

Govt. College, Solan

BCA Vith Semesters

List of Project Titles

Session 2024-25

| Sr. No | Roll No. | Name | Project Title | Software Used |
|--------|----------|-------------|--|----------------------|
| 1 | 2201 | Pankaj | Uber Website (Food Delivery) | HTML,CSS,JavaScript |
| 2 | 2202 | Aditya | Real Estate Property sale Rent Application | HTML,CSS,JavaScript |
| 3 | 2203 | Mansi | Portfolio website | HTML,CSS |
| 4 | 2204 | Saksham | Uber Website (Food Delivery) | HTML,CSS,JavaScript |
| 5 | 2205 | Preeti | Weather Website | HTML,CSS,JavaScript |
| 6 | 2207 | Simran | E commerce website (Fashion World) | HTML,CSS |
| 7 | 2208 | Divya | E commerce website (Fashion World) | HTML,CSS |
| 8 | 2209 | Ruchika | Online Voting System | HTML,PHP,CSS |
| 9 | 2210 | Varun | Uber Website (Food Delivery) | HTML,CSS,JavaScript |
| 10 | 2211 | Isha | Travel Website | HTML,CSS |
| 11 | 2212 | Muskan | Student Attendance System | HTML,CSS,JavaScript |
| 12 | 2214 | Kunal | E commerce website | HTML,JAVA script,css |
| 13 | 2215 | Arun | Stock Vision | HTML,CSS,Chart.JS |
| 14 | 2216 | Manasvi | Crime Reporting and monitoring system | HTML,CSS |
| 15 | 2218 | Preeti | Crime Reporting and monitoring system | HTML,CSS |
| 16 | 2220 | Sahil | Uber Website (Food Delivery) | HTML,CSS,JavaScript |
| 17 | 2221 | Mayur | E commerce website | HTML,JAVA script,css |
| 18 | 2222 | Abhinav | Airline Reservation Website | HTML,CSS,JavaScript |
| 19 | 2223 | Deepak | Stock Vision | HTML,CSS,Chart.JS |
| 20 | 2225 | Ram Kumar | Online Voting System | HTML,PHP,CSS |
| 21 | 2226 | Aakriti | Productivity Hub | HTML,CSS,JavaScript |
| 22 | 2227 | Priyanka | Productivity Hub | HTML,CSS,JavaScript |
| 23 | 2228 | Nidhi | Productivity Hub | HTML,CSS,JavaScript |
| 24 | 2229 | Komal | Portfolio website | HTML,CSS |
| 25 | 2230 | Malika | Weather Website | HTML,CSS,JavaScript |
| 26 | 2231 | Sahil Kumar | Ludo Game | JavaScript,HTML |
| 27 | 2232 | Tushar | Airline Reservation Website | HTML,CSS,JavaScript |
| 28 | 2233 | Deepika | Crime Reporting and monitoring system | HTML,CSS |
| 29 | 2234 | Gurpreet | Airline Reservation Website | HTML,CSS,JavaScript |

| | | | | |
|----|------|---------------|--|----------------------|
| 30 | 2235 | Lavya | Ludo Game | |
| 31 | 2236 | Kartik Sharma | Flappy Bird Clone | JavaScript,HTML |
| 32 | 2237 | Vansh | AUDIONIX | HTML,CSS,JavaScript |
| 33 | 2238 | Kunal Bhaskar | Ludo Game | HTML,CSS |
| 34 | 2239 | Akash | Airline Reservation Website | JavaScript,HTML |
| 35 | 2240 | Shruti | Online Shopping Platform | HTML,CSS,JavaScript |
| 36 | 2241 | Suman | Skincare website | HTML,CSS |
| 37 | 2243 | Shiva | Online Shopping System | Xampp,Apache,MySQL |
| 38 | 2244 | Varsha | Online Shopping System | HTML,CSS |
| 39 | 2245 | Diya | Skincare website | Xampp,Apache,MySQL |
| 40 | 2246 | Suhani | Online Shopping Platform | HTML,CSS |
| 41 | 2247 | Muskan | Online Shopping System | HTML,CSS |
| 42 | 2248 | Sarthak | Real Estate Property sale Rent Application | HTML,CSS,JavaScript |
| 43 | 2249 | Sneha | Travel Website | HTML,CSS |
| 44 | 2250 | Uma | Portfolio website | HTML,CSS |
| 45 | 2251 | Pratik | AUDIONIX | HTML,CSS |
| 46 | 2252 | Shreya Thakur | Travel Website | HTML,CSS |
| 47 | 2253 | Sachin | Flappy Bird Clone | HTML,CSS,JavaScript |
| 48 | 2254 | Kunal Bharti | Real Estate Property sale Rent Application | HTML,CSS,JavaScript |
| 49 | 2255 | Gargi | Crime Reporting and monitoring system | HTML,CSS |
| 50 | 2256 | Kartik | Flappy Bird Clone | HTML,CSS,JavaScript |
| 51 | 2257 | Vaibhav | Flappy Bird Clone | HTML,CSS,JavaScript |
| 52 | 2259 | Sahitya | Online Voting System | HTML,PHP,CSS |
| 53 | 2260 | Manish | Real Estate Property sale Rent Application | HTML,CSS,JavaScript |
| 54 | 2261 | Bhavika | Portfolio website | HTML,CSS |
| 55 | 2262 | Jatin | E-commerce (Airbuds) | HTML,JAVA script,css |
| 56 | 2263 | Priya | Skincare website | Xampp,Apache,MySQL |



Govt. Degree College, Solan

PROJECT REPORT ON E-COMMERCE WEBSITE



Submitted to:
Mrs. Simmi Sahni

Submitted By:
Kunal and Mayur
2214, 2221

CERTIFICATE

OF Completion

Govt. Degree College, Solan

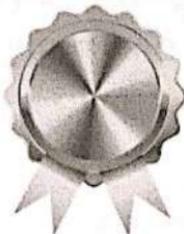
This is to certify that the project report titled "Ecommerce Website" submitted by [Kunal and Mayur] [Roll No: 2214 and 2221] of BCA department during the academic year [2024-25] is an authentic work carried out under my supervision and guidance.

The project report is submitted in partial fulfillment of the requirements .

This work presented in this project report has not submitted elsewhere for the award of any other degree or diploma



Mrs. Simmi Sahni
Project Guide
Department of
BCA
Date:



Principal
Govt College Solan
Date:

Signature of Co-Ordinator





Acknowledgment



We would like to express our sincere gratitude to all those who contributed to the successful completion of this project.

Firstly we would like to thank our college for providing us the opportunity to undertake this project.

A special thanks to our guide (Mrs. Simmi mam) for their valuable support, guidance and advice throughout the project.



This project has been a wonderful learning experience, and I sincerely thanks everyone who helped us along the way.



**Name: Mayur and Kunal
Roll no: 2221 and 2214
Student of BCA department**

Declaration

We [Mayur] and [Kunal] students of BCA department at Govt. Degree College, Solan hereby declare that the project titled "ecommerce website" submitted in partial fulfillment is our original work.

We confirm that this work has not been submitted elsewhere for any other degree or diploma. Any references, quotations, or borrowed ideas have been duly acknowledged in the report.

This project was carried out under the guidance of [Mrs. Simmi Sahni]

Date:

Signature

[Mayur, Roll No: 2221]:

Mayur

Signature

[Kunal, Roll No: 2214]:

Kunal

Project Report:

Ecommerce Website(content)

Introduction:

The E-commerce Website project is designed to provide users with an efficient online shopping experience.

It offers features like product browsing, secure payments, and order management.

This report outlines the key aspects of the project, including its objectives, design, and implementation.

Objectives:

To create a user-friendly platform for online shopping.

To ensure secure and seamless payment processing.

To provide an efficient product management system for administrators.

System Testing:

The existing manual shopping systems are time-consuming and inconvenient. The proposed e-commerce website addresses these issues with automated product management, secure payment integration, and order tracking.

Development Environment:

Frontend: React.js, HTML, CSS, JavaScript

Backend: Node.js, Express.js

Database: MongoDB

Implementation:

User Module: Users can register, log in, browse products, and manage orders.

Admin Module: Admins can manage products, view sales reports, and oversee orders.

Product Management: Products are organized into categories with detailed descriptions and images.

Shopping Cart and Checkout: Users can add products to the cart, review orders, and make secure payments.

Results and Discussion:

The website provides an intuitive user interface, secure payment gateway integration, and efficient product management. Performance analysis shows minimal latency, ensuring a smooth user experience.

References:

MongoDB Documentation

Node.js and Express.js Guides

React.js Official Documentation

Working Description of the Project

This project is an **E-commerce Website for Clothing**, designed to provide users with a seamless online shopping experience. The platform allows customers to browse a wide range of apparel, view product details, add items to their cart, and complete purchases securely.

The website features **user authentication**, enabling customers to create accounts, manage their orders, and track deliveries. It also includes an **admin panel** for store owners to manage inventory, update product details, and monitor sales.

Developed using **HTML, CSS, JavaScript, and backend technologies**, the website ensures a **responsive design** for mobile and desktop users. Additionally, secure payment gateways and order management systems are integrated to enhance user convenience and security.

This project aims to bridge the gap between retailers and consumers by offering a **user-friendly, visually appealing, and functional online clothing store**.

Coding

Root Layout (React JS):

```
client > app > layouts > ...
1
2 import './globals.css';
3
4 export const metadata = {
5   title: "Create Next App",
6   description: "Generated by create next app",
7 };
8
9 export default function RootLayout({ children }) {
10  return (
11    <html lang="en">
12      <body>
13
14
15        {children}
16
17      </body>
18    </html>
19  );
20 }
21
22
23
```

Category Form (React JS):

```
1 import React from 'react'
2
3 const CategoryForm = ({handleSubmit,value,setValue}) => {
4   return (
5     <div>
6       <h2>Category Form</h2>
7       <div>
8         <form onSubmit={handleSubmit}>
9           <input type="text"placeholder='Enter new category' value={value} onChange={(e)=>setValue(e.target.value)}></input>
10          <button type="submit">Submit</button>
11        </div>
12      </div>
13    </div>
14  )
15 }
16 </div>
17 </div>
18 </>
19 }
20 }
21
22 export default CategoryForm
```

Search Input (React JS):

```
client > src > components > form > SearchInput ...
1 import React, { useState } from 'react'
2 import { useSearch } from './.../context/Search'
3 import { useNavigate } from 'react-router-dom'
4 import axios from 'axios'
5
6 const SearchInput = () => {
7   const [values, setValues] = useSearch()
8   const navigate = useNavigate()
9   const handleSubmit = async (e) => {
10     e.preventDefault()
11     try {
12       const { data } = await axios.get(`/api/v1/product/search/${values.keyword}`)
13       setValues({ ...values, results: data })
14       navigate('/search')
15     } catch (error) {
16       console.log(error)
17     }
18   }
19   return (
20     <div>
21       <form role="search" onSubmit={handleSubmit}>
22         <input className="search" type="text" placeholder="search" value={values.keyword} onChange={e => setValues({ ...values, keyword: e.target.value })} />
23         <button className="btn" type="submit">Search</button>
24       </form>
25     </div>
26   )
27 }
28
29 export default SearchInput
```

Admin Menu (React JS):

```
client > src > components > Layout > AdminMenu.js ...
1 import React from 'react'
2 import { NavLink } from 'react-router-dom'
3 const AdminMenu = () => {
4
5   return (
6
7     <div className="list-group">
8       <h1>Admin Panel</h1>
9
10      <NavLink className="a" to="/dashboard/admin/create-category">Create Category</NavLink>
11      <br></br>
12      <NavLink className="b" to="/dashboard/admin/create-product">Create Product</NavLink>
13      <br></br>
14      <NavLink className="c" to="/dashboard/admin/products">products</NavLink>
15      <br></br>
16      <NavLink className="d" to="/dashboard/admin/users">Users</NavLink>
17
18    </div>
19  )
20
21 }
22
23 }
24
25 export default AdminMenu
```

Footer (React JS):

```
1
2 import React from 'react'
3 import { Link } from 'react-router-dom'
4
5 const Footer = () => {
6   return (
7     <div class="foot"><div class="foot2">All Right Reserved &copy; Technifoyt</div>
8     <Link className="foot2-link" to="/about">About</Link>|
9     <Link className="foot2-link" to="/contact">Contact</Link>|
10    <Link className="foot2-link" to="/policy">Privacy Policy</Link>
11
12    </div>
13  )
14 }
15
16 export default Footer
```

Header (React JS):

```
1 import React from 'react'
2 import {Link, NavLink} from 'react-router-dom'
3 import { useAuth } from '../context/auth'
4 import toast from "react-hot-toast"
5 import SearchInput from '../Form/SearchInput'
6 import useCategory from '../hooks/useCategory'
7 import { useCart } from '../context/Cart'
8 import {Badge} from 'antd'
9
10 const Header = () => {
11   const [cart]=useCart()
12   const [auth,setAuth]=useAuth()
13   const categories= useCategory()
14   const handleLogout()->(){
15     setAuth({
16       ...auth,user:null,token:''
17     })
18     localStorage.removeItem('auth')
19     toast.success('Logout Successfully')
20   }
21   return (
22     <>
23     <div className='nav2'>
24
25     <div class="nav1">
26
27
28     <div class="shift">
29
30     <Link className='nav2-link' to="/">Zenz Collection</Link>
31
32
33     </div>
34
35
36
37     <SearchInput/>
38
39     <ul>
40     <div class="flex">
```

Layout (React JS):

```
1 import React from 'react'
2 import Header from './Header'
3 import Footer from './Footer'
4
5
6 //import {ToastContainer} from 'react-toastify'
7 //import 'react-toastify/ReactToastify.css'
8 import {Toaster} from 'react-hot-toast';
9
10 const Layout = (props) => { //you can use destructuring ({children}) then in main {children}
11   return (
12     <div>
13       <Header/>
14
15
16       <main style={{minHeight:"65vh"}}>
17         <Toaster/>
18         {props.children}
19       </main>
20
21       <Footer/>
22
23     </div>
24   )
25 }
26
27 export default Layout
```

User Menu (React JS):

```
1 import React from 'react'
2 import { NavLink } from 'react-router-dom'
3
4 const UserMenu = () => {
5
6
7   return (
8
9
10     <div className='nap'>
11       <h1 className='pap'>Dashboard</h1>
12     <div className='v'>
13       <NavLink className="na" to="/dashboard/user/profile">Profile</NavLink>
14     </div>
15     <br></br>
16     <div className='h'>
17       <NavLink className="ha" to="/dashboard/user/orders">Orders</NavLink>
18     </div>
19
20
21     </div>
22   )
23 }
24
25 export default UserMenu
```

(React JS):

```
1 import React, { useState,useEffect } from 'react'
2 import { useLocation, useNavigate } from 'react-router-dom'
3
4 const Spinner = ({path = "login"}) => {
5   const [count ,setCount]=useState(5)
6   const navigate = useNavigate()
7   const location =useLocation()
8   useEffect(()=>{
9     const interval = setInterval(() => {
10      |   setCount((prevValue)=>--prevValue)
11      | }, 1000);
12     count === 0 && navigate(`/${path}`, {
13     |   state:location.pathname,
14     | })
15     return ()=> clearInterval(interval)
16   },[count,navigate,location,path])
17   return (
18     <div>
19     |   <h1>redirecting to you in {count} seconds</h1>
20     | </div>
21   )
22 }
23
24 export default Spinner
```

Authorization (React JS):

```
1 import axios from 'axios'
2 import { useContext,useEffect, useState,createContext } from 'react'
3
4 const AuthContext = createContext()//store authcontext
5
6
7 const AuthProvider=(props)=>{
8   const [auth,setAuth]=useState({
9     user:null,
10    token:""
11  })
12  //default axios
13  axios.defaults.headers.common['Authorization']=auth?.token
14  useEffect(()=>{
15    const data=localStorage.getItem('auth')
16    if(data){
17      const parseData=JSON.parse(data)
18      setAuth({
19        ...auth,
20        user:parseData.user,
21        token:parseData.token,
22      })
23    }
24    //eslint-disable-next-line
25    //for fix the warnings
26  },[])
27  return(
28    <AuthContext.Provider value={{[auth,setAuth]}}>
29      {props.children}
30    </AuthContext.Provider>
31  )
32 }
33
34 //custom hook
35 const useAuth={()=>useContext(AuthContext)}
36 export {useAuth,AuthProvider}
37
```

Cart (React JS):

```
1 import { useContext, useState,createContext,useEffect } from 'react'
2
3 const CartContext = createContext()//store searchcontext
4
5
6 const CartProvider=(props)=>{
7   const [cart,setCart]=useState([])
8
9   useEffect(()=>{
10     let existingCartItem =localStorage.getItem('cart')
11     if(existingCartItem) setCart(JSON.parse(existingCartItem))
12   },[])
13
14   return(
15     <CartContext.Provider value={{[cart,setCart]}}>
16       {props.children}
17     </CartContext.Provider>
18   )
19 }
20
21 //custom hook
22 const useCart={()=>useContext(CartContext)}
23 export {useCart,CartProvider}
24
```

Search (React JS):

```
client > src > context > Searchjs > ...
1
2 import { useContext, useState, createContext } from 'react'
3
4 const SearchContext = createContext()//store searchcontext
5
6
7 const SearchProvider=(props)=>{
8   const [auth,setAuth]=useState({
9     keyword:"",
10    results:[],
11  })
12
13
14  return(
15    <SearchContext.Provider value={{auth,setAuth}}>
16      {props.children}
17    </SearchContext.Provider>
18  )
19 }
20
21 //custom hook
22 const useSearch=()=>useContext(SearchContext)
23 export {useSearch,SearchProvider}
24
```

Use Category (React JS):

```
1 import {useState,useEffect} from 'react'
2 import axios from 'axios'
3
4 export default function useCategory(){
5   const [categories,setCategories]=useState([])
6
7
8   //get cate
9   const getCategories=async()->{
10    try{
11      const {data}=await axios.get('/api/v1/category/get-category')
12      setCategories(data?.category)
13    }catch(error){
14      console.log(error)
15    }
16  }
17  useEffect(()=>{
18    getCategories()
19  },[])
20  return categories;
21 }
```

Register Page (React JS):

```
1 import React, { useState } from 'react'
2 import Layout from '../components/Layout/Layout'
3 //import { toast } from 'react-toastify'
4 import axios from 'axios'
5 import { useNavigate } from 'react-router-dom'
6 import toast from 'react-hot-toast'
7
8 const Register = () => {
9   const [name, setName]=useState("")
10  const [email, setEmail]=useState("")
11  const [password, setPassword]=useState("")
12  const [phone, setPhone]=useState("")
13  const [address, setAddress]=useState("")
14  const [answer, setAnswer]=useState("")
15  const navigate = useNavigate()
16
17  //form function
18  const handleSubmit=async(e)=>{
19    e.preventDefault()
20    //console.log(name,email,password,address,phone);//for check name email or thing going to backend or not
21    // toast.success('register sucessfully');
22
23    try{
24      const res =await axios.post(`${process.env.REACT_APP_API}/api/v1/auth/register`,{
25        name,
26        email,
27        password,
28        phone,
29        address,
30        answer})
31
32
33      if(res.data.success){
34        toast.success(res.data.message)
35        navigate('/login')
36      }else{
37        toast.error(res.data.message)
38      }
39    } catch(error){
40      console.log(error)
41      toast.error('something went wrong')
42    }
43  };
44  console.log(process.env.REACT_APP_API)
```

```

47 return (
48   <Layout>
49     <div className="register">
50       <form onSubmit={handleSubmit} >
51         <h1 className='hi'>Register Page</h1>
52         <label>Name</label><br></br>
53         <input type="text" className='w1'
54           value={name}
55           onChange={(e)=>setName(e.target.value)}
56           required>
57         </input>
58         <br></br>
59
60         <label>Email</label><br></br>
61         <input type="email" className='w1'
62           value={email}
63           onChange={(e)=>setEmail(e.target.value)}
64           required></input>
65
66         <br></br>
67         <label>Password</label><br></br>
68         <input type="password" className='w1'
69           value={password}
70           onChange={(e)=>setPassword(e.target.value)}
71           required></input>
72         <br></br>
73
74         <label>Phone</label><br></br>
75         <input type="text" className='w1'
76           value={phone}
77           onChange={(e)=>setPhone(e.target.value)}
78           required></input>
79         <br></br>
80

```

```

81         <label>Address</label><br></br>
82         <input type="text" className='w1'
83           value={address}
84           onChange={(e)=>setAddress(e.target.value)}
85           required></input>
86         <br></br>
87         <label>Answer</label><br></br>
88         <input type="text" className='w1'
89           value={answer}
90           onChange={(e)=>setAnswer(e.target.value)}
91           placeholder="what is your favourite sports"
92           required></input>
93
94
95         <button className='but1' type="submit">Register</button>
96
97       </form>
98     </div>
99   </Layout>
100 )
101
102
103 export default Register
104

```

Login Page (React JS):

```
1 import React, { useState } from 'react'
2 import Layout from '../components/Layout/Layout'
3 //import { toast } from 'react-toastify'
4 import axios from 'axios'
5 import { useNavigate, useLocation } from 'react-router-dom'
6 import toast from 'react-hot-toast'
7 import { useAuth } from '../context/auth'
8
9 const Login = () => {
10
11
12   const [email, setEmail] = useState('')
13   const [password, setPassword] = useState('')
14   const [auth, setAuth] = useAuth()
15
16
17
18   const navigate = useNavigate()
19   const location = useLocation()
20
21   //form function
22   const handleSubmit = async (e) => {
23     e.preventDefault()
24     //console.log(name, email, password, address, phone); //for check name email or thing going to backend or not
25     // toast.success('register successfully');
26
27     try {
28       const res = await axios.post(`${process.env.REACT_APP_API}/api/v1/auth/login`, {email, password})
29
30
31       if (res.data.success) {
32         toast.success(res.data.message)
33         setAuth({
34           ...auth,
35           user: res.data.user,
36           token: res.data.token,
37         });
38         localStorage.setItem('auth', JSON.stringify(res.data))
39         navigate(location.state || '/') //navigate use for redirect
40       } else {
41         toast.error(res.data.message)
42       }
43     }
44   }
45 }
```

Login Page (React JS):

```
1 import React, { useState } from 'react'
2 import Layout from '../components/Layout/Layout'
3 //import { toast } from 'react-toastify'
4 import axios from 'axios'
5 import { useNavigate ,useLocation} from 'react-router-dom'
6 import toast from 'react-hot-toast'
7 import { useAuth } from '../context/auth'
8
9 const Login = () => {
10
11
12   const [email,setEmail]=useState("")
13   const [password,setPassword]=useState("")
14   const [auth,setAuth]=useAuth()
15
16
17
18   const navigate = useNavigate()
19   const location=useLocation()
20
21   //form function
22   const handleSubmit=async(e)=>{
23     e.preventDefault()
24     //console.log(name,email,password,address,phone);//for check name email or thing going to backend or not
25     // toast.success('register successfully');
26
27     try{
28       const res =await axios.post(`${process.env.REACT_APP_API}/api/v1/auth/login`,{email,password})
29
30
31       if(res.data.success){
32         toast.success(res.data.message)
33         setAuth({
34           ...auth,
35           user:res.data.user,
36           token:res.data.token,
37         });
38         localStorage.setItem('auth',JSON.stringify(res.data))
39         navigate(location.state|| '/')//navigate use for redirect
40       }else{
41         toast.error(res.data.message)
42       }
43     }
```

App (all routes) React JS:

```
1 import './App.css';
2 import React from 'react'
3 import {Routes,Route} from 'react-router-dom'
4 import HomePage from './pages/HomePage';
5 import Policy from './pages/Policy';
6 import About from './pages/About';
7 import Contact from './pages/Contact';
8 import Pagenotfound from './pages/Pagenotfound';
9 import Register from './pages/Auth/Register';
10 import {ToastContainer} from 'react-toastify'
11 import 'react-toastify/ReactToastify.css'
12 import Login from './pages/Auth/Login';
13 import Dashboard from './pages/user/Dashboard';
14 import PrivateRoute from './components/Routes/Private';
15 import ForgotPassword from './pages/Auth/ForgotPassword';
16 import AdminRoute from './components/Routes/AdminRoute';
17 import AdminDashboard from './pages/Admin/AdminDashboard';
18 import CreateCategory from './pages/Admin/CreateCategory';
19 import CreateProduct from './pages/Admin/CreateProduct';
20 import Userss from './pages/Admin/Userss';
21 import Orders from './pages/user/Orders';
22 import Profile from './pages/user/Profile';
23 import Products from './pages/Admin/Products';
24 import UpdateProduct from './pages/Admin/UpdateProduct';
25 import Search from './pages/Search';
26 import ProductDetails from './pages/ProductDetails'
27 import Categories from './pages/Categories'
28 import CategoryProduct from './pages/CategoryProduct'
29 import CartPage from './pages/CartPage';
30
31 const App = () => {
32   return (
33     <>
34     <Routes>
35       <Route path='/' element={<HomePage/>}/>
36       <Route path='/product/:slug' element={<ProductDetails/>}/>
37       <Route path='/categories' element={<Categories/>}/>
38       <Route path='/cart' element={<CartPage/>}/>
39       <Route path='/category/:slug' element={<CategoryProduct/>}/>
40       <Route path='/search' element={<Search/>}/>
```

```

48 <Route path='/dashboard' element={<AdminRoute />}/>
49 <Route path='admin' element={<AdminDashboard />}/>
50 <Route path='admin/create-category' element={<CreateCategory />}/>
51 <Route path='admin/create-product' element={<CreateProduct />}/>
52 <Route path='admin/product/:slug' element={<UpdateProduct />}/>
53 <Route path='admin/products' element={<Products />}/>
54 <Route path='admin/users' element={<Userss />}/>
55
56 </Route>
57
58 <Route path='/about' element={<About />}/>
59 <Route path='/contact' element={<Contact />}/>
60 <Route path='/policy' element={<Policy />}/>
61 <Route path='/register' element={<Register />}/>
62 <Route path='/forgot-password' element={<ForgotPassword />}/>
63 <Route path='/login' element={<Login />}/>
64 <Route path='*' element={<Pagenotfound />}/>
65
66 </Routes>
67 </>
68 )// * means if the four route not working then it will work
69 }
70
71 export default App

```

pp Test (React JS):

```

1 ✓ import { render, screen } from '@testing-library/react';
2 import App from './App';
3
4 ✓ test('renders learn react link', () => {
5   render(<App />);
6   const linkElement = screen.getByText(/learn react/i);
7   expect(linkElement).toBeInTheDocument();
8 });
9

```

Report Web Titles (React JS):

```
1  const reportWebVitals = onPerfEntry => {
2    if (onPerfEntry && onPerfEntry instanceof Function) {
3      import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
4        getCLS(onPerfEntry);
5        getFID(onPerfEntry);
6        getFCP(onPerfEntry);
7        getLCP(onPerfEntry);
8        getTTFB(onPerfEntry);
9      });
10   }
11 };
12
13 export default reportWebVitals;
14
```

Setup Test (React JS):

```
client > src >  setupTests.js
```

```
1 // jest-dom adds custom jest matchers for asserting on DOM nodes.
```

```
2 // allows you to do things like:
```

```
3 // expect(element).toHaveTextContent(/react/i)
```

```
4 // learn more: https://github.com/testing-library/jest-dom
```

```
5 import '@testing-library/jest-dom';
```

```
6
```

Home Page Code (React JS):

```
client > src > pages > HomePage.js > ...
1
2 import React, { useEffect, useState } from 'react'
3 import Layout from '../components/Layout/Layout'
4
5 import axios from 'axios'
6 import { Checkbox, Radio } from 'antd'
7 import { prices } from '../components/Prices'
8 import { useNavigate } from 'react-router-dom'
9 import { useCart } from '../context/Cart'
10 import toast from 'react-hot-toast'
11
12 const HomePage = () => {
13   const navigate=useNavigate();
14   const [cart,setCart]=useCart();
15   const [products,setProducts] = useState([])
16   const [categories,setCategories]=useState([])
17   const[checked,setChecked]=useState([])
18   const[radio,setRadio]=useState([])
19   const [total,setTotal]=useState(0)
20   const [page,setPage]=useState(1)
21   const [loading,setLoading] = useState(false)
22
23
24
25   //get all category
26   const getAllCategory=async()->{
27     try{
28       const {data} = await axios.get('/api/v1/category/get-category')
29       if(data?.success){
30         setCategories(data?.category);
31       }
32     }catch(error){
33       console.log(error)
34     }
35
36
37
38   }
39 };
40
41 useEffect(())->{
42   getAllCategory();
43   getTotal();
44 },[])
```

```

46 //get all products
47
48 const getAllProducts=async()->{
49   try{
50     setloading(true)
51     const {data}= await axios.get(`/api/v1/product/product-list/${page}`);
52     setLoading(false)
53     setProducts(data?.products);
54
55   }catch(error){
56     setloading(false)
57     console.log(error)
58   }
59 }
60
61 //get total count
62 const getTotal = async()->{
63   try{
64     const {data}=await axios.get(`/api/v1/product/product-count`)
65     if(data?.success){
66       setTotal(data?.total)
67     }
68   }catch(error){
69     console.log(error)
70   }
71 }
72
73 useEffect()->{
74   if (page===1) return;
75   loadMore()
76 },[page])
77 //loadmore
78 const loadMore=async()->{
79   try{
80     setloading(true)
81     const {data}= await axios.get(`/api/v1/product/product-list/${page}`)
82     setloading(false)
83     setProducts([...products,...data?.products])
84   }catch(error){
85     console.log(error)
86     setLoading(false)
87   }
88 }

```

```

98 //filter by category
99
100 const handleFilter=(value,id)->{
101   let all=[...checked]
102   if(value){
103     all.push(id)
104   }else{
105     all=all.filter(c=> c!==(id))
106   }
107   setChecked(all);
108 }
109
110 useEffect()->{
111   if(!checked.length || !radio.length) getAllProducts();
112   //eslint-disable-next-line
113   },[checked.length,radio.length])
114
115   useEffect()->{
116     if(checked.length || radio.length) filterProduct()
117   },[checked,radio])
118
119 //get filtered product
120 const filterProduct = async()->{
121   try{
122     const {data}=await axios.post(`/api/v1/product/product-filters`,{checked,radio})
123     setProducts(data?.products)
124   }catch(error){
125     console.log(error)
126   }
127 }
128
129 }

```

```

120 return (
121   <div className='baap'>
122     <layout>
123       <div className='flex8'>
124         <div className='right'>
125           <div>
126             <h1 className='pass'>Filter by category</h1>
127             {categories?.map(c=><div className='check'>
128               <Checkbox key={c._id} onChange={e=> handleFilter(e.target.checked,c._id)}>
129                 {c.name}
130               </Checkbox>
131             </div>
132           </div>
133         </div>
134       </div>
135       <div className='radio'>
136         <h1>Filter by prices</h1>
137         <Radio.Group onChange={e=> setRadio(e.target.value)}>
138           {prices?.map(p=>{
139             <div key={p._id}>
140               <Radio value={p.array}>{p.name}</Radio>
141             </div>
142           </div>
143         </Radio.Group>
144       </div>
145       <div className='reset'>
146         <button onClick={()=>window.location.reload()}>Reset Filter</button>
147       </div>
148       <div className='text'><h1 className='h9'><u>All Products</u></h1>
149     </div>
150     <div>
151       <div className='flex5'>
152         {products?.map((p)=>{

```

```

160
161       <div className='design'>
162         <img className='hot' src={`\api/v1/product/product-photo/${p._id}`} alt={p.name} />
163
164         <h1>{p.name}</h1>
165         <h4>${p.price}</h4>
166         <h4>{p.description.substring(0,30)}</h4>
167
168         <button onClick={()=>navigate(`/product/${p.slug}`)}>More Details</button>
169
170         <button onClick={()=>{setCart([...cart,p]);
171           localStorage.setItem('cart',JSON.stringify([...cart,p])
172           toast.success('Item added to a cart');
173           }}>Add to cart</button>
174
175       </div>
176     </div>
177
178     </div>
179
180     </div>
181     <div>
182       {products && products.length < total && (
183         <button onClick={e=>{
184           e.preventDefault();
185           setPage(page + 1);
186         }}>
187         {loading ? "Loading....": "Loadmore"}</button>
188       )}
189     </div>
190   </div>
191 </div>

```

OUTPUT

Forget and Reset Page:

The screenshot shows the 'Reset Password' page of the Gen-Z Collection website. The page features a dark-themed navigation bar at the top with the site name 'Gen-Z Collection', a search bar, and links for 'HOME', 'CATEGORY', 'LOGIN', 'REGISTER', and 'CART'. The main content area contains a 'RESET PASSWORD' form with the following fields: 'Email' (containing 'admin@gmail.com'), 'password', and 'confirm password'. A 'Reset' button is located at the bottom of the form. The footer of the page includes the text 'All Right Reserved © TechinfoKM' and links for 'About', 'Contact', and 'Privacy Policy'.

Register Page:

The screenshot shows the 'Register Page' of the Gen-Z Collection website. The page features a dark-themed navigation bar at the top with the site name 'Gen-Z Collection', a search bar, and links for 'HOME', 'CATEGORY', 'LOGIN', 'REGISTER', and 'CART'. The main content area contains a 'Register Page' form with the following fields: 'Name', 'Email', 'password', 'confirm password', and 'what is your favourite sport'. A 'Register' button is located at the bottom of the form. The footer of the page includes the text 'All Right Reserved © TechinfoKM' and links for 'About', 'Contact', and 'Privacy Policy'.

Men's Category Page:

Gen-Z Collection Search Search HOME CATEGORY KUNAL CART

Category - Men Collections
10 result found



Shirt
It is very cool shirt...
£17
More Details | ADD TO CART



Shrug
Coat...
£15
More Details | ADD TO CART



Jacket
Coat...
£27
More Details | ADD TO CART

Women's Category Page:

Gen-Z Collection Search Search HOME CATEGORY KUNAL CART

Category - Women Collections
7 result found



Coat and Pants
Coat...
£15
More Details | ADD TO CART



Long Tunic
Tunic...
£30
More Details | ADD TO CART



Long tunic
Tunic...
£100
More Details | ADD TO CART

- ALL CATEGORIES
- MEN COLLECTIONS
- WOMEN COLLECTIONS
- CHILDREN COLLECTIONS

Kid's Category Page:

Gen-Z Collection Search Search HOME CATEGORY KUNAL CART

Category - Children Collections
5 result found



Shoes
Shoes...
£20
More Details | ADD TO CART



Rain
Coat...
£10
More Details | ADD TO CART



Polo-Shorts
Pants...
£100
More Details | ADD TO CART

- ALL CATEGORIES
- MEN COLLECTIONS
- WOMEN COLLECTIONS
- CHILDREN COLLECTIONS

Login Page:

Gen-Z Collection part [HOME](#) [CATEGORY](#) [LOGIN](#) [REGISTER](#) [CART](#)

Login Form

All Right Reserved © TechinfoKM
[About](#) | [Contact](#) | [Privacy Policy](#)

Home Page:

Gen-Z Collection [HOME](#) [CATEGORY](#) [ADMIN](#) [CART](#)

Filter by category

- Men's Collections
- Women's Collections
- Children's Collections

Filter by prices

- ₹ 0 to 1000
- ₹ 2000 to 3000
- ₹ 4000 to 5000
- ₹ 6000 to 7000
- ₹ 8000 to 9000

FRESH THREADS

All Products

Pant, Shoes, T-Shirt

₹ 990

Full Package

[View Details](#) [Add to cart](#)

Warm Shoes

₹ 1000

₹ 999/₹ 999

[View Details](#) [Add to cart](#)

Sbarara

₹ 2000

₹ 1990

[View Details](#) [Add to cart](#)

Selected by filters Page:

The screenshot shows a product page for 'Lehnga Choli'. On the left, there are two filter sections: 'Filter by category' with radio buttons for 'Men Collections', 'Women Collections' (selected), and 'Children Collections'; and 'Filter by prices' with radio buttons for price ranges from '₹ 0 to 1000' to '₹ 10000 to 20000'. The main content area features a banner image of a clothing store, a navigation bar with '<' and '>' arrows, and the text 'All Products'. Below the banner is a product card for 'Lehnga Choli' with a '₹ 1000' price tag and a 'View Details | Add to cart' button.

Admin Dashboard:

The screenshot shows the Admin Dashboard for 'Gen-Z Collection'. The top navigation bar includes 'HOME', 'CATEGORY', 'ADMIN', and 'CART'. The main content area is divided into two sections: 'Admin Panel' on the left with links for 'Create Category', 'Create Product', 'products', and 'Users'; and 'Create Product' on the right with a form containing fields for 'Upload Photo', 'Name', 'Write a short phrase', 'Set up a Price', and 'Set up a quantity', along with a 'CREATE PRODUCT' button. At the bottom, a footer contains the text 'All Right Reserved © TechinfoKM' and links for 'About', 'Contact', and 'Privacy Policy'.

Create Category, Delete and edit page:

The screenshot shows the Admin Dashboard for 'Gen-Z Collection' with a focus on category management. The top navigation bar is the same as in the previous screenshot. The main content area is divided into three sections: 'Admin Panel' on the left with links for 'Create Category', 'Create Product', 'products', and 'Users'; 'Manage Category' in the center with a 'Category Form' containing an 'Enter new category' input field and a 'Submit' button; and 'Categories' on the right, which is a table listing categories and their actions.

| Categories | |
|----------------------|-------------|
| Name | Actions |
| Men Collections | Edit Delete |
| Women Collections | Edit Delete |
| Children Collections | Edit Delete |

Data collected by admin through Mongo DB:

The screenshot shows the MongoDB Compass interface. The breadcrumb path is 'localhost:27017 > ecommerce > users'. The top navigation bar includes 'Documents 6', 'Aggregations', 'Schema', 'Indexes 2', and 'Validation'. A search bar contains the text 'Type a query: [field: 'value'] or Generate query'. Below the search bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. On the right, there are buttons for 'Explain', 'Reset', 'Find', and 'Options'. The main area displays two JSON documents representing users.

```
{
  "_id": "ObjectId('679602b08b79924b75614e9')",
  "name": "admin",
  "email": "admin@gmail.com",
  "password": "$2b5105j9jc3c0JfPBHNTs06t9uHTfBwGHVWTEqNqB79f8JrfKsXGn0",
  "phone": "12345",
  "address": "olgi",
  "answer": "golf",
  "role": 1,
  "createdAt": "2025-01-26T09:39:07.522+00:00",
  "updatedAt": "2025-01-26T09:39:07.522+00:00",
  "__v": 0
}
```

```
{
  "_id": "ObjectId('6796072170e35326e835942d')",
  "name": "kunal",
  "email": "kunal12@gmail.com",
  "password": "$2b5105bddj.ma5/p.62WMyA35PkeJenDV79HgKcJAD0p610CYHUcY6g3XC",
  "phone": "81234567899",
  "address": "subathu",
  "answer": "golf",
  "role": 0,
  "createdAt": "2025-01-26T09:57:53.443+00:00",
  "updatedAt": "2025-01-26T09:57:53.443+00:00",
  "__v": 0
}
```

All Category:

The screenshot shows a web application interface for 'Gen-Z Collection'. The top navigation bar includes 'HOME', 'CATEGORY', 'ADMIN', and 'CART'. The main content area is divided into three sections: 'Admin Panel', 'Manage Category', and 'Categories'.

Admin Panel

- Create Category
- Create Product
- products
- Users

Manage Category

Category Form

Enter row category

Submit

Categories

| Name | Actions |
|----------------------|-------------|
| Men Collections | Edit Delete |
| Women Collections | Edit Delete |
| Children Collections | Edit Delete |

Update and delete product by admin:

Gen-Z Collection HOME CATEGORY ADMIN CART⁰²

Admin Panel

- Create Category
- Create Product
- products
- Users

Update Product

Childrens Collections

Upload Photo



Pant Shoes T-Shirt
Full Package
990
0

No

All Right Reserved © TechinfoKM
About | Contact | Privacy Policy

All product list by admin

Admin Panel

- Create Category
- Create Product
- products
- Users

All Products List



Pant, Shoes, T-Shirt
990
Full Package



Warm Shoes
1000
warmy!



Sharara
2300
cool

© TechinfoKM 2023. All Rights Reserved. (p=0/02)

Add to Cart

Gen-Z Collection

HOME CATEGORY KUNAL CART 2

Hello kunal
You have 3 items in your cart



Pant, Shorts, T-Shirt
Full Package
₹990



Sharara
coat
₹2300



Warm Shoes
wanny!
₹1000

Cart Summary

Total/Checkout/Payment

Total: ₹4,290.00

Current Address

Product Detail

Gen-Z Collection

Product Detail



Product details
Name: Lehnga Choli
Description: deshy
Price: ₹5600
Category: Women Collections

User Dashboard

Gen-Z Collection HOME CATEGORY KUNAL CART

Dashboard

[Profile](#)

[Orders](#)

Profile

Name

Search Result

Gen-Z Collection HOME CATEGORY

Search Result

Found: 1

All Products



pant
\$10
tighty

[More Details](#) | [Add to cart](#)



Coat and Pants
\$45
cool

[More Details](#) | [Add to cart](#)



Pant, Shoes, T-Shirt
\$990
Full Package

[More Details](#) | [Add to cart](#)

Contact Page

Contact Page

Welcome to Gen-G collection ,your premier online destination for products and categories.We are team of passionate individuals who are dedicated to providing you with best possible shopping experience.

Contact Number:93179-89696

All Right Reserved © TechinfoKM

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

About Page

About Page

Welcome to Gen-G collection ,your premier online destination for products and categories.We are team of passionate individuals who are dedicated to providing you with best possible shopping experience.



All Right Reserved © TechinfoKM

Bibliography

YouTube Channel:

- Code with Harry
- Sheryian's coding school

Websites:

- W3schools.org
- Geeks for greeks
- Hackerranks

Software Used:

- Visual Studio Code (developed by **Microsoft**, in **2015**)
- MongoDB Compass (developed by **MongoDB Incorporation**, and released in **2016**).