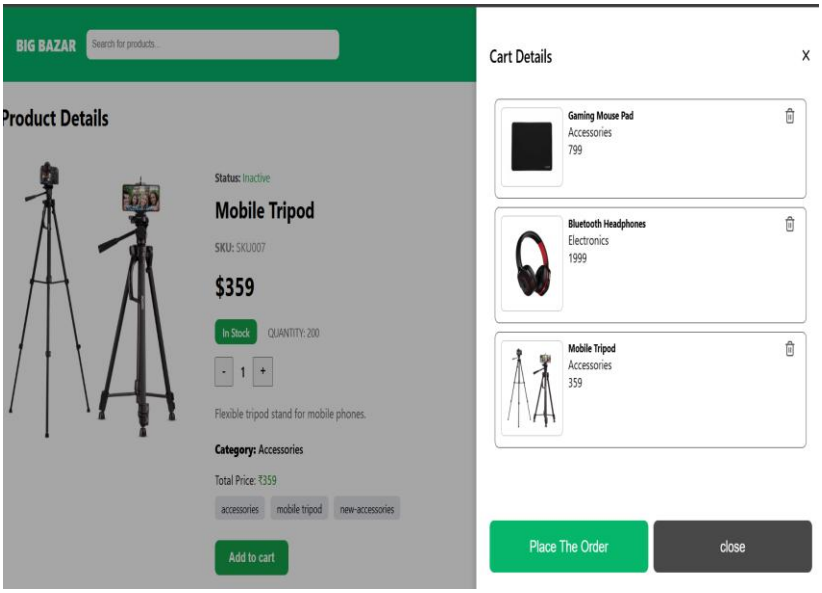
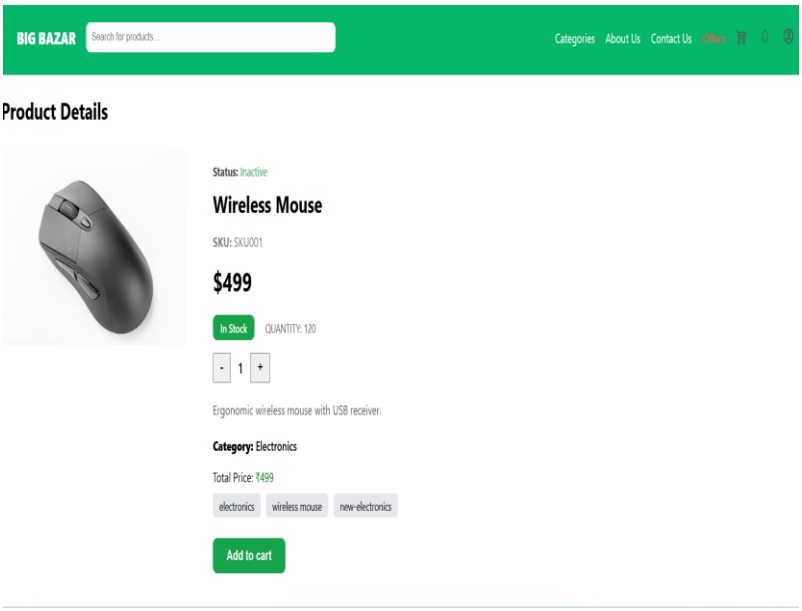
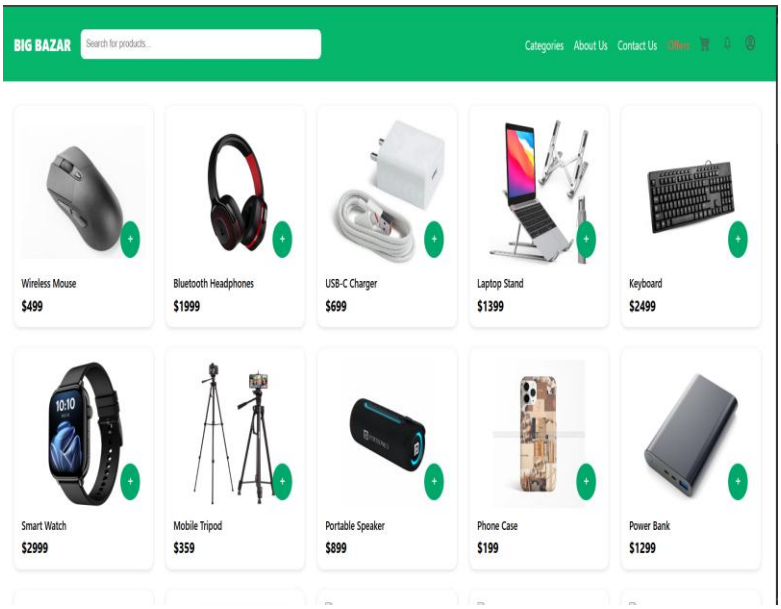
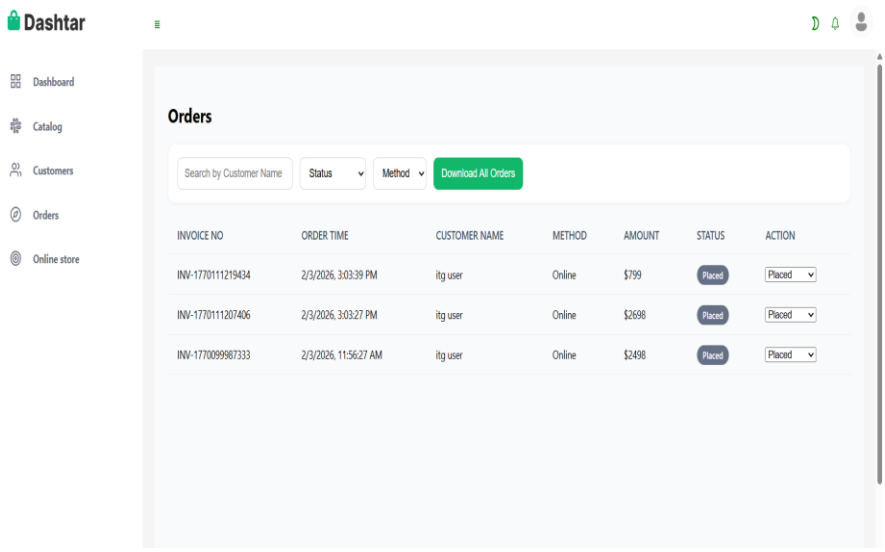
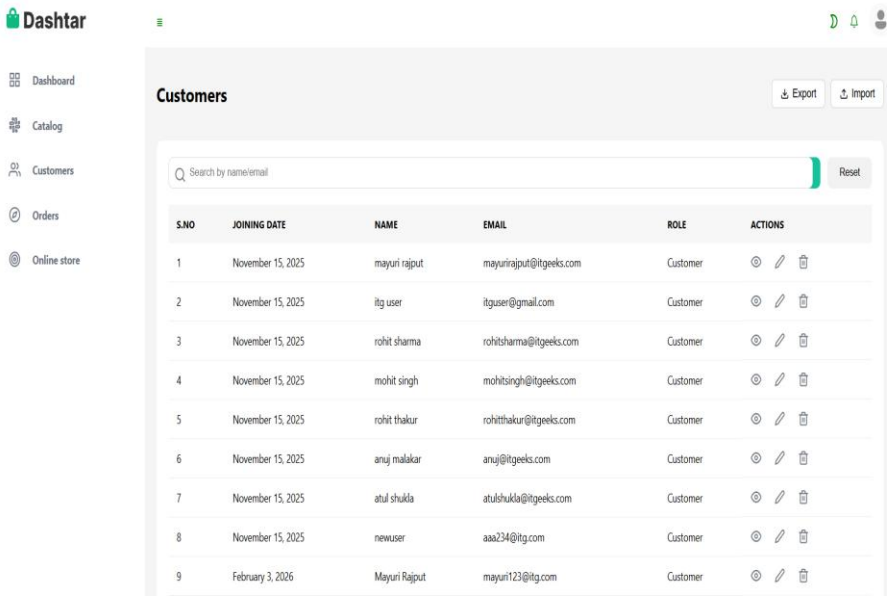
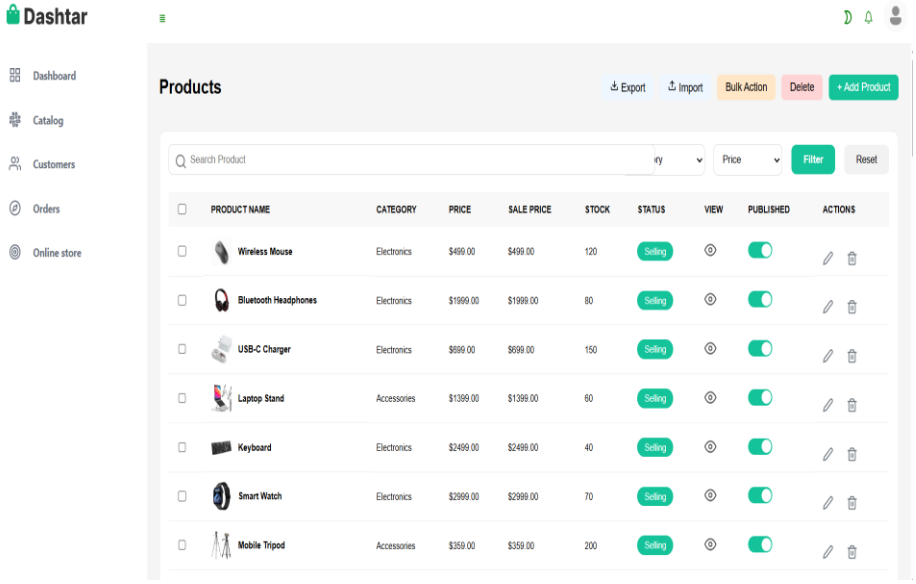

BIG BAZAR
Search for products...
Product Details
Status: Inactive
Mobile Tripod
SKU: SKU007
$359
In Stock
QUANTITY: 200
1
Flexible tripod stand for mobile phones.
Category: Accessories
Total Price: ₹359
accessories
mobile tripod
new-accessories
Add to cart
Cart Details
X
Gaming Mouse Pad
Accessories
799
Bluetooth Headphones
Electronics
1999
Mobile Tripod
Accessories
359
Place The Order
close

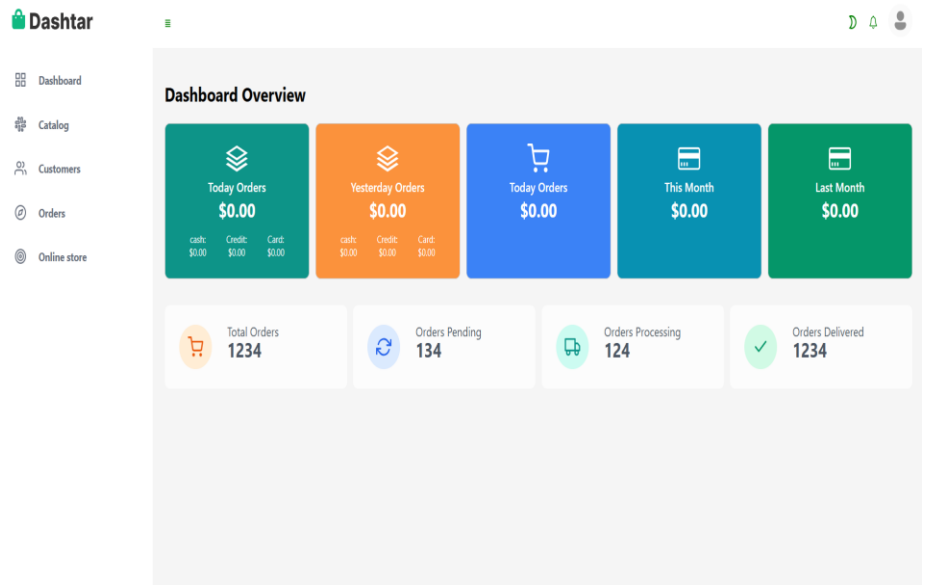

BIG BAZAR
Search for products...
Categories
About Us
Contact Us
Offers
Product Details
Status: Inactive
Wireless Mouse
SKU: SKU001
$499
In Stock
QUANTITY: 120
1
Ergonomic wireless mouse with USB receiver.
Category: Electronics
Total Price: ₹499
electronics
wireless mouse
new-electronics
Add to cart

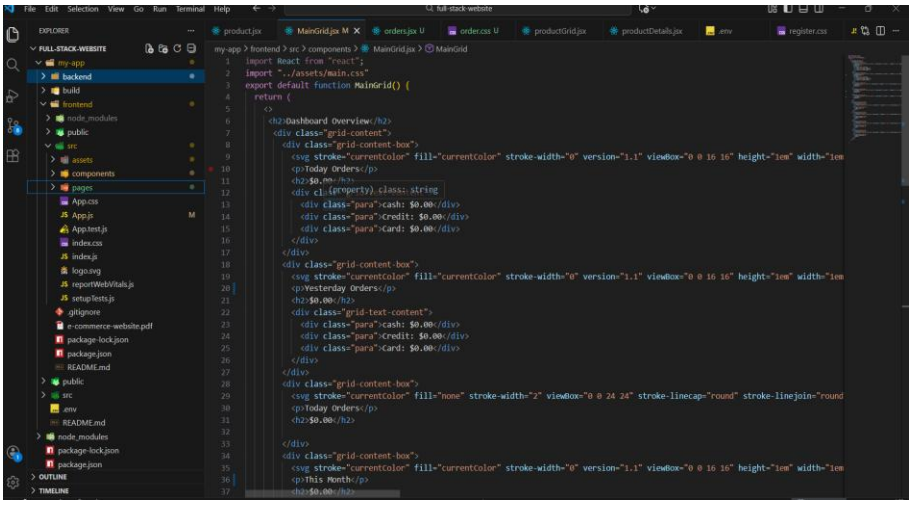

BIG BAZAR
Search for products...
Categories
About Us
Contact Us
Offers
Wireless Mouse
$499
Bluetooth Headphones
$1999
USB-C Charger
$699
Laptop Stand
$1399
Keyboard
$2499
Smart Watch
$2999
Mobile Tripod
$359
Portable Speaker
$899
Phone Case
$199
Power Bank
$1299

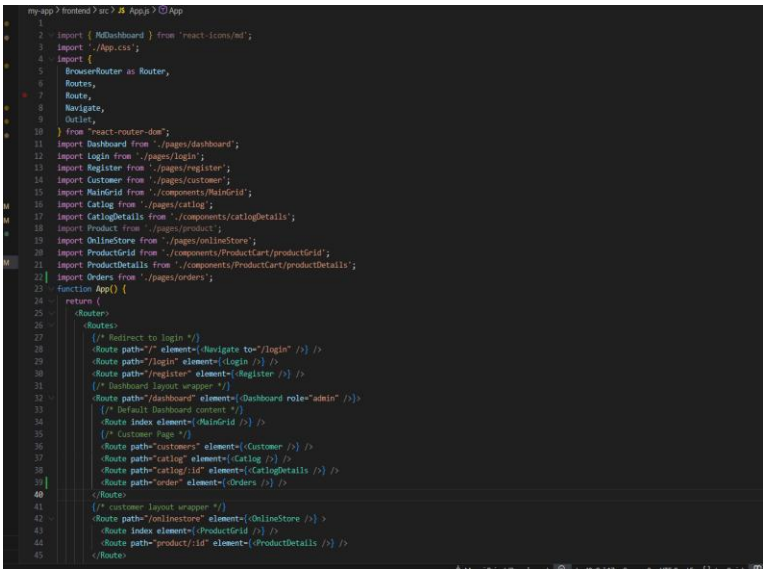

Dashtar
Dashboard
Catalog
Customers
Orders
Online store
Orders
Search by Customer Name
Status
Method
Download All Orders
INVOICE NO
ORDER TIME
CUSTOMER NAME
METHOD
AMOUNT
STATUS
ACTION
INV-1770111219434
2/3/2026, 3:03:39 PM
itg user
Online
$799
Placed
Placed
INV-1770111207406
2/3/2026, 3:03:27 PM
itg user
Online
$2698
Placed
Placed
INV-1770099987333
2/3/2026, 11:56:27 AM
itg user
Online
$2498
Placed
Placed

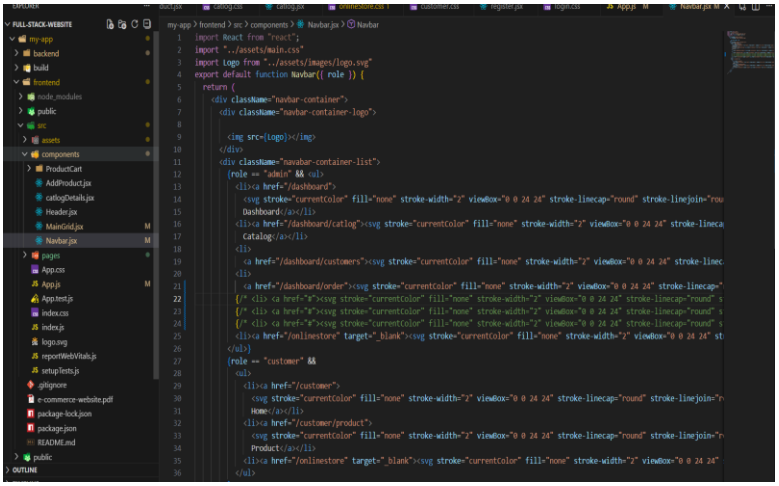

Dashtar
Dashboard
Catalog
Customers
Orders
Online store
Customers
Export
Import
Search by name/email
Reset
S.NO
JOINING DATE
NAME
EMAIL
ROLE
ACTIONS
1 November 15, 2025 mayuri rajput mayurirajput@itgeeks.com Customer
2 November 15, 2025 itg user itguser@gmail.com Customer
3 November 15, 2025 rohit sharma rohitsharma@itgeeks.com Customer
4 November 15, 2025 mohit singh mohitsingh@itgeeks.com Customer
5 November 15, 2025 rohit thakur rohitthakur@itgeeks.com Customer
6 November 15, 2025 anuj malakar anuj@itgeeks.com Customer
7 November 15, 2025 atul shukla atulshukla@itgeeks.com Customer
8 November 15, 2025 newuser aaa234@itg.com Customer
9 February 3, 2026 Mayuri Rajput mayuri123@itg.com Customer


Dashtar
Dashboard
Catalog
Customers
Orders
Online store
Products
Export
Import
Bulk Action
Delete
+ Add Product
Search Product
Price
Filter
Reset
PRODUCT NAME
CATEGORY
PRICE
SALE PRICE
STOCK
STATUS
VIEW
PUBLISHED
ACTIONS
Wireless Mouse Electronics $499.00 $499.00 120 Selling
Bluetooth Headphones Electronics $1999.00 $1999.00 80 Selling
USB-C Charger Electronics $699.00 $699.00 150 Selling
Laptop Stand Accessories $1399.00 $1399.00 60 Selling
Keyboard Electronics $2499.00 $2499.00 40 Selling
Smart Watch Electronics $2999.00 $2999.00 70 Selling
Mobile Tripod Accessories $359.00 $359.00 200 Selling


Dashtar
Dashboard
Catalog
Customers
Orders
Online store
Dashboard Overview
Today Orders
$0.00
cash: $0.00
Credit: $0.00
Card: $0.00
Yesterday Orders
$0.00
cash: $0.00
Credit: $0.00
Card: $0.00
Today Orders
$0.00
This Month
$0.00
Last Month
$0.00
Total Orders
1234
Orders Pending
134
Orders Processing
124
Orders Delivered
1234

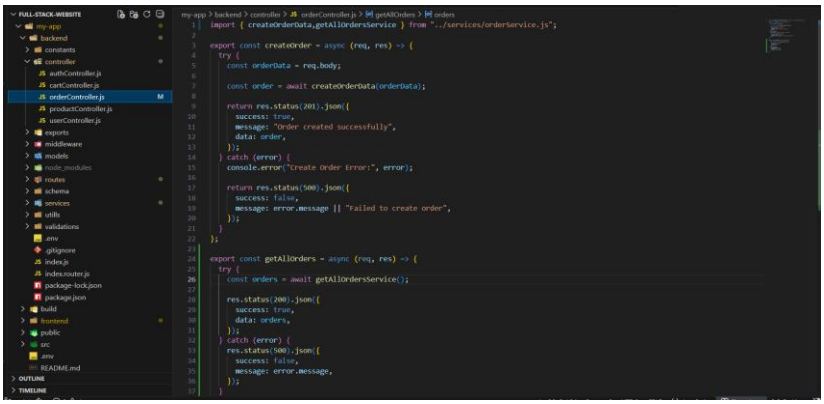




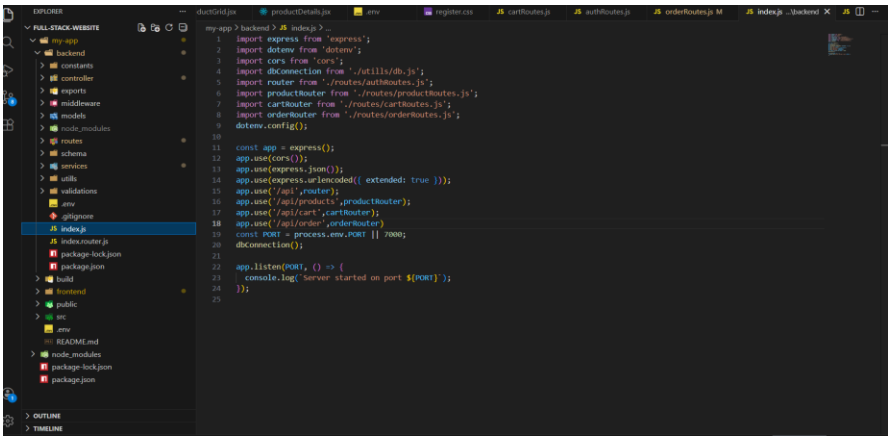


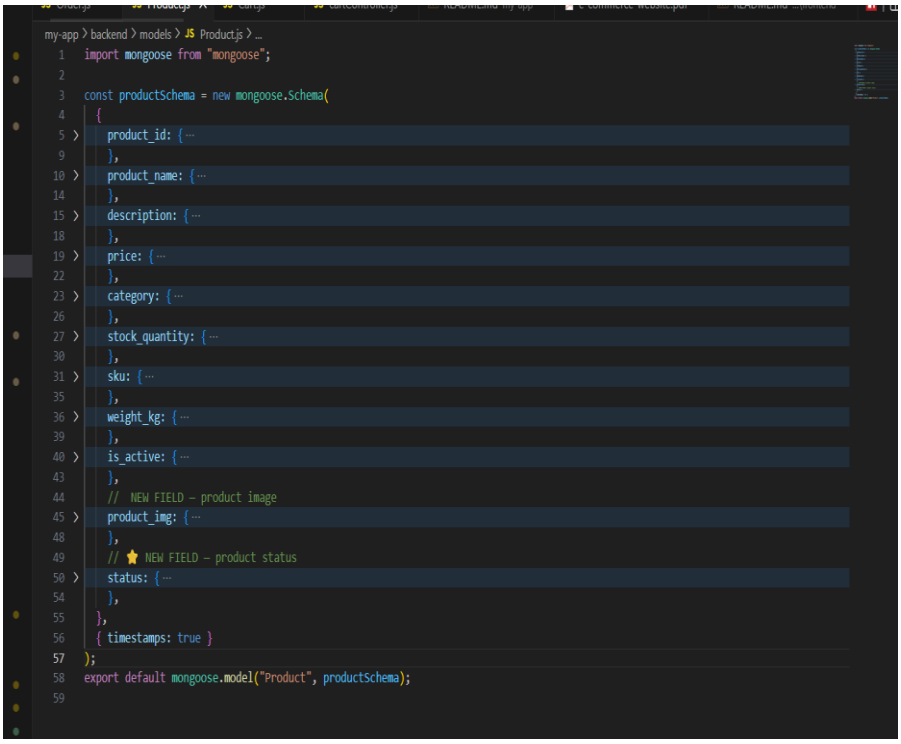
```
import React from "react";
import "../assets/main.css"
import Logo from "../assets/images/logo.svg"
export default function Navbar({ role }) {
  return (
    <div className="navbar-container">
      <div classNamme="navbar-container-logo">

        <img src={Logo}></img>
      </div>
      <div classNamme="navabar-container-list">
        {role == "admin" && <ul>
          <li><a href="/dashboard">
            <svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" stroke-linejoin="rou
            Dashboard</a></li>
          <li><a href="/dashboard/catlog"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-lineca
            Catalog</a></li>
          <li>
            <a href="/dashboard/customers"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linec
          <li>
            <a href="/dashboard/order"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="
          {/* <li> <a href="#"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" s
          {/* <li> <a href="#"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" s
          {/* <li> <a href="#"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" s
          <li><a href="/onlinestore" target="_blank"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" st
        </ul>}
        {role == "customer" &&
          <ul>
            <li><a href="/customer">
              <svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" stroke-linejoin="r
              Home</a></li>
            <li><a href="/customer/product">
              <svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24" stroke-linecap="round" stroke-linejoin="r
              Product</a></li>
            <li><a href="/onlinestore" target="_blank"><svg stroke="currentColor" fill="none" stroke-width="2" viewBox="0 0 24 24"
          </ul>
```

```
import Product from "../models/Product.js";
import Cart from "../models/Cart.js";

export const addedToCart = async (customer_id, product_id, quantity, price) => {

  const product = await Product.findById(product_id);
  if (!product)
    throw new Error("Product not found");
  // Prepare snapshot
  const productSnapshot = {
    product_id: product_id,
    product_name: product?.product_name,
    description: product?.description,
    price: product?.price,
    category: product?.category,
    stock_quantity: product?.stock_quantity,
    sku: product?.sku,
    weight_kg: product?.weight_kg,
    is_active: product?.is_active,
    product_img: product?.product_img,
    status: product?.status,
    quantity,
  };
  let cart = await Cart.findOne({ customer_id });
  if (!cart) {
    const createCart = await Cart.create({
      customer_id,
      items: [productSnapshot],
    })
  } else {
    const existingItem = cart.items.find(
      (item) => item.product_id === product_id
    );
    if (existingItem) {
      existingItem.quantity += quantity;
      existingItem.price += price;
    } else {
```



```
import { createOrderData,getAllOrdersService } from "../services/orderService.js";

export const createOrder = async (req, res) => {
  try {
    const orderData = req.body;

    const order = await createOrderData(orderData);

    return res.status(201).json({
      success: true,
      message: "Order created successfully",
      data: order,
    });
  } catch (error) {
    console.error("Create Order Error:", error);

    return res.status(500).json({
      success: false,
      message: error.message || "Failed to create order",
    });
  }
};

export const getAllOrders = async (req, res) => {
  try {
    const orders = await getAllOrdersService();

    res.status(200).json({
      success: true,
      data: orders,
    });
  } catch (error) {
    res.status(500).json({
      success: false,
      message: error.message,
    });
  }
```


EXPLORER
productDetails.jsx
.env
register.css
cartRoutes.js
authRoutes.js
orderRoutes.js M
index.js ..\backend
FULL-STACK-WEBSITE
my-app
backend
constants
controller
exports
middleware
models
node_modules
routes
schema
services
utills
validations
.env
.gitignore
index.js
index.router.js
package-lock.json
package.json
build
frontend
public
src
.env
README.md
node_modules
package-lock.json
package.json
OUTLINE
TIMELINE
my-app > backend > index.js > ...
import express from 'express';
import dotenv from 'dotenv';
import cors from 'cors';
import dbConnection from './utills/db.js';
import router from './routes/authRoutes.js';
import productRouter from './routes/productRoutes.js';
import cartRouter from './routes/cartRoutes.js';
import orderRouter from './routes/orderRoutes.js';
dotenv.config();
const app = express();
app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use('/api',router);
app.use('/api/products',productRouter);
app.use('/api/cart',cartRouter);
app.use('/api/order',orderRouter)
const PORT = process.env.PORT || 7000;
dbConnection();
app.listen(PORT, () => {
console.log(`Server started on port ${PORT}`);
});


my-app > backend > models > Product.js > ...
import mongoose from "mongoose";
const productSchema = new mongoose.Schema(
{
product_id: { ...
},
product_name: { ...
},
description: { ...
},
price: { ...
},
category: { ...
},
stock_quantity: { ...
},
sku: { ...
},
weight_kg: { ...
},
is_active: { ...
},
// NEW FIELD — product image
product_img: { ...
},
// ⭐ NEW FIELD — product status
status: { ...
},
},
{ timestamps: true }
);
export default mongoose.model("Product", productSchema);