line
},[])

//get all categories
const getAllCategories =async()=>{
    try {
        const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/category/get-category`)
        if(data?.success){
            setCategories(data?.category);
        }
    } catch (error) {
        console.log(error)
        toast.error('some thing went wrong in getting category')
    }
};

useEffect(()=>{
    getAllCategories()
},[])

//create product function
const handleUpdate = async(e)=>{
e.preventDefault()
try {
    const productData = new FormData();
    productData.append("name",name)
    productData.append("description",description)
    productData.append("price",price)
    productData.append("quantity",quantity)
    photo && productData.append("photo",photo)
    productData.append("category",category)
    const {data}= await axios.put(` ${process.env.REACT_APP_API}/api/v1/product/update-product/${id}` ,
    productData
)
    if(data?.success){
        toast.success('product updated successfully')
        navigate('/dashboard/admin/products')
    }else{
        toast.error(data?.message)
    }
} catch (error) {
    console.log(error)
    toast.error("Something went wrong")
}
}

//delete product

const handleDelete =async()=>{
try {
    let answer = window.prompt("Are you sure want to delete this product?")
    if(!answer) return ;
    const {data}= await axios.delete(` ${process.env.REACT_APP_API}/api/v1/product//delete-product/${id}` )
    toast.success('product deleted successfully')
    navigate('/dashboard/admin/products')
} catch (error) {
    console.log(error)
    toast.error("something went wrong")
}
}

return (
<Layout title="Dashboard - Update Product">
<div className='Container-fluid m-3 p-3 '>
<div className='row'>
<div className='col-md-3'>
<AdminMenu/>
</div>
<div className='col-md-9'>
<h1>Update Product</h1>
<div className='m-1 w-75'>
<Select bordered ={false} placeholder= "Select a category"

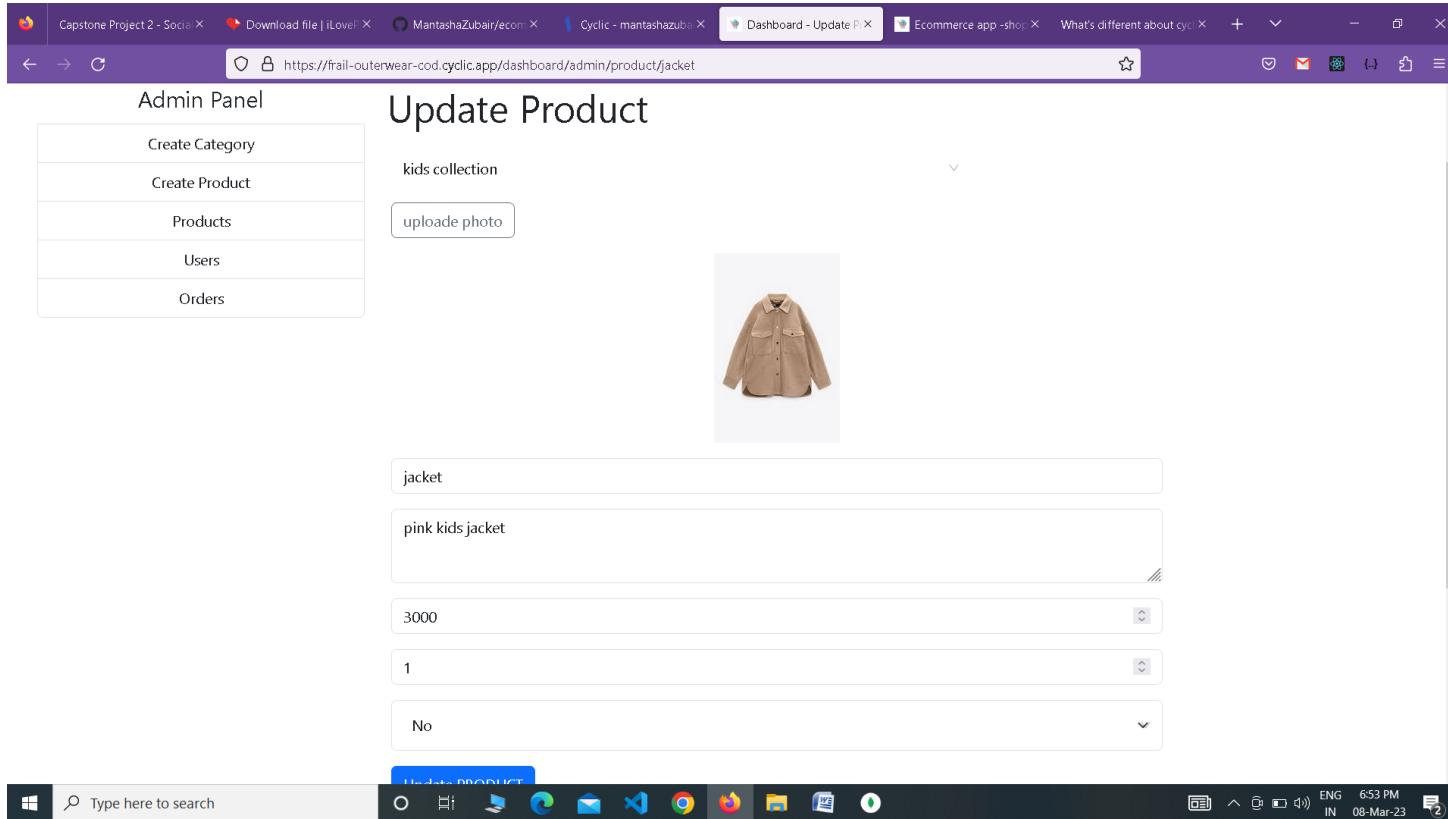
```

```

size='large'
showSearch className='from-select mb-3 w-75' onChange={(value)=>{setCategory(value)}} value={category}>
  {categories?.map(citem =>
    <Option key={citem._id} value={citem._id}>{citem.name}</Option>
  )}
</Select>
<div className="mb-3 w-75">
  <label className='btn btn-outline-secondary'>
    {photo ? photo.name : "upload photo"}
  <input type="file" name='photo' accept='image/*' onChange={(e)=>setphoto(e.target.files[0])} hidden />
  </label>
</div>
<div className="mb-3">
  {photo ?(
    <div className='text-center'>
      <img src={URL.createObjectURL(photo)} alt= "product photo" height={"200px"} className="img img-responsive"/>
    </div>
  ):(
    <div className='text-center'>
      <img src={'${process.env.REACT_APP_API}/api/v1/product/product-photo/${id}' } alt= "product photo" height={"200px"} className="img img-responsive"/>
    </div>
  )}
</div>
<div className='mb-3'>
  <input type="text"
  value={name}
  placeholder="write a name"
  className='form-control'
  onChange={(e)=> setName(e.target.value)}
  />
</div>
<div className='mb-3'>
  <textarea type="text"
  value={description}
  placeholder="write a description"
  className='form-control'
  onChange={(e)=> setDescription(e.target.value)}
  />
</div>
<div className='mb-3'>
  <input type="Number"
  value={price}
  placeholder="write a price"
  className='form-control'
  onChange={(e)=> setPrice(e.target.value)}
  />
</div>
<div className='mb-3'>
  <input type="Number"
  value={quantity}
  placeholder="write a quantity"
  className='form-control'
  onChange={(e)=> setQuantity(e.target.value)}
  />
</div>
<div className='mb-3'>
  <Select
    bordered={false}
    placeholder="Select shipping"
    size='large'
    className='form-select mb-3'
    onChange={(value)=>{setShipping(value)}}
    value={shipping ?"Yes" :"No"}
  >
    <Option value = "0">No</Option>
    <Option value = "1">Yes</Option>
  </Select>
</div>
<div className='mb-3'>
  <button className='btn btn-primary' onClick={handleUpdate}>Update PRODUCT</button>
</div>
<div className='mb-3'>
  <button className='btn btn-danger' onClick={handleDelete}>Delete PRODUCT</button>
</div>
</div>
</div>
</Layout>
)
}

```

```
export default UpdateProduct
```



The screenshot shows a web browser window with the URL <https://frail-outerwear-cod.cyclic.app/dashboard/admin/product/jacket>. The page title is "Update Product". On the left, there's a sidebar titled "Admin Panel" with links for Create Category, Create Product, Products, Users, and Orders. The main content area shows a product image of a pink kids jacket. Below the image are several input fields: a text input for "jacket", a dropdown for "pink kids jacket", a number input for "3000", a number input for "1", and a dropdown menu set to "No". At the bottom, there's a blue button labeled "UPDATE PRODUCT". The browser taskbar at the bottom shows various pinned sites like Capstone Project 2, MantashaZubair/ecom, Cyclic, Dashboard - Update P, Ecommerce app -shop, and What's different about cycli.

Pages/Admin/User.js

```
import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { toast } from 'react-hot-toast'
import AdminMenu from '../../../../../component/Layout/AdminMenu'
import Layout from '../../../../../component/Layout/Layout'

const User = () => {
  const [user, setUser] = useState([])

  const getAllUsers = async()=>{
    try {
      const {data} = await axios.get(`.${process.env.REACT_APP_API}/api/v1/auth/get-all-users`)
      setUser(data)
    } catch (error) {
      console.log(error)
      toast.error("something went wrong")
    }
  }

  useEffect(()=>{
    getAllUsers()
  },[])
  return (
    <Layout>
      <div className='Container-fluid m-3 p-3 '>
        <div className='row'>
          <div className='col-md-3'>
            <AdminMenu/>
          </div>
          <div className='col-md-9'>
            <h1 className='text-center mt-4 fw-bold'>All User</h1>

            <div className='border shadow table-responsive'>
              <table className='table'>
                <thead>
                  <tr>
                    <th scope="col">#</th>
                    <th scope="col">Name</th>
                    <th scope="col">Email</th>
                    <th scope="col">Phone</th>
                    <th scope="col">Address</th>
                  </tr>
                </thead>
                <tbody>
                  <tr>
                    <td>1</td>
                    <td>John Doe</td>
                    <td>johndoe@example.com</td>
                    <td>123-4567890</td>
                    <td>123 Main St, Anytown, USA</td>
                  </tr>
                  <tr>
                    <td>2</td>
                    <td>Jane Smith</td>
                    <td>janesmith@example.com</td>
                    <td>987-6543210</td>
                    <td>456 Elm St, Anytown, USA</td>
                  </tr>
                </tbody>
              </table>
            </div>
          </div>
        </div>
      </div>
    </Layout>
  )
}

export default User
```

```

        <th scope="col">Role</th>
      </tr>
    </thead>
    {
      user?.map((u,i)=>{
        return (
          <tbody>
            <tr key={u._id}>
              <td>i+1</td>
              <td>{u?.name}</td>
              <td>{u?.email}</td>
              <td>{u?.phone}</td>
              <td>{u?.address}</td>
              <td>{u?.role === 1 ? "Admin" : "user"}</td>
            </tr>
          </tbody>
        );
      })
    }
  </table>
</div>

</div>
</div>
</Layout>
)
}
}

export default User

```

#	Name	Email	Phone	Address	Role
1	test	test@gmail.com	678954321	mevdan	user
2	admin	admin@gmail.com	7897744166	admin	Admin
3	mantasha	mantasha@gmail.com	7897744166	mevdan	user
4	fazil	fazil@gmail.com	7897744166	handia	user

Admin Panel

Create Category
Create Product
Products
Users
Orders

All User

#	Name	Email	Phone	Address	Role
1	test	test@gmail.com	678954321	mevdan	user
2	admin	admin@gmail.com	7897744166	admin	Admin
3	mantasha	mantasha@gmail.com	7897744166	mevdan	user
4	fazil	fazil@gmail.com	7897744166	handia	user



Pages/Admin/AdminOrder.js

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import AdminMenu from '../../component/Layout/AdminMenu'
import { useAuth } from '../../context/auth'
import moment from 'moment'
import { Select } from 'antd'
import Layout from '../../component/Layout/Layout'
const { Option } = Select

const AdminOrders = () => {
  const [status, setStatus] = useState(["Not Process", "processing", "shipped", "deliver", "cancel"])
  const [changeStatus, setChangeStatus] = useState("")
  const [orders, setOrders] = useState([])

```

```

const [auth, setAuth] = useAuth()
const getOrders = async() =>{
  try {
    const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/auth/all-orders`)
    setOrders(data)
  } catch (error) {
    console.log(error)
  }
}
useEffect(()=>{
  if(auth?.token) getOrders()
},[auth?.token])

const handleChange =async(orderId, value)=>{
try {
  const {data} = await axios.put(` ${process.env.REACT_APP_API}/api/v1/auth/order-status/${orderId}` ,
  {status:value})
  getOrders();
} catch (error) {
  console.log(error)
}
}
return (
<Layout title="Dashboard - AdminOrders">
<div className='container' >
  <div className='row mt-4' >
    <div className='col-md-3 p-3' >
      <AdminMenu/>
    </div>
    <div className='col-md-9' >
      <h1 className='text-center mt-4 fw-bold'>All Orders</h1>
      {
        orders?.map((o,i)=>{
          return (
            <div className='border shadow table-responsive' >
              <table className='table' >
                <thead>
                  <tr>
                    <th scope="col">#</th>
                    <th scope="col">Status</th>
                    <th scope="col">Buyer</th>
                    <th scope="col">date</th>
                    <th scope="col">Payment</th>
                    <th scope="col">Quantity</th>
                  </tr>
                </thead>
                <tbody>
                  <tr key={o?._id}>
                    <td>{i+1}</td>
                    <td>
                      <Select bordered={false} onChange={(value) => handleChange(o._id,value)} defaultValue={o?.status}>
                        {status.map((s,i)=>(
                          <Option key={i} value={s}>{s}</Option>
                        )))
                      </Select>
                    </td>
                    <td>{o?.buyer?.name}</td>
                    <td>{moment(o?.createdAt).fromNow()}</td>
                    <td>{o?.payment.success ? "Success" : "Failed"}</td>
                    <td>{o?.products?.length}</td>
                  </tr>
                </tbody>
              </table>
            <div className='container' >
              {o?.products?.map((p)=>(
                <div className='row mb-2 p-3 card flex-row shadow' >
                  <div className='col-md-4' key={p._id}>
                    <img src={` ${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}` } className="card-img-top" alt={p.name}
                      width={"100px"}
                      height={"200px"}
                      />
                  </div>
                  <div className='col-md-8' >
                    <h4>{p.name}</h4>
                    <p>{p.description.substring(0, 30)}</p>
                    <h4>Price: {p.price}</h4>
                  </div>
                </div>
              </div>
            )))
      }

```

```

        </div>
      );
    }
  </div>
</div>
</Layout>
)
}

export default AdminOrders

```

The screenshot shows a web browser window with the URL <https://trial-outerwear-cod.cyclic.app/dashboard/admin/orders>. The page title is "Ecommerce APP". The main content area is titled "All Orders". A table lists one order:

#	Status	Buyer	Date	Payment	Quantity
1	shipped	fazil	a day ago	Success	1

Below the table, there is a placeholder image of a man wearing a purple shirt and black pants, with the text "shirt" and "men shirt" next to it, and "Price: 5000" below.

Component/UserMenu.js

```

import React from 'react'
import { NavLink } from 'react-router-dom'

const UserMenu = () => {
  return (
    <>
      <div className='text-center'>
        <div className="list-group">
          <h4>User Panel</h4>
          <NavLink to="/dashboard/user/profile" className="list-group-item list-group-item-action ">Profile</NavLink>
          <NavLink to="/dashboard/user/orders" className="list-group-item list-group-item-action">Orders</NavLink>
        </div>
      </div>
    </>
  )
}

export default UserMenu

```

pages/user/UserDashboard.js

```

import React from 'react'
import Layout from '../../../../../component/Layout/Layout'
import UserMenu from '../../../../../component/Layout/UserMenu'
import { useAuth } from '../../../../../context/auth'

const DashBoard = () => {
  const [auth] = useAuth()
  return (
    <Layout title="DashBoard - Ecommerce App">

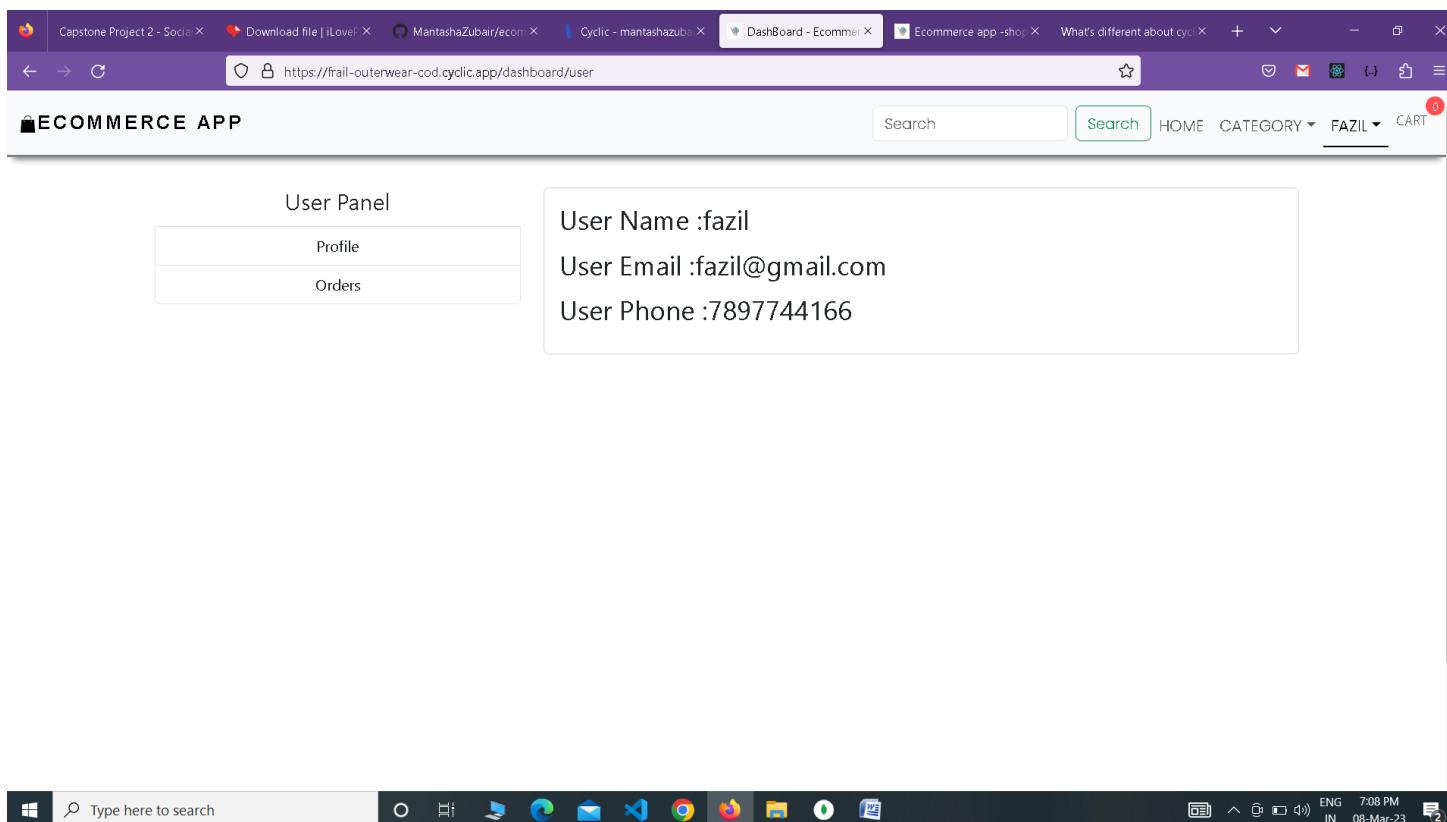
```

```

<div className='container '>
  <div className='row m-3 p-3'>
    <div className='col-md-4 mb-4'>
      <UserMenu/>
    </div>
    <div className='col-md-8 '>
      <div className='card p-3 '>
        <h3>User Name :{auth?.user?.name}</h3>
        <h3>User Email :{auth?.user?.email}</h3>
        <h3>User Phone :{auth?.user?.phone}</h3>
      </div>
    </div>
  </div>
</Layout>
)
}

export default DashBoard

```



Pages/user/Profile.js

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { toast } from 'react-hot-toast'
import Layout from '../../../../../component/Layout/Layout'
import UserMenu from '../../../../../component/Layout/UserMenu'
import { useAuth } from '../../../../../context/auth'

const Profile = () => {
  //context
  const [ auth, setAuth ]= useAuth()
  //state
  const [ name, setName ] = useState("");
  const [ email, setEmail ] = useState("");
  const [ password, setPassword ] = useState("");
  const [ phone, setPhone ] = useState("");
  const [ address, setAddress ] = useState("");

  //get user data
  useEffect(()=>{
    const {name,email,phone,address} = auth?.user
    setName(name)
    setEmail(email)
    setPhone(phone)
    setAddress(address)
  })
}

```

```

}, [auth?.user])

///Form submit handler
const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    const {data} = await axios.put(
      `${process.env.REACT_APP_API}/api/v1/auth/profile`,
      { name, email, password, phone, address }
    );
    if (data?.error) {
      toast.error(data?.error);
    } else {
      setAuth({...auth, user:data?.updatedUser})
      let ls = localStorage.getItem("auth")
      ls = JSON.parse(ls)
      ls.user= data.updatedUser
      localStorage.setItem("auth",JSON.stringify(ls))
      toast.success("Profile updated successfully")
    }
  } catch (error) {
    console.log(error);
    toast.error("Something Went wrong");
  }
};

return (
  <Layout>
  <div className='container'>
    <div className='row mt-2'>
      <div className='col-md-3 p-3'>
        <UserMenu/>
      </div>
      <div className='col-md-9'>
        <div className="form-container">
          <form onSubmit={handleSubmit}>
            <h4 className="title">USER PROFILE</h4>
            <div className="mb-3">
              <label htmlFor="name" className="form-label">
                Name
              </label>
              <input
                type="text"
                name="name"
                className="form-control"
                value={name}
                onChange={(e) => setName(e.target.value)}
                />
            </div>
            <div className="mb-3">
              <label htmlFor="exampleInputEmail1" className="form-label">
                Email address
              </label>
              <input
                type="email"
                name="email"
                className="form-control"
                value={email}
                onChange={(e) => setEmail(e.target.value)}
                disabled
                />
            </div>
            <div className="mb-3">
              <label htmlFor="exampleInputPassword1" className="form-label">
                Password
              </label>
              <input
                type="password"
                name="password"
                className="form-control"
                value={password}
                onChange={(e) => setPassword(e.target.value)}
                />
            </div>
            <div className="mb-3">
              <label htmlFor="phone" className="form-label">
                Phone
              </label>
              <input

```

```

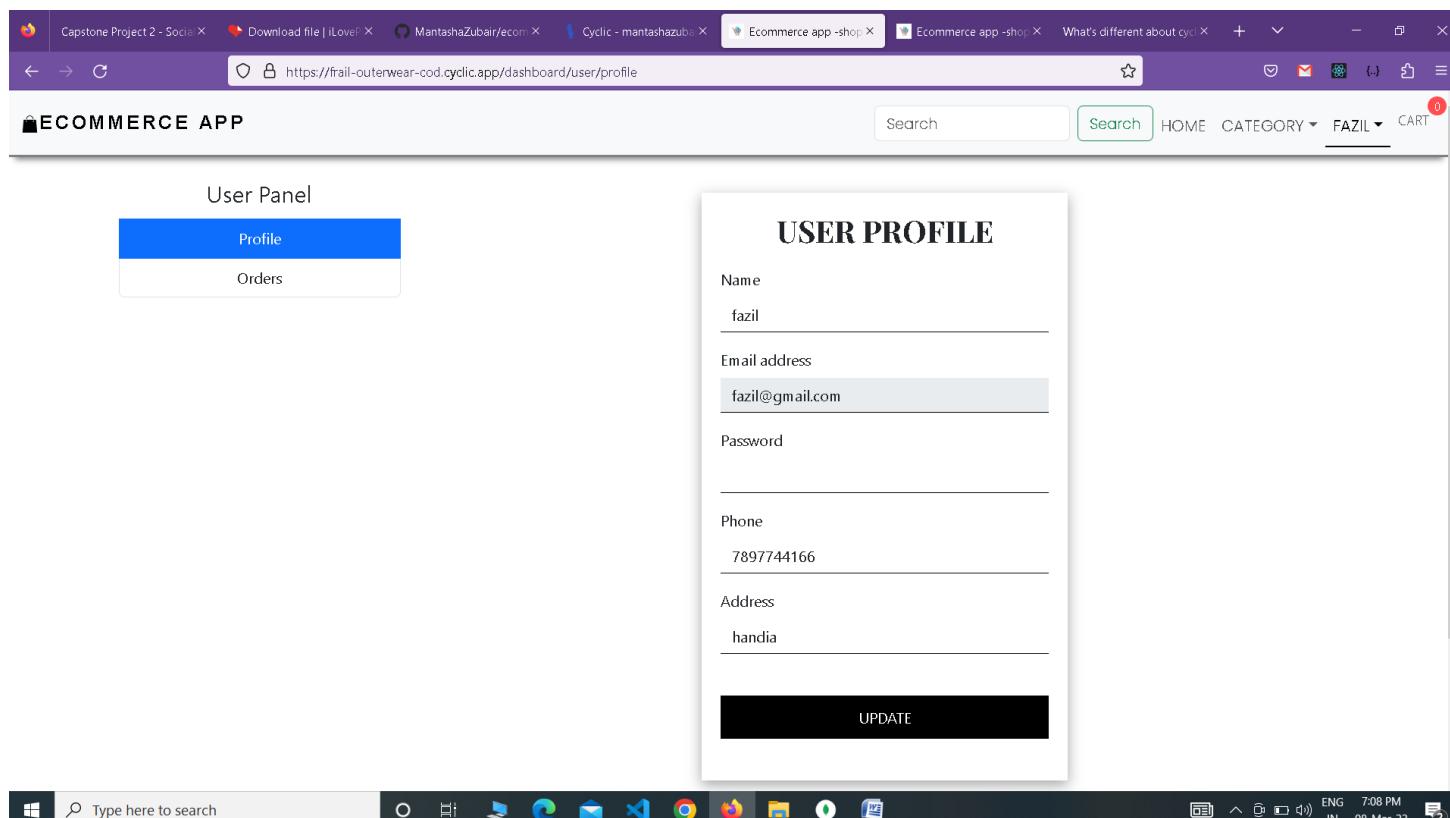
        type="text"
        name="phone"
        className="form-control"
        value={phone}
        onChange={(e) => setPhone(e.target.value)}


    />
</div>
<div className="mb-3">
    <label htmlFor="address" className="form-label">
        Address
    </label>
    <input
        type="text"
        name="address"
        className="form-control"
        value={address}
        onChange={(e) => setAddress(e.target.value)}
    />
</div>

<div className="mb-4 mt-4 d-grid">
    <button type="submit" className="btn btn-primary">
        UPDATE
    </button>
</div>
</form>
</div>
</div>
</div>
</div>
</Layout>
)
}

export default Profile

```



Pages/user/Order.js

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import Layout from '../../../../../component/Layout/Layout'
import UserMenu from '../../../../../component/Layout/UserMenu'
import { useAuth } from '../../../../../context/auth'
import moment from 'moment'

```

```

const Orders = () => {
  const [orders, setOrders] = useState([])
  const [auth, setAuth] = useAuth()
  const getOrders = async() =>{
    try {
      const {data} = await axios.get(`[${process.env.REACT_APP_API}]/api/v1/auth/orders`)
      setOrders(data)
    } catch (error) {
      console.log(error)
    }
  }
  useEffect(()=>{
    if(auth?.token) getOrders()
  },[auth?.token])

  return (
    <Layout title="Ecommerce - user-Orders">
      <div className='container'>
        <div className='row mt-4'>
          <div className='col-md-3 p-3'>
            <UserMenu/>
          </div>
          <div className='col-md-9'>
            <h1 className='fw-bold text-center mt-4'>All Orders</h1>
            {
              orders?.map((o,i)=>{
                return (
                  <div className='border shadow table-responsive'>
                    <table className='table'>
                      <thead>
                        <tr>
                          <th scope="col">#</th>
                          <th scope="col">Status</th>
                          <th scope="col">Buyer</th>
                          <th scope="col">date</th>
                          <th scope="col">Payment</th>
                          <th scope="col">Quantity</th>
                        </tr>
                      </thead>
                      <tbody>
                        <tr key={o._id}>
                          <td>{i+1}</td>
                          <td>{o?.status}</td>
                          <td>{o?.buyer?.name}</td>
                          <td>{moment(o?.createdAt).fromNow()}</td>
                          <td>{o?.payment.success ? "Success" : "Failed"}</td>
                          <td>{o?.products?.length}</td>
                        </tr>
                      </tbody>
                    </table>
                  <div className='container'>
                    {o?.products?.map((p)=>
                      <div className='row mb-2 mx-2 card flex-row shadow'>
                        <div className='col-md-4 ' key={p._id}>
                          <img src={`${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}`}>
                            className="card-img-top"
                            alt={p.name}

                            width={"100px"}
                            height={"200px"}
                          />
                        </div>
                        <div className='col-md-8'>
                          <h4>{p.name}</h4>
                          <p>{p.description.substring(0, 30)}</p>
                          <h4>Price: {p.price}</h4>

                        </div>
                      </div>
                    )));
                  </div>
                )
              )
            )
          </div>
        </div>
      </Layout>
    )
  }

  export default Orders
}

```

User Panel

All Orders

#	Status	Buyer	Date	Payment	Quantity
1	shipped	fazil	a day ago	Success	1

shirt
men shirt
Price: 5000

Pages/CartPage.js=> in cart page use contextApi for created globally cart item and for payment gatway use Braintree-web-drop-in-react for gatting dynamic cart ui in backend api use Braintree also conditionaly when user login and also show user address then payment is processing if user not login show makepayment

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom'
import Layout from '../component/Layout/Layout'
import { useAuth } from '../context/auth'
import { useCart } from '../context/cart'
import DropIn from "braintree-web-drop-in-react";
import { toast } from 'react-hot-toast'

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

  //total price
  const totalPrice = () => {
    try {
      let total = 0;
      cart?.map((item) => (total = total + item.price * item.quantity))
      return total.toLocaleString("en-US", {
        style: "currency",
        currency: "INR",
      })
    } catch (error) {
      console.log(error)
    }
  }

  //delete item
  const removeCartItem = (pid) => {
    try {
      let myCart = [...cart]
      let index = myCart.findIndex(item => item._id === pid)
    }
  }
}


```

```

myCart.splice(index,1)
setCart(myCart)
localStorage.setItem('cart',JSON.stringify(myCart))
} catch (error) {
  console.log(error)
}
}

//get payment Token
const getToken = async()=>{
  try {
    const {data}=await axios.get(`[${process.env.REACT_APP_API}]/api/v1/product/braintree/token`)
    setClientToken(data?.clientToken)
  } catch (error) {
    console.log(error)
  }
}
useEffect(()=>{
  getToken()
},[auth?.token])

//handle payment

const handlePayment = async()=>{
  try {
    setLoading(true)
    const {nonce} = await instance.requestPaymentMethod()
    const {data} = await axios.post(`[${process.env.REACT_APP_API}]/api/v1/product/braintree/payment`,{cart,nonce})
    setLoading(false)
    localStorage.removeItem('cart')
    setCart([])
    navigate('/dashboard/user/orders')
    toast.success('Payment completed sucessfully')

  } catch (error) {
    console.log(error)
    setLoading(false)
  }
}

return (
<Layout>
  <div className='container'>
    <div className='row'>
      <div className='col-md-12'>
        <h1 className='text-center bg-light p-2 mb-1'>
          {!auth?.user ? "welcome user" : `welcome ${auth?.token} && auth?.user?.name`}
        </h1>
        <p className='text-center'>
          {cart?.length ? `You Have ${cart.length} items in your cart` : "your cart is empty"}
          ${auth?.token ? "" : "please login to checked out"}
        </p>
      </div>
    </div>
    <div className='container'>
      <div className='row'>
        <div className='col-md-8 col-10 mx-auto'>
          {cart?.map((p)=>
            <div className='row mb-2 p-3 card flex-row shadow'>
              <div className='col-md-4' key={p._id}>
                <img src='[${process.env.REACT_APP_API}]/api/v1/product/product-photo/${p._id}'>
                className="card-img-top"
                alt={p.name}
                width={"100px"}
                height={"200px"/>
              </div>
              <div className='col-md-4'>
                <h4>{p.name}</h4>
                <p>{p.description.substring(0, 30)}</p>
                <h4>Price: {p.price}</h4>
              </div>
              <div className="col-md-4">
                <button className='btn btn-danger' onClick={()=>removeCartItem(p._id)}>Remove</button>
              </div>
            </div>
          ))
        </div>
        <div className='col-md-4 text-center'>
          <h4>Cart Summary</h4>
        </div>
      </div>
    </div>
  </div>
)

```

```

<p>Total | Checkout | Payment</p>
<hr/>
<h4>Total : {totalPrice()}</h4>
{auth?.user?.address ?
(
    <>
        <div className='mb-3'>
            <h4>Current Address</h4>
            <h5>{auth?.user?.address}</h5>
            <button className='btn btn-outline-warning' onClick={()=>navigate('/dashboard/user/profile')}>Update
Address</button>
        </div>
    </>
):(
    <div className='mb-3'>
        {auth?.token ? (
            <button className='btn btn-outline-warning' onClick={()=>navigate('/dashboard/user/profile')}>Update
Address</button>
        ) : (
            <button className='btn btn-outline-warning' onClick={()=>navigate('/login',{state:"/cart"})}>Please
Login to checkout</button>
        )
    )
)
}

<div className="mt-2">
{
    !clientToken || !cart?.length ? ("") : (
        <>
            <DropIn
                options={{
                    authorization:clientToken,
                    paypal:{
                        flow:'vault',
                    }
                }}
                onInstance={(instance) => setInstance(instance)}
            />
            <button className='btn btn-primary' onClick={handlePayment}
disabled={loading|| !instance ||!auth?.user?.address}>
{loading ? "Processing...." :"MakePayment"}</button>
        </>
    )
}

</div>
</div>
</div>
</div>
</Layout>
)

}

export default CartPage

```

welcome fazil
You Have 2 items in your cart

gray pant men gray pant Price: 1000	<button>Remove</button>
white shirt men casual white shirt Price: 4000	<button>Remove</button>

Cart Summary
Total | Checkout | Payment

Total : ₹5,000.00
Current Address
handia
[Update Address](#)

Pay with card
VISA | MasterCard | American Express | JCB | DISCOVER

Card Number
Expiration Date (MM/YY) CVV (3 digits)
MM/YY ...

Context/cart.js

```
import { useState, useContext, createContext, useEffect } from "react";

const CartContext = createContext()

const CartProvider = ({children})=>{
  const [cart, setCart] = useState([])

  useEffect(()=>{
    let existingCartItem= localStorage.getItem('cart')
    if(existingCartItem) setCart(JSON.parse(existingCartItem))
  },[])

  return (
    <CartContext.Provider value={[cart, setCart]}>
      {children}
    </CartContext.Provider>
  )
}

//custom hook

const useCart = ()=> useContext(CartContext)
export {useCart, CartProvider}
```

component/form/SearchInput.js

```
import axios from 'axios'
import React from 'react'
import { useNavigate } from 'react-router-dom'
import { useSearch } from '../../context/search'

const SearchInput= ()=>{
  const [values, setValues] = useSearch()
  const navigate = useNavigate()

  const handleSubmit = async(e)=>{
    e.preventDefault()
    try {

```

```

        const {data}= await axios.get(`${process.env.REACT_APP_API}/api/v1/product/search/${values.keyword}`)
        setValues({...values,results:data})
        navigate("/search")
    } catch (error) {
        console.log(error)
    }
}

return (
<div>
<form className="d-flex" onSubmit={handleSubmit}>
<input className="form-control me-2"
type="search"
placeholder="Search"
aria-label="Search"
value={values.keyword}
onChange= {{(e) => setValues({...values, keyword: e.target.value})}}
/>
<button className="btn btn-outline-success" type="submit">Search</button>
</form>
</div>
)
}

export default SearchInput

```

context/search.js

```

import { useState,useContext,createContext } from "react";

const SearchContext =createContext()

const SearchProvider =({children})=>{
    const [auth,setAuth] = useState({
        keyword:"",
        results:[]
    })

    return (
        <SearchContext.Provider value={[auth,setAuth]}>
            {children}
        </SearchContext.Provider>
    )
}

//custom hook

const useSearch =()=> useContext(SearchContext)
export {useSearch,SearchProvider}

```

pages/search.js

```

import React from 'react'
import Layout from '../component/Layout/Layout'
import { useSearch } from '../context/search'

const Search = () => {
    const [values,setValue]=useSearch()
    return (
        <Layout title = "Search result">
        <div className='container'>
            <div className='text-center'>
                <h1>Search Result</h1>
                <h6>{values?.results.length<1 ?"No Result Found":`Found ${values?.results.length}`}</h6>
                <div className='d-flex flex-wrap mt-4'>
                    {values?.results.map((p)=>
                        <div className="card m-2" style={{width: '18rem'}} key={p._id}>
                            <img src={`${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}`} className="card-img-top" alt={p.name} />
                            <div className="card-body">

```

```

        <h5 className="card-title">{p.name}</h5>
        <p className="card-text">{p.description.substring(0,30)}</p>
        <p className="card-text">{p.price}</p>
        <button className="btn btn-primary ms-1">More Details</button>
        <button className="btn btn-secondary ms-1">Add to cart</button>
    </div>
</div>

))}
</div>
</div>
</div>
</Layout>
)
}

export default Search

```

hook/useCategory.js

```

import axios from "axios";
import { useEffect, useState } from "react";

export default function useCategory() {
    const [categories, setCategories] = useState([])

    //get category

    const getCategories = async ()=>{
        try {
            const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/category/get-category`);
            setCategories(data?.category);
        } catch (error) {
            console.log(error)
        }
    }
    useEffect(()=>{
        getCategories();
    },[])
    return categories;
}

```

pages/Categories.js

```

import React from 'react'
import { Link } from 'react-router-dom'
import Layout from '../component/Layout/Layout'
import useCategory from '../hooks/useCategory'

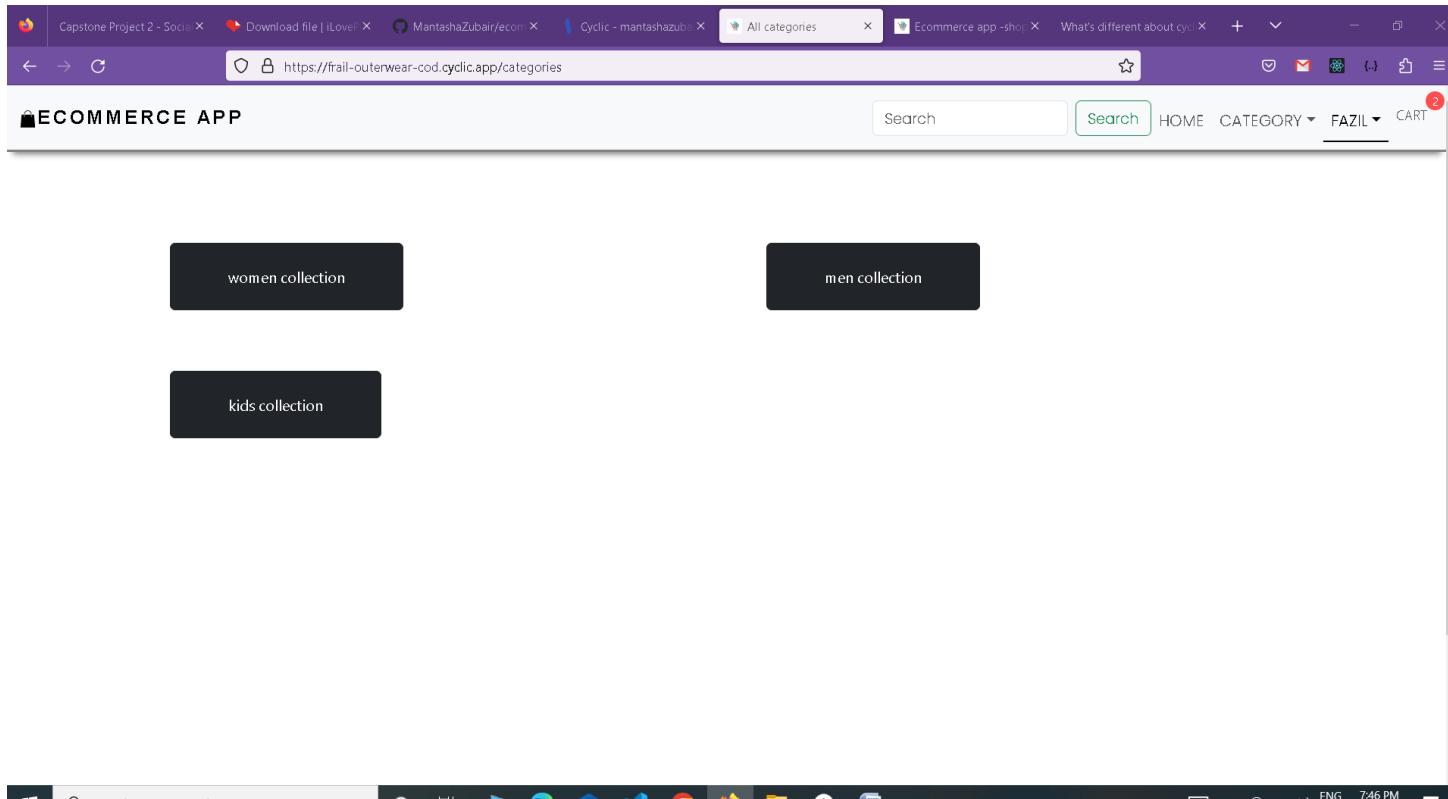
const Categories = () => {
    const categories = useCategory()

    return (
        <Layout title= "All categories ">
            <div className='container mt-5'>
                <div className='row ms-5'>

                    {categories.map((citem)=>(
                        <div className='col-md-6 mt-5 mb-3' key={citem._id}>
                            <button className='btn btn-dark text-light px-5 py-3 d-flex align-item-center justify-content-center ' >
                                <Link to={`/category/${citem.slug}`} className='btn btn-dark'>{citem.name}</Link>
                            </button>
                        </div>
                    )))
                </div>
            </div>
        </Layout>
    )
}

export default Categories

```



pages/CategoryProduct.js

```
import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { toast } from 'react-hot-toast'
import { useNavigate, useParams } from 'react-router-dom'
import Layout from '../component/Layout/Layout'
import { useCart } from '../context/cart';
import "../style/CategoryProduct.css";

const CategoryProduct = () => {
    const navigate = useNavigate()
    const [cart, setCart] = useCart()
    const params = useParams()
    const [products, setProducts] = useState([])
    const [category, setCategory] = useState([])

    const getProductByCategory = async()=>{
        try {
            const {data} = await axios.get(` ${process.env.REACT_APP_API}/api/v1/product//product-category/${params.slug}`)
            setProducts(data?.products)
            setCategory(data?.category)
        } catch (error) {
            console.log(error)
        }
    }
    useEffect(()=>{
        if(params?.slug) getProductByCategory()
    },[params?.slug])
    return (
        <Layout>
            <div className='container mt-3 '>
                <h4 className='text-center'>Category - {category?.name}</h4>
                <h6 className='text-center'>{products?.length} result found</h6>
                <div className='row cartegory-product-details'>
                    <div className='d-flex flex-wrap '>
                        {products?.map((p)=>(
                            <div className="card mx-auto ms-md-2 mb-4" >
                                <img src={` ${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}`}
                                    className="card-img-top"
                                    alt={p.name}
                                    key={p._id}
                            </div>
                        ))
                    )
                </div>
            </div>
        </Layout>
    )
}

export default CategoryProduct
```

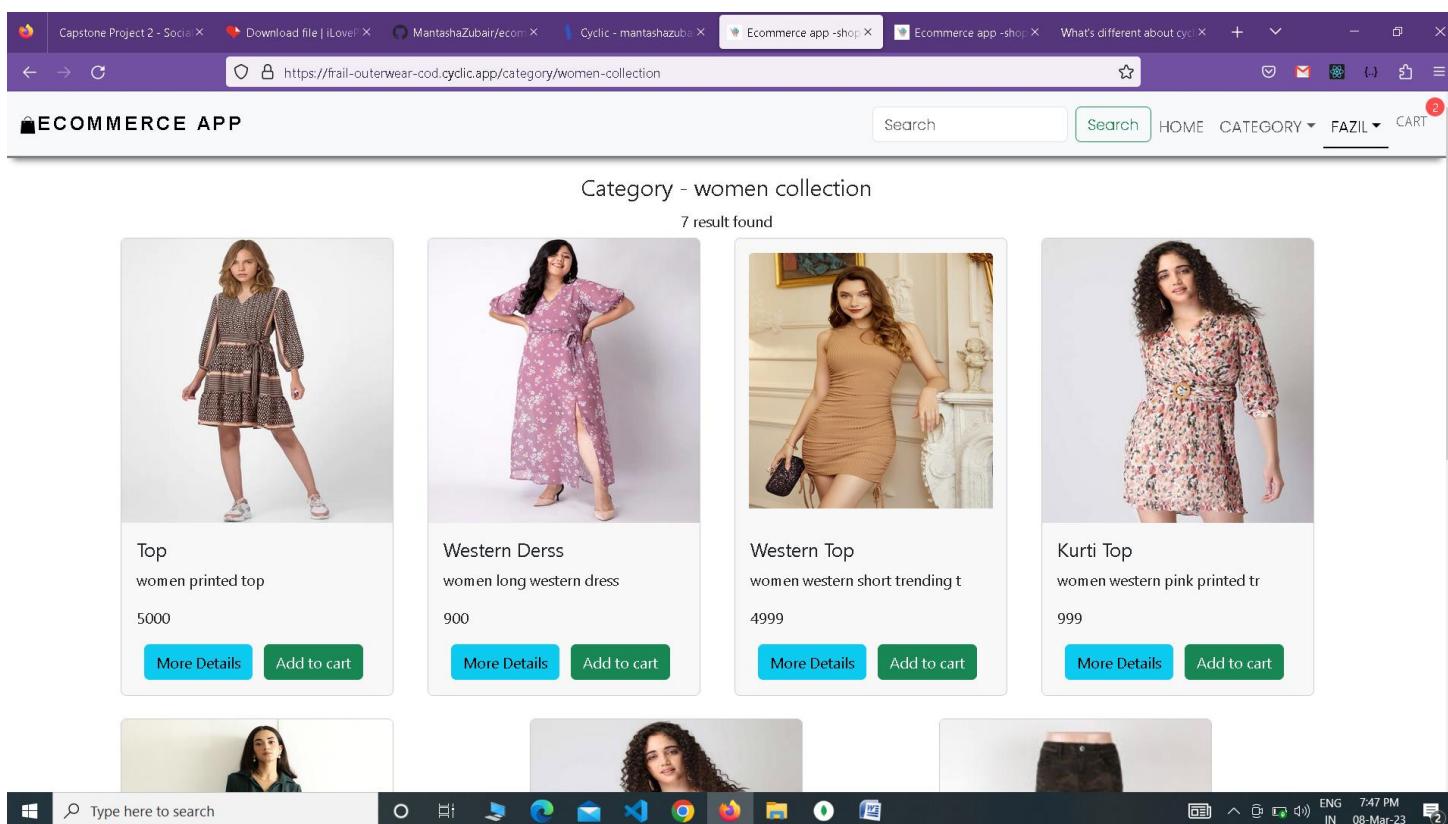
```

        >/>

      <div className="card-body">
        <h5 className="card-title">{p.name}</h5>
        <p className="card-text">{p.description.substring(0,30)}</p>
        <p className="card-text">{p.price}</p>
        <button className="btn btn-info ms-2 me-1" onClick={()=>navigate(`/product/${p.slug}`)}>More Details</button>
        <button className="btn btn-success ms-2 me-1" onClick={()=>{
          setCart([...cart,p])
          localStorage.setItem('cart', JSON.stringify([...cart,p]))
          toast.success('item added to cart')
        }}>Add to cart</button>
      </div>
    </div>
  )
}

export default CategoryProduct

```



Pages/HomePage.js => in home page use category wise filter and price wise filter product Loading functionality is also include

```

import React, { useEffect, useState } from 'react'
import Layout from '../../component/Layout/Layout'
import axios from 'axios'
import { toast } from 'react-hot-toast';
import { Checkbox, Radio } from 'antd'
import { Prices } from '../../component/Prices';
import { useNavigate } from 'react-router-dom';
import { useCart } from '../../context/cart';
import banner from '../../images/banner.jpg'

```

```

import "../style/HomePage.css"

const HomePage = () => {
  const [cart, setCart] = useState()
  const [products, setProducts] = useState([])
  const [categories, setCategories] = useState([])
  const [checked, setChecked] = useState([])
  const [radio, setRadio] = useState([])
  const [total, setTotal] = useState(0)
  const [page, setPage] = useState(1)
  const [loading, setLoading] = useState(false)
  const navigate = useNavigate()

  //get all categories
  const getAllCategories = async()=>{
    try {
      const {data} = await axios.get(`process.env.REACT_APP_API}/api/v1/category/get-category`)
      if(data?.success){
        setCategories(data?.category);
      }
    } catch (error) {
      console.log(error)
    }
  };

  useEffect(()=>{
    getAllCategories()
    getTotal()
  },[[]])

  //get product
  const getAllProducts = async()=>{
    try {
      setLoading(true)
      const {data}= await axios.get(`process.env.REACT_APP_API}/api/v1/product/product-list/${page}`)
      setLoading(false)
      setProducts(data.products);
    } catch (error) {
      console.log(error)
      setLoading(false)
      toast.error("Something went wrong")
    }
  }

  //get total count
  const getTotal = async() => {
    try {
      const {data } = await axios.get(`process.env.REACT_APP_API}/api/v1/product/product-count`)
      setTotal(data?.total)
    } catch (error) {
      console.log(error)
    }
  }

  useEffect(()=>{
    if(page==1) return;
    loadMore()
  },[page])

  //loadMore
  const loadMore = async()=>{
    try {
      setLoading(true)
      const {data} = await axios.get(`process.env.REACT_APP_API}/api/v1/product/product-list/${page}`)
      setLoading(false)
      setProducts([...products,...data?.products])
    } catch (error) {
      console.log(error)
      setLoading(false)
    }
  }
}

```

```

//filter by category
const handleFilter = (value,id)=>{
  let all = [...checked]
  if(value){
    all.push(id)
  }else{
    all = all.filter(c => c!==id)
  }
  setChecked(all)
}
useEffect(()=>{
  if(!checked.length||!radio.length)getAllProducts()
  ,[checked.length,radio.length])

useEffect(()=>{
  if(checked.length || radio.length)filterProduct()
  ,[checked,radio])

//get filter product
const filterProduct = async()=>{
  try {
    const {data}= await axios.post(`[${process.env.REACT_APP_API}]/api/v1/product/product-filter`,{checked,radio})
    setProducts(data?.products)
  } catch (error) {
    console.log(error)
  }
}
return (
<Layout title = "All product -Best Offers">
  <img src = {banner}
  className="banner-img img-fluid"
  alt={"banner"}
  width={"100%"}
  height={"100%"}
/>
<div className='container'>
  <div className='row mt-3 home-page'>
    <div className='col-lg-3 filters'>
      <h4 className='text-center'>Filter By Category</h4>
      <div className='d-flex flex-column'>
        {categories?.map((citem)=>(
          <Checkbox key={citem._id} onChange={((e)=>handleFilter(e.target.checked, citem._id))}>
            {citem.name}
          </Checkbox>
        )))
      </div>

      {/* price filter */}
      <h4 className='text-center'>Filter By Price</h4>
      <div className='d-flex flex-column'>
        <Radio.Group onChange={((e)=>setRadio(e.target.value))}>
          {Prices?.map((p)=>(
            <div>
              <Radio key={p._id} value={p.array}>{p.name}</Radio>
            </div>
          )))
        </Radio.Group>
      </div>

      <div className='d-flex flex-column'>
        <button className="btn btn-dark mt-5" onClick={()=>window.location.reload()}>RESET FILTER</button>
      </div>

    </div>
    <div className='col-lg-9'>
      <h1 className='text-center'>All Products</h1>
      <div className='d-flex flex-wrap'>
        {products?.map((p)=>(
          <div className="card mx-auto ms-md-2 mb-4" >
            <img src={`${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}`}>
            <div className="card-img-top" alt={p.name}>
              key={p._id}
            </div>
            <div className="card-body">

```

```

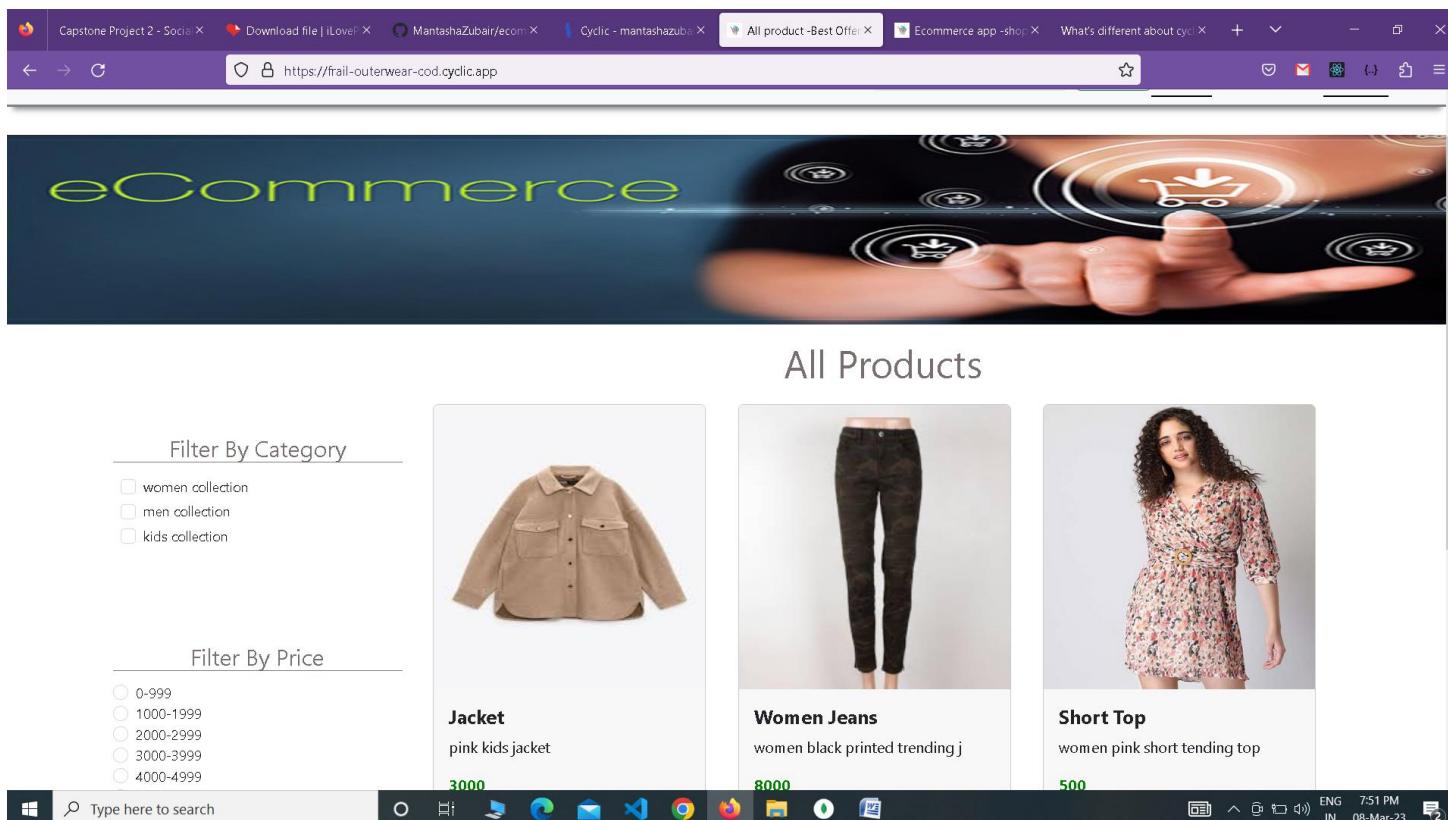
        <h5 className="card-title fw-bold">{p.name}</h5>
        <p className="card-text">{p.description.substring(0,30)}</p>
        <p className="card-text-price fw-bold">{p.price}</p>
        <button className="btn btn-primary ms-2" onClick={()=>navigate(`/product/${p.slug}`)}>More Details</button>
        <button className="btn btn-success ms-3" onClick={()=>{
            setCart([...cart,p])
            localStorage.setItem('cart', JSON.stringify([...cart,p]))
            toast.success("item added to cart")
        }}>
            Add to cart
        </button>
    </div>
</div>

))}

</div>
<div className='m-2 p-3'>
    {products && products.length < total &&(
        <button className='btn btn-warning' onClick={(e)=>{
            e.preventDefault()
            setPage(page+1);
        }}>
            {loading ? "Loading" :"Loadmore"}
        </button>
    )}
    </div>
</div>
</div>
</div>
</Layout>
)
}

export default HomePage;

```



Pages/ProductDetails.js => when click more details button in our cart page then it redirect to the productDetails page where see product details and also similer product

```

import axios from 'axios'
import React, { useEffect, useState } from 'react'
import { toast } from 'react-hot-toast'
import { useNavigate, useParams } from 'react-router-dom'
import Layout from '../component/Layout/Layout'
import { useCart } from '../context/cart'
import "../style/ProductDetails.css"

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

  //initial details
  useEffect(()=>{
    if(params?.slug) getProduct()
  },[params?.slug])

  //get Product
  const getProduct = async()=>{
    try {
      const {data}= await axios.get(`process.env.REACT_APP_API}/api/v1/product/get-product/${params.slug}`);
      setProduct(data?.product)
      getSimilarProduct(data?.product._id, data?.product.category._id)
    } catch (error) {
      console.log(error)
    }
  }

  //get similer products
  const getSimilarProduct = async(pid,cid)=>{
    try {
      const {data}= await axios.get(`process.env.REACT_APP_API}/api/v1/product/related-product/${pid}/${cid}`);
      setRelatedProducts(data?.products)
    } catch (error) {
      console.log(error)
    }
  }

  return (
    <Layout>
    <div className='container product-details'>
      <div className='row m-4 m-md-2'>
        <div className='col-md-4 mx-auto'>
          <img src={`process.env.REACT_APP_API}/api/v1/product/product-photo/${product._id}`}
            className="card-img-top"
            alt={product.name}
            height="300px"
            width={"170px"}
          />
        </div>
        <div className='col-md-6 product-details-info'>
          <h1 className='text-center mt-4 fw-bold'> Product Details</h1>
          <h6>Name :{product.name}</h6>
          <h6>Description :{product.description}</h6>
          <h6>Price :{product.price}</h6>
          <h6>Category :{product.category?.name}</h6>
          <button className="btn btn-secondary ms-1" onClick={()=>{
            setCart([...cart,product])
            localStorage.setItem('cart',JSON.stringify([...cart,product]))
            toast.success("item added to cart")
          }}>Add to cart</button>
        </div>
      </div>
      <hr/>

      <div className='row similer-product'>
        <h3>Similer Product</h3>
        {relatedProducts.length<1 && <p>No Similer Product Found</p>}
        <div className='d-flex flex-wrap'>
          {relatedProducts?.map((p)=>(

```

```

        <div className="card mx-auto mb-4" >
          <img src={`${process.env.REACT_APP_API}/api/v1/product/product-photo/${p._id}`}
            className="card-img-top"
            alt={p.name}
            key={p._id}
          />

          <div className="card-body">
            <h5 className="card-title fw-bold ">{p.name}</h5>
            <p className="card-text mb-2">{p.description.substring(0,30)}</p>
            <p className="card-text-price fw-bold ">{p.price}</p>
            <button className="btn btn-primary ms-1 me-1" onClick={()=>navigate(`/product/${p.slug}`)}>More Details</button>
            <button className="btn btn-success ms-2 " onClick={()=>{
              setCart([...cart,p])
              localStorage.setItem('cart',JSON.stringify([...cart,p]))
              toast.success("item added to cart")
            }}>Add to cart</button>
          </div>
        </div>
      ))}

    </div>
  </div>
</Layout>
)
}

export default ProductDetails

```

The screenshot shows a web browser window with multiple tabs open. The active tab displays a product detail page for a 'jacket'. The page has a header with the text 'Ecommerce APP'. Below the header is a large image of a brown jacket. To the right of the image, the title 'Product Details' is displayed, followed by the product name 'jacket', description 'pink kids jacket', price '\$3000', and category 'kids collection'. A dark grey 'Add to cart' button is located at the bottom right of this section.

Similer Product

No Similer Product Found



Pages/Contact.js

```

import axios from 'axios';
import React, { useState } from 'react'
import { toast } from 'react-hot-toast';
import Layout from '../../component/Layout/Layout'
import contact from "../../images/banner.jpg"

const Contact = () => {
  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [message, setMessage] = useState("")

```

```

const handleSubmit = async(e)=>{
  e.preventDefault()
  try{
    const {data}= await axios.post(
      `${process.env.REACT_APP_API}/api/v1/auth/contactus` ,
      {name,email,message}
    )
    if (data.success) {
      toast.success("thanku");
    } else {
      toast.error("something went wrong");
    }
  } catch (error) {
    console.log(error);
    toast.error("Something Went wrong");
  }
}
return (
  <Layout title="Contact us - Ecommerce App">
    <div className='container'>
      <div className='row d-flex align-items-center justify-content-center mt-md-5 mt-1'>
        <div className='col-md-8 mt-md-5 mt-1'>
          <div className="form-container">
            <form onSubmit={handleSubmit}>
              <h4 className="title">contact Us</h4>
              <div className="mb-3">
                <label htmlFor="name" className="form-label">
                  Name
                </label>
                <input
                  type="text"
                  name="name"
                  className="form-control"
                  value={name}
                  onChange={(e) => setName(e.target.value)}
                  required
                />
              </div>
              <div className="mb-3">
                <textarea type="text"
                  value={message}
                  placeholder="write a description"
                  className='form-control'
                  onChange={(e)=> setMessage(e.target.value)}
                />
              </div>
              <div className="mb-3">
                <label htmlFor="exampleInputEmail1" className="form-label">
                  Email address
                </label>
                <input
                  type="email"
                  name="email"
                  className="form-control"
                  value={email}
                  onChange={(e) => setEmail(e.target.value)}
                  required
                />
              </div>
              <div className="mb-4 mt-4 d-grid">
                <button type="submit" className="btn btn-primary">
                  Send
                </button>
              </div>
            </form>
          </div>
        </div>
        <div className='col-md-4 mt-md-5 mt-2'>
          <img src = {contact} alt="contact" className='img-fluid'/>
        </div>
      </div>
    </div>
  </Layout>
)

```

```

        </Layout>
    )
}

export default Contact

```

The screenshot shows a contact form titled "contact Us" on a website. The form has fields for "Name", "Email address", and a message area with placeholder text "write a description". A "Send" button is at the bottom. To the right of the form is a decorative image of a hand pointing at a digital interface with the word "eCommerce". The browser's address bar shows the URL: https://frail-outerwear-cod.cyclic.app/contact. The top navigation bar includes links for HOME, CATEGORY, FAZIL, and CART.

Pages/About.js

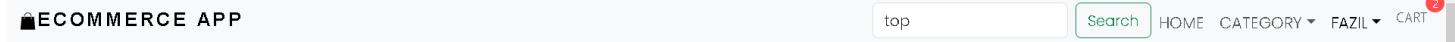
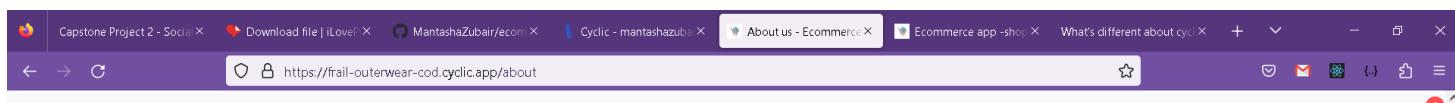
```

import React from 'react'
import Layout from '../../../../../component/Layout/Layout'
import about from "../../images/banner.jpg"

const About = () => {
  return (
    <Layout title="About us - Ecommerce app">
      <div className='container'>
        <div className='row d-flex align-items-center justify-content-center mt-md-5 mt-1'>
          <div className='col-md-6 mt-md-5 mt-1'>
            <p className='mt-5 fw-5 fs-5 text-secondary'> Welcome user this is our ecommerce website. Venture into the universe of form with <br/> 'Ecommerce website , a one-stop online mold entrance. Our comprehensive ladies' wear alter is propelled by worldwide runways, VIP looks and on-drift road style. Internet looking for men also accompanies the guarantee of acclaimed names without the substantial sticker prices. </p>
          </div>
          <div className='col-md-4 ms-2 mt-md-5 mt-2'>
            <img src = {about} alt="about" className='img-fluid' />
          </div>
        </div>
      </div>
    </Layout>
  )
}

export default About

```



Welcome user this is our ecommerce website. Venture into the universe of form with 'Ecommerce website , a one-stop online mold entrance. Our comprehensive ladies' wear alter is propelled by worldwide runways, VIP looks and on-drift road style. Internet looking for men also accompanies the guarantee of acclaimed names without the substantial sticker prices.



Package.json

```
{
  "name": "ecommerce_frontend",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^5.2.1",
    "axios": "^1.3.3",
    "braintree-web-drop-in-react": "^1.2.1",
    "moment": "^2.29.4",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-helmet": "^6.1.0",
    "react-helmet-async": "^1.3.0",
    "react-hot-toast": "^2.4.0",
    "react-icons": "^4.7.1",
    "react-router-dom": "^6.8.0",
    "react-scripts": "5.0.1",
    "react-toastify": "^9.1.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

```

        "last 1 chrome version",
        "last 1 firefox version",
        "last 1 safari version"
    ]
}
}

```

.env

```
REACT_APP_API = http://localhost:8080
```

Style/Homepage.css

```

.banner-img{
    margin-top: 30px;
    height: 200px;
}
.home-page{
    display: flex;
    flex-wrap: wrap;
}
.home-page .card{
    background-color: #f8f8f8;
    width: 18rem;
}
.home-page .card img:hover{
    transform: scale(0.9);
}
.home-page .card-img-top{
    height: 300px !important;
}
.home-page h1{
    color: rgb(120, 111, 111);
}
.home-page .card-title{
    text-transform: capitalize;
}
.home-page .card-text-price{
    color: green;
}
/* -----filters----- */
.filters h4{
    margin-top: 100px;
    color: rgb(120, 111, 111) !important;
    border-bottom: 1px solid gray;
}
.ant-checkbox-wrapper:first-of-type{
margin-left: 8px;
}
.ant-checkbox-wrapper{
    margin: 2px;
}

```

Style/CategoryProduct.js

```

/* =====All Category product===== */

/* =====CategoryProduct Cart===== */
.category-product-details .card{
    background-color: #f8f8f8;
    width: 18rem;
}
.category-product-details .card img:hover{
    transform: scale(0.9);
}
.category-product-details .card-img-top{
    height: 300px !important;
}

```

```

.cartegory-product-details h1{
    color: rgb(120, 111, 111);
}
.cartegory-product-details .card-title{
    text-transform: capitalize;
}
.cartegory-product-details .card-text-price{
    color:green
}

```

Style/ProductDetails.js

```

.product-details{
    margin-top: 100px !important;
}
/* .product-details-info{
    display: flex;
    flex-direction: column;
    justify-content: center;
} */
.product-details-info h6{
    margin-bottom: 20px !important;
}
.similer-product .card{
    background-color: #f8f8f8;
    width:18rem
}
.similer-product .card img:hover{
    transform: scale(0.9);
}
.similer-product .card-img-top{
    height: 300px !important;
}
.similer-product h3{
    color: rgb(120, 111, 111);
    margin-top: 20px;
    margin-bottom: 25px;
    margin-left: 50px;
    font-weight: bold;
}
.similer-product .card-title{
    text-transform: capitalize;
}
.similer-product .card-text-price{
    color:green
}

```

<https://frail-outerwear-cod.cyclic.app/>