VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590014

Internship Report on

"FULL STACK WEB DEVELOPMENT"

Submitted in Partial fulfilment of the Requirements for the award of the Degree of

Degree of Bachelor of Engineering in

INFORMATION SCIENCE & ENGINEERING

SUBMITTED BY

DEEPIKA M L (1DB20IS043)

FULL STACK WEB DEVELOPMENT INTERNSHIP AT
PRINSTON SMART ENGINEERS

Under the Guidance of
Mr Abhishek K

DON BOSCO INSTITUTE OF TECHNOLOGY
Kumbalagodu, Mysore Road, Bengaluru-560074

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Project work entitled **“ONLINE WATCH SHOP”** carried out by **DEEPIKA M L (1DB20IS043)** is a bona-fide student at Don Bosco Institute of Technology in partial fulfilment of the requirement of VII semester (Full Stack Web Development Internship Project) Bachelor of Engineering in Information Science and Engineering Visvesvaraya Technological University, Belgaum during the year 2023 – 2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Internship Project report has been approved as it satisfies the academic requirements with respect of the Mini Project work prescribed for the said degree.

Project Guide

Mrs. Yashodara R
Assistant Professor
Dept. of ISE,
DBIT, Bangalore.

Head of Department

Dr. B K Raghavendra
Professor & HoD
Dept. of ISE,
DBIT, Bangalore.

Name of the Examiners **Signature with Date**

1 ______________________ ______________________

2 ______________________ ______________________

ABSTRACT

Web development involves creating and maintaining websites, web applications, and other digital products. It includes designing the user interface, coding the functionality, and integrating various components such as databases, APIs, and web servers. Common languages and technologies used in web development include HTML, CSS, JavaScript, NodeJS, and MongoDB. Web developers need to stay up to date with emerging trends and technologies to deliver high-quality and engaging digital experiences. In recent years, the world has witnessed a significant shift in the way people shop for groceries. With the advent of e-commerce and the increasing popularity of online shopping, more and more people are turning to online watch shops to purchase their gadgets. An online watch shop is a virtual store that allows customers to browse, select, and purchase gadgets and items through the internet. Online watch shopping offers many benefits over traditional brick- and-mortar stores. This report scrutinizes the dynamic realm of online watch stores specializing in the sale of different watches. It dissects the driving forces behind this burgeoning market, highlighting the consumer's escalating appetite for convenient access to premium produce. The report investigates diverse business models, scrutinizes their advantages, and challenges, and assesses their role in enhancing the customer experience. Additionally, it explores the pivotal role of technology in optimizing supply chains and customer satisfaction. Case studies of successful online shopping offers practical insights for market strategies and operational excellence.

ACKNOWLEDGEMENT

While presenting this Full Stack Web Development Project on **"ONLINE WATCH SHOP”,** I feel that it is our duty to acknowledge the help rendered to us by various persons.

I would also like to express our heartfelt gratitude to **Dr. B K Raghavendra,** HOD, Information Science and Engineering whose guidance and support was truly invaluable.

I am very grateful to our guide**, Mrs Yashodara R,** Associate Prof.,Department of Information Science, for her able guidance and valuable advice

I Sincerely Acknowledge Guidance and Constant Encouragement of our Internship Guides

Ms Fraheen Farhath and Mr Abhishesk K PRINSTON SMART ENGINEERS, for their guidance and valuable advice at every stage of our project which helped us in the successful completion of our project.

DEEPIKA M L
(1DB20IS043)

TABLE OF CONTENTS

SL NO. CHAPTERS PAGE NO.

Chapter 1

INTRODUCTION

Web development is the process of creating and maintaining websites, web applications, and other digital products that are accessible through the internet. The field of web development has grown rapidly over the years, and it is now a critical aspect of business operations, marketing, and communication. The internet has become an essential part of our lives, and web development has played a significant role in making it possible.

Web development encompasses various disciplines, including web design, front- end development, back-end development, and web server administration. Web design involves creating the user interface, determining the layout, typography, and colour scheme. Front-end development involves coding the visual components using languages such as HTML, CSS, and JavaScript. Back-end development involves coding the server-side functionality, such as database integration, server-side scripting, and API development using languages like PHP, Python, and Ruby on Rails. Web server administration involves configuring and maintaining web servers and server applications, such as Apache and Nginx.

Web development has seen significant advancements in recent years, with the introduction of technologies such as ReactJS, AngularJS, and Vue JS. These technologies allow for the creation of dynamic and responsive user interfaces that provide a better user experience. Single-page applications (SPAs) are also becoming more popular, which means that the user can interact with the website without the need for a page reload. This improves the speed and responsiveness of the website.

Web development also includes e-commerce development, which is the creation of online stores and shopping carts. E-commerce development involves integrating payment gateways, shopping carts, and other features that facilitate the buying and selling of products and services online.

Mobile-first design is also an important aspect of web development. With the increasing number of mobile users, it is essential to design websites that are mobile-friendly and responsive. This involves optimizing the website for smaller screens and ensuring that the website is easy to navigate on a mobile device.

Web development is a fast-paced and dynamic field, and developers need to stay up to date with emerging technologies and trends. Continuous learning and skills development are essential for web developers to deliver high-quality and engaging digital experiences that meet the needs of the users. Overall, web development is an exciting and rewarding field that plays a critical role in shaping the future of the internet.

ReactJS

ReactJS is an open-source JavaScript library used for building user interfaces. Developed by Facebook, it has become one of the most popular front-end libraries due to its simplicity and flexibility. ReactJS utilizes a component-based architecture, allowing developers to create reusable UI elements that can be easily integrated into larger applications. It also enables the creation of dynamic and responsive interfaces using virtual DOM and JSX syntax. ReactJS has a vast ecosystem of tools and libraries, making it a go-to choose for building modern web applications.

MongoDB

MongoDB is a popular NoSQL database management system that stores data in JSON-like documents with dynamic schemas, rather than traditional table-based structures. This makes MongoDB a flexible and scalable option for handling large volumes of unstructured or semi structured data. It also allows for easy integration with modern web development frameworks such as Node.js and Express.js. MongoDB has a rich query language and supports various data formats, including geospatial data and binary data. Additionally, it offers features like automatic sharing and replication, providing high availability and fault tolerance. MongoDB is used in a wide range of applications, from e- commerce to social media and analytics.

NodeJS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that allows developers to run JavaScript code outside of a web browser. It uses an event-driven, non-blocking I/O model, making it efficient and scalable for building high- performance, real-time applications. Node.js has a vast ecosystem of modules and packages available through the Node Package Manager (NPM), making it easy to add functionality to applications. It is also frequently used with popular front-end frameworks like React.js and Angular.js. Node.js is ideal for building server-side web applications, APIs, and microservices, and it supports various web protocols and frameworks such as Express, Socket.IO etc. It is a popular choice for building scalable and reliable applications, and its lightweight architecture and extensive community support make it a popular choice for developer.

Express JS

Express.js is a crucial component in the MERN stack (MongoDB, Express.js, React, Node.js) and a minimalist web application framework for Node.js. It simplifies the development of web applications and APIs by providing a robust set of tools and features. Express.js allows developers to handle HTTP requests, define routes, manage middleware, and serve static assets efficiently. Its flexibility and scalability make it ideal for building everything from small, single-page applications to large-scale, complex web platforms. With its lightweight nature and extensive ecosystem of middleware, Express.js empowers developers to create robust and high-performance web applications while maintaining code simplicity and flexibility.

HTML

HTML, or Hypertext Markup Language, is the standard markup language used to create and structure content for the web. It is a foundational technology for web development and is used in conjunction with CSS and JavaScript to create interactive and visually appealing web pages. HTML allows developers to structure content with semantic tags, making it more accessible and easier to understand for both humans and machines. With HTML, developers can create headings, paragraphs, lists, links, images, forms, and other elements that make up web pages.

HTML has gone through several versions since its initial creation in the early 1990s, with the latest version being HTML5. HTML5 introduced new semantic elements, audio and video playback, and improved accessibility features. It also includes new APIs for real- time communication, geolocation, and offline web applications. HTML is a crucial technology for web development, and its simplicity and accessibility make it easy for developers of all skill levels to create compelling and accessible web content.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file which reduces complexity and repetition in the structural content as well as enabling the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speechbased browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified.

priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium(W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC2318 (March 1998). TheW3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

LITERATURE SURVEY

Aim of the Project

Aim of an online watch shop is to provide customers with the convenience of ordering gadgets from the comfort of their own homes or workplaces and have them delivered directly to their doorsteps. Some specific goals of the project might include:

- Building a responsive and intuitive user interface that provides user to easily buy the items.
- Integrating with a backend system to order various products real-time.
- Implementing a secure payment gateway to handle online transactions.

Overall, the goal is to save customers time and effort by eliminating the need to physically visit a store and navigate through crowded aisles, checkout lines, and parking lots.

Purpose of the Project

The purpose of the project in an online watch shop using ReactJS is to provide a modern, efficient, and user-friendly platform for buying the products online.

By creating a web application using ReactJS, the project aims to achieve the following:

- Convenience: Online watch shopping allows customers to shop for watches from the comfort of their own homes, saving them time and effort. Customers can easily browse products and place orders online, without having to navigate crowded stores or wait in long checkout lines.
- Accessibility: Online shopping can be accessed from anywhere with an internet connection, making it easier for customers who may have mobility issues or live in remote areas to access a wide selection of products.
- Variety: Online watch shops offer a wide variety of products, often including specialty items that may not be available in local stores. Customers can easily compare prices and select from a variety of brands and options.
- Timesaving: Online watch shopping saves customers time by eliminating the need to travel to and from the store, search for products, and wait in checkout lines.

- Flexibility: Online watch shopping allows customers to place orders at any time of the day or night, providing greater flexibility and convenience.

Overall, the purpose of an online watch shop is to provide customers with a convenient and efficient way to purchase watches, while also offering a wide selection of products and a seamless shopping experience.

Benefits of the project

- Simplicity: The project is easy and simple to use.
- Usability: It is easy to use and implement.
- Flexibility: It is very flexible since it is easy to add new features to it.

Chapter 3

REQUIREMENT SPECIFICATION

The system requirement and specification of our project is as follows:

Hardware requirements

Processor: i5 Core Processor

Clock speed: 2.5GHz

Monitor: 1024 * 768 Resolution Colour

Keyboard: QWERTY

RAM: 1 GB

Motherboard: 845c Intel Motherboard

Backup Media: Floppy/pen drive/Hard disk.

Hard disk: 1 TB HDD

I/O Device: Standard input and output devices.

Software requirements

Backend: Node.js and NPM

Frontend: ReactJS

Connectivity: Express.js

Database: MongoDB.

Text editor: Visual Studio Code

Chapter 4

SOFTWARE DESIGN

Frontend Functionalities

1. Create user by signing up for first time.
2. User can login using credentials if the user exists in database.
3. User can add products of his/her wish to cart.
4. Products and their price are displayed, and user can choose any of them.
5. Once added to cart, user can order products by providing address.
6. User must then provide card details to complete money transaction.

Backend Tasks

1. Created 3 tables named users, products, and cart to store the schema in Users.js, Products.js and orders.js files. 2. Add required products in Product table in mongo dB atlas.

Nodejs MongoDB Connection

1. Express.js is a minimalist and flexible web framework for Node.js.
2. It allows developers to easily create web applications and APIs by providing a set of robust features and middleware.
3. With its simplicity and wide range of plugins, Express.js is a popular choice for building scalable and high-performance server-side.

Chapter 5

IMPLEMENTATION

Frontend

Header.js

```
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("Logout Successfully");
  };
  return (
    <>
      <nav className="navbar navbar-expand-lg bg-body-tertiary">
      <nav className="navbar navbar-expand-lg bg-body-tertiary fixed-top">
        <div className="container-fluid">
          <button
            className="navbar-toggler"
@@ -23,29 +40,93 @@ const Header = () => {
              🛒 Watch Shop
            </Link>
            <ul className="navbar-nav ms-auto mb-2 mb-lg-0">
              <SearchInput />
```

```
<li className="nav-item">

  <NavLink to="/" className="nav-link ">
    Home
  </NavLink>
</li>
<li className="nav-item">
  <NavLink to="/category" className="nav-link ">
    Category
  </NavLink>
</li>
<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 className="nav-item dropdown">
  <Link
    className="nav-link dropdown-toggle"
    to={"/categories"}
    data-bs-toggle="dropdown"
  >
    Categories
  </Link>
  <ul className="dropdown-menu">
    <li>
      <Link className="dropdown-item" to={"/categories"}>
        All Categories
      </Link>
    </li>
    {categories?.map((c) => (
      <li>
        <Link
          className="dropdown-item"
          to={`/category/${c.slug}`}
        >
          {c.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>
  </>
) : (
  <>
    <li className="nav-item dropdown">
      <NavLink
        className="nav-link dropdown-toggle"
        href="#"
        role="button"
        data-bs-toggle="dropdown"
        style={{ border: "none" }}
      >
        {auth?.user?.name}
      </NavLink>
      <ul className="dropdown-menu">
        <li>
          <NavLink
            to={`/dashboard/${
              auth?.user?.role === 1 ? "admin" : "user"
            }`}
            className="dropdown-item"
          >
            Dashboard
          </NavLink>
        </li>
        <li>
          <NavLink
```

```
                 onClick={handleLogout}
                 to="/login"
                 className="dropdown-item"
                >
                 Logout
                </NavLink>
               </li>
              </ul>
             </li>
            </>
           )}
           <li className="nav-item">
            <NavLink to="/cart" className="nav-link">
             Cart (0)
             <Badge count={cart?.length} showZero offset={[10, -5]}>
              Cart
             </Badge>
            </NavLink>
           </li>
          </ul>
```

HomePage.js

```
import React from "react";
import React, { useState, useEffect } from "react";
import { useNavigate } from "react-router-dom";
import { Checkbox, Radio } from "antd";
import { Prices } from "../components/Prices";
import { useCart } from "../context/cart";
import axios from "axios";
import toast from "react-hot-toast";
import Layout from "./../components/Layout/Layout";
import { AiOutlineReload } from "react-icons/ai";
import "../styles/Homepage.css";

const HomePage = () => {
 const navigate = useNavigate();
 const [cart, setCart] = useCart();
 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);


//get all cat
const getAllCategory = async () => {
  try {
    const { data } = await axios.get("/api/v1/category/get-category");
    if (data?.success) {
      setCategories(data?.category);
    }
  } catch (error) {
    console.log(error);
  }
};

useEffect(() => {
  getAllCategory();
  getTotal();
}, []);
//get products
const getAllProducts = async () => {
  try {
    setLoading(true);
    const { data } = await axios.get(`/api/v1/product/product-list/${page}`);
    setLoading(false);
    setProducts(data.products);
  } catch (error) {
    setLoading(false);
    console.log(error);
  }
};

//getTOtal COunt
const getTotal = async () => {
  try {
    const { data } = await axios.get("/api/v1/product/product-count");
    setTotal(data?.total);
  } catch (error) {
    console.log(error);
  }
};

useEffect(() => {
  if (page === 1) return;
  loadMore();
}, [page]);
//load more
const loadMore = async () => {
  try {
    setLoading(true);
    const { data } = await axios.get(`/api/v1/product/product-list/${page}`);
    setLoading(false);
    setProducts([...products, ...data?.products]);
  } catch (error) {
    console.log(error);
    setLoading(false);
  }
};

// filter by cat
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 filterd product
```

```
const filterProduct = async () => {
  try {
    const { data } = await axios.post("/api/v1/product/product-filters", {
      checked,
      radio,
    });
    setProducts(data?.products);
  } catch (error) {
    console.log(error);
  }
};
return (
  <Layout>
    <h1>HomePage</h1>
  <Layout title={"ALl Products  "}>
    {/* banner image */}
    <img
      src="/images/banner.png"
      className="banner-img"
      alt="bannerimage"
      width={"100%"}
    />
    {/* banner image */}
    <div className="container-fluid row mt-3 home-page">
      <div className="col-md-3 filters">
        <h4 className="text-center">Filter By Category</h4>
        <div className="d-flex flex-column">
          {categories?.map((c) => (
            <Checkbox
              key={c._id}
              onChange={(e) => handleFilter(e.target.checked, c._id)}
            >
              {c.name}
            </Checkbox>
          ))}
        </div>
        {/* price filter */}
        <h4 className="text-center mt-4">Filter By Price</h4>
        <div className="d-flex flex-column">
          <Radio.Group onChange={(e) => setRadio(e.target.value)}>
            {Prices?.map((p) => (
              <div key={p._id}>
                <Radio value={p.array}>{p.name}</Radio>
              </div>
            ))}
          </Radio.Group>
        </div>
        <div className="d-flex flex-column">
          <button
            className="btn btn-danger"
            onClick={() => window.location.reload()}
          >
            RESET FILTERS
          </button>
        </div>
      </div>
      <div className="col-md-9 ">
        <h1 className="text-center">All Products</h1>
        <div className="d-flex flex-wrap">
          {products?.map((p) => (
            <div className="card m-2" key={p._id}>
              <img
                src={`/api/v1/product/product-photo/${p._id}`}
                className="card-img-top"
                alt={p.name}
              />
              <div className="card-body">
                <div className="card-name-price">
                  <h5 className="card-title">{p.name}</h5>
                  <h5 className="card-title card-price">
                    {p.price.toLocaleString("en-US", {
                      style: "currency",
                      currency: "USD",
                    })}
                  </h5>
                </div>
                <p className="card-text ">
                  {p.description.substring(0, 60)}...
                </p>
                <div className="card-name-price">
                  <button
                    className="btn btn-info ms-1"
            onClick={() => navigate(`/product/${p.slug}`)}
           >
            More Details
           </button>
           <button
            className="btn btn-dark ms-1"
            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>
      <div className="m-2 p-3">
       {products && products.length < total && (
        <button
         className="btn loadmore"
         onClick={(e) => {
          e.preventDefault();
          setPage(page + 1);
         }}
        >
         {loading ? (
          "Loading ..."
         ) : (
          <>
           {" "}
           Loadmore <AiOutlineReload />
          </>
         )}
        </button>
       )}
```

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

LoginPage.js

```
import React, { useState } from "react";
import Layout from "./../../components/Layout/Layout";
import axios from "axios";
import { useNavigate, useLocation } from "react-router-dom";
import toast from "react-hot-toast";
import "../../styles/AuthStyles.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 function
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {

const res = await axios.post("/api/v1/auth/login", {
        email,
        password,
      });
      if (res && res.data.success) {
        toast.success(res.data && 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="Register – Watch Shop">
      <div className="form-container " style={{ minHeight: "90vh" }}>
        <form onSubmit={handleSubmit}>
          <h4 className="title">LOGIN FORM</h4>

          <div className="mb-3">
            <input
              type="email"
              autoFocus
              value={email}
              onChange={(e) => setEmail(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="Enter Your Email "
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="password"
              value={password}
              onChange={(e) => setPassword(e.target.value)}
              className="form-control"
              id="exampleInputPassword1"
              placeholder="Enter Your Password"
              required
            />
          </div>
          <div className="mb-3">
            <button
              type="button"
              className="btn forgot-btn"
              onClick={() => {
                navigate("/forgot-password");
              }}
            >
              Forgot Password
            </button>
          </div>

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

export default Login;
```

Orders.js

```
import React, { useState, useEffect } from "react";
import UserMenu from "../../components/Layout/UserMenu";
import Layout from "./../../components/Layout/Layout";
import axios from "axios";
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("/api/v1/auth/orders");
      setOrders(data);
    } catch (error) {
      console.log(error);
    }
  };

  useEffect(() => {
    if (auth?.token) getOrders();
  }, [auth?.token]);
  return (
<Layout title={"Your Orders"}>
  <div className="container-flui p-3 m-3 dashboard">
    <div className="row">
      <div className="col-md-3">
        <UserMenu />
      </div>
      <div className="col-md-9">
        <h1 className="text-center">All Orders</h1>
        {orders?.map((o, i) => {
          return (
            <div className="border shadow">
              <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>
                    <td>{i + 1}</td>
                    <td>{o?.status}</td>
                    <td>{o?.buyer?.name}</td>
                    <td>{moment(o?.createAt).fromNow()}</td>
                    <td>{o?.payment.success ? "Success" : "Failed"}</td>
                    <td>{o?.products?.length}</td>
                  </tr>
                </tbody>
              </table>
              <div className="container">
                {o?.products?.map((p, i) => (
                  <div className="row mb-2 p-3 card flex-row" key={p._id}>
                    <div className="col-md-4">
                      <img
                        src={`/api/v1/product/product-photo/${p._id}`}
                        className="card-img-top"
                        alt={p.name}
                        width="100px"
                        height={"100px"}
                      />
                    </div>
                    <div className="col-md-8">
                      <p>{p.name}</p>
                      <p>{p.description.substring(0, 30)}</p>
                      <p>Price : {p.price}</p>
                    </div>
                  </div>
                ))}
              </div>
            </div>
          );
        })}
      </div>
    </div>
  </div>
</Layout>
);
};

export default Orders;
```

RegisterPage.js

```
import React, { useState } from "react";
import Layout from "../../components/Layout/Layout";
import axios from "axios";
import { useNavigate } from "react-router-dom";
import toast from "react-hot-toast";
import "../../Styles/Authstyles.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  function
  const handleSubmit = async (e) => {
    // e means event
    e.preventDefault();
    try {
      const res = await axios.post("/api/v1/auth/register", {
        name,
        email,
        password,
        phone,
        address,
        answer,
      });
      if (res && 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 – Watch Shop">
      <div className="form-container">
        <form onSubmit={handleSubmit}>
          <h4 className="titlt">REGISTER FORM</h4>
          <div className="mb-3">
            <input
              type="text"
              value={name}
              onChange={(e) => setName(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="Enter Your Name"
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="email"
              value={email}
              onChange={(e) => setEmail(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="Enter Your Email"
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="password"
              value={password}
              onChange={(e) => setPassword(e.target.value)}
              className="form-control"
              placeholder="Enter Your Password"
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="text"
              value={phone}
              onChange={(e) => setPhone(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="Enter Your Phone"
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="text"
              value={address}
              onChange={(e) => setAddress(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="Enter Your Address"
              required
            />
          </div>
          <div className="mb-3">
            <input
              type="text"
              value={answer}
              onChange={(e) => setAnswer(e.target.value)}
              className="form-control"
              id="exampleInputEmail1"
              placeholder="What is your Favorite sports"
              required
```

```
            />
          </div>
          <div className="mb-3 form-check"></div>
          <button type="submit" className="btn btn-primary">
            Register
          </button>
        </form>
      </div>
    </Layout>
  );
};

export default Register;
```

Index.js

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

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

Products.js

```
import React, { useState, useEffect } from "react";
import AdminMenu from "../../components/Layout/AdminMenu";
import Layout from "./../../components/Layout/Layout";
import axios from "axios";
import toast from "react-hot-toast";
import { Link } from "react-router-dom";
const Products = () => {
const [products, setProducts] = useState([]);

  //getall products
  const getAllProducts = async () => {
    try {
      const { data } = await axios.get("/api/v1/product/get-product");
      setProducts(data.products);
    } catch (error) {
      console.log(error);
      toast.error("Someething Went Wrong");
    }
  };

  //lifecycle method
  useEffect(() => {
    getAllProducts();
  }, []);
  return (
    <Layout>
      <div className="row dashboard">
        <div className="col-md-3">
          <AdminMenu />
        </div>
        <div className="col-md-9 ">
          <h1 className="text-center">All Products List</h1>
          <div className="d-flex flex-wrap">
            {products?.map((p) => (
              <Link
                key={p._id}
                to={`/dashboard/admin/product/${p.slug}`}
                className="product-link"
              >
                <div className="card m-2" style={{ width: "18rem" }}>
                  <img
                    src={`/api/v1/product/product-photo/${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>
                  </div>
                </div>
              </Link>
            ))}
          </div>
        </div>
      </div>
    </Layout>
  );
};

export default Products;
```

. env frontend

```
REACT_APP_API = http://localhost:8000
REACT_APP_ADMIN_EMAIL = deepika246@gmail.com
REACT_APP_PUBLIC_KEY = 9t965w989yrxxp5f
```

Cart.js

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

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

  //total price
  const totalPrice = () => {
    try {
      let total = 0;
      cart?.map((item) => {
        total = total + item.price;
      });
      return total.toLocaleString("en-US", {
        style: "currency",
        currency: "USD",
      });
    } catch (error) {
      console.log(error);
    }
  };
  //detele 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 gateway token
  const getToken = async () => {
    try {
      const { data } = await
axios.get("/api/v1/product/braintree/token");
      setClientToken(data?.clientToken);
    } catch (error) {
      console.log(error);
    }
  };
  useEffect(() => {
    getToken();
  }, [auth?.token]);

  //handle payments
  const handlePayment = async () => {
    try {
      setLoading(true);
      const { nonce } = await
instance.requestPaymentMethod();
      const { data } = await
axios.post("/api/v1/product/braintree/payment", {
        nonce,
        cart,
      });
      setLoading(false);
      localStorage.removeItem("cart");
      setCart([]);
      navigate("/dashboard/user/orders");
      toast.success("Payment Completed Successfully
");
    } catch (error) {
      console.log(error);
      setLoading(false);
    }
  };
  return (
<Layout>
      <div className=" cart-page">
        <div className="row">
          <div className="col-md-12">
            <h1 className="text-center bg-light p-2 mb-
1">
              {!auth?.user
                ? "Hello Guest"
                : `Hello  ${auth?.token &&
auth?.user?.name}`}
              <p className="text-center">
                {cart?.length
                  ? `You Have ${cart.length} items in your
cart ${
                      auth?.token ? "" : "please login to
checkout !"
                    }`
                  : " Your Cart Is Empty"}
              </p>
            </h1>
          </div>
        </div>
        <div className="container ">
          <div className="row ">
            <div className="col-md-7  p-0 m-0">
              {cart?.map((p) => (
                <div className="row card flex-row"
key={p._id}>
                  <div className="col-md-4">
                    <img
                      src={`/api/v1/product/product-
photo/${p._id}`}
                      className="card-img-top"
                      alt={p.name}
                      width="100%"
                      height={"130px"}
                    />
                  </div>
                  <div className="col-md-4">
                    <p>{p.name}</p>
                    <p>{p.description.substring(0, 30)}</p>
                    <p>Price : {p.price}</p>
                  </div>
                  <div className="col-md-4 cart-remove-
```

```
btn">
                    <button
                      className="btn btn-danger"
                      onClick={() =>
removeCartItem(p._id)}
                    >
                      Remove
                    </button>
                  </div>
                </div>
              ))}
            </div>
            <div className="col-md-5 cart-summary ">
              <h2>Cart Summary</h2>
              <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",
            })
           }
          >
           Plase Login to checkout
          </button>
         )}
        </div>
       )}
       <div className="mt-2">
        {!clientToken || !auth?.token || !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 ...." : "Make
Payment"}
          </button>
         </>
        )}
       </div>
      </div>
     </div>
    </div>
   </div>
```

```
</Layout>
  );
};

export default CartPage;
```

ProductDetails.js

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

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

  //initalp details
  useEffect(() => {
    if (params?.slug) getProduct();
  }, [params?.slug]);
  //getProduct
  const getProduct = async () => {
    try {
      const { data } = await axios.get(
        `/api/v1/product/get-product/${params.slug}`
      );
      setProduct(data?.product);
      getSimilarProduct(data?.product._id, data?.product.category._id);
    } catch (error) {
      console.log(error);
    }
  };
  //get similar product
  const getSimilarProduct = async (pid, cid) => {
    try {
      const { data } = await axios.get(
        `/api/v1/product/related-product/${pid}/${cid}`
      );
      setRelatedProducts(data?.products);
    }
catch (error) {
      console.log(error);
    }
  };
  return (
    <Layout>
      <div className="row container product-details">
        <div className="col-md-6">
          <img
            src={`/api/v1/product/product-photo/${product._id}`}
            className="card-img-top"
            alt={product.name}
            height="300"
            width={"350px"}
          />
        </div>
        <div className="col-md-6 product-details-info">
          <h1 className="text-center">Product Details</h1>
          <hr />
          <h6>Name : {product.name}</h6>
          <h6>Description : {product.description}</h6>
          <h6>
            Price :
            {product?.price?.toLocaleString("en-US", {
              style: "currency",
              currency: "USD",
            })}
          </h6>
          <h6>Category : {product?.category?.name}</h6>
          <button class="btn btn-secondary ms-1">ADD TO CART</button>
        </div>
      </div>
      <hr />
      <div className="row container similar-products">
        <h4>Similar Products ➡️</h4>
        {relatedProducts.length < 1 && (
          <p className="text-center">No Similar Products found</p>
        )}
        <div className="d-flex flex-wrap">
          {relatedProducts?.map((p) => (
            <div className="card m-2" key={p._id}>
              <img
                src={`/api/v1/product/product-photo/${p._id}`}
                className="card-img-top"
alt={p.name}
              />
              <div className="card-body">
                <div className="card-name-price">
                  <h5 className="card-title">{p.name}</h5>
                  <h5 className="card-title card-price">
                    {p.price.toLocaleString("en-US", {
                      style: "currency",
                      currency: "USD",
                    })}
                  </h5>
                </div>
                <p className="card-text ">
                  {p.description.substring(0, 60)}...
                </p>
                <div className="card-name-price">
                  <button
                    className="btn btn-info ms-1"
                    onClick={() => navigate(`/product/${p.slug}`)}
                  >
                    More Details
                  </button>
                  {/* <button
                  className="btn btn-dark ms-1"
                  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>
      </div>
    </Layout>
  );
};
```

```
export default ProductDetails;
```

ProductCategoryController.js

```
 import productModel from "../models/productModel.js";
import categoryModel from "../models/categoryModel.js";
import orderModel from "../models/orderModel.js";
import fs from "fs";
import slugify from "slugify";
import braintree from "braintree";
import dotenv from "dotenv";

dotenv.config();
//payment gateway

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,
});

export const createProductController = async (req, res) => {
  try {
    const { name, 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(201).send({
      success: true,
      message: "Product Created Successfully",
      products,
    });
  } catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      error,
      message: "Error in crearing product",
    });
  }
};

//get all products
export const getProductController = async (req, res) => {
  try {
    const products = await productModel
      .find({})
      .populate("category")
      .select("-photo")
      .limit(12)
      .sort({ createdAt: -1 });
    res.status(200).send({
      success: true,
      counTotal: products.length,
      message: "ALL Products ",
      products,
    });
  } catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      message: "Erorr in getting products",
      error: error.message,
    });
  }
};

// get single product
export 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 Fetched",
      product,
    });
  } catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      message: "Eror while getitng single product",
      error,
    });
  }
};

// get photo
export 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: "Erorr while getting photo",
      error,
    });
  }
};

//delete controller
export const deleteProductController = async (req, res) => {
  try {
    await productModel.findByIdAndDelete(req.params.pid).select("-photo");
    res.status(200).send({
      success: true,
      message: "Product Deleted successfully",
    });
  } catch (error) {
    console.log(error);
```

```
res.status(500).send({
      success: false,
      message: "Error while deleting product",
      error,
    });
  }
};

//upate product
export const updateProductController = async (req, res) => {
  try {
    const { name, 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 = 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(201).send({
      success: true,
      message: "Product Updated Successfully",
      products,
    });
  }
catch (error) {
    console.log(error);
    res.status(500).send({
      success: false,
      error,
      message: "Error in Update product",
    });
  }
};

//filters
export const productFilterController = 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,
      products,
    });
  } catch (error) {
    console.log(error);
    res.status(400).send({
      success: false,
      message: "Error WHile Filtering Products",
      error,
    });
  }
};

// product count
export const productCountController = async (req, res) => {
  try {
    const total = await productModel.find({}).estimatedDocumentCount();
    res.status(200).send({
      success: true,
      total,
    });
  } catch (error) {
    console.log(error);
    res.status(400).send({
      message: "Error in product count",
      error,
      success: false,
    });
  }
```

```
};

// product list base on page
export 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,
      products,
    });
  } catch (error) {
    console.log(error);
    res.status(400).send({
      success: false,
      message: "error in per page ctrl",
      error,
    });
  }
};

// search product

export const searchProductController = async (req, res) => {
  try {
    const { keyword } = req.params;
    const resutls = await productModel
      .find({
        $or: [
          { name: { $regex: keyword, $options: "i" } },
          { description: { $regex: keyword, $options: "i" } },
        ],
      })
      .select("-photo");
    res.json(resutls);
  } catch (error) {
    console.log(error);
    res.status(400).send({
      success: false,
      message: "Error In Search Product API",
      error,
    });
  }
```

```
};
// similar products

export const realtedProductController = 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(400).send({
      success: false,
      message: "error while geting related product",
      error,
    });
  }
};

// get products by catgory
export 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,
      error,
      message: "Error While Getting products",
    });
  }

};
```

```
//payment gateway api
//token
export 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

export const brainTreePaymentController = async (req, res) => {
  try {
    const { nonce, cart } = req.body;
    let total = 0;
    cart.map((i) => {
      total += i.price;
    });
    let newTransaction = 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);
  }
};
```

CategoryProduct.js

```
import React, { useState, useEffect } from
"react";
import Layout from
"../components/Layout/Layout";
import { useParams, useNavigate } from
"react-router-dom";
import "../Styles/CategoryProductStyles.css";
import axios from "axios";
const CategoryProduct = () => {
  const params = useParams();
  const navigate = useNavigate();
  const [products, setProducts] = useState([]);
  const [category, setCategory] = useState([]);

  useEffect(() => {
    if (params?.slug) getPrductsByCat();
  }, [params?.slug]);
  const getPrductsByCat = async () => {
    try {
      const { data } = await axios.get(
        `/api/v1/product/product-
category/${params.slug}`
      );
      setProducts(data?.products);
      setCategory(data?.category);
    } catch (error) {
      console.log(error);
    }
  };

  return (
    <Layout>
      <div className="container mt-3
category">
        <h4 className="text-center">Category
- {category?.name}</h4>
```

```
<h6
className="text-center">{products?.length}
result found </h6>
        <div className="row">
          <div className="col-md-9 offset-1">
            <div className="d-flex flex-wrap">
              {products?.map((p) => (
                <div className="card m-3"
key={p._id}>
                  <img
                    src={`/api/v1/product/product-
photo/${p._id}`}
                    className="card-img-top"
                    alt={p.name}
                  />
                  <div className="card-body">
                    <div className="card-name-
price">
                      <h5 className="card-
title">{p.name}</h5>
                      <h5 className="card-title
card-price">
                        {p.price.toLocaleString("en-
US", {
                          style: "currency",
                          currency: "USD",
                        })}
                      </h5>
                    </div>
                    <p className="card-text ">
                      {p.description.substring(0,
60)}...
                    </p>
                    <div className="card-name-
price">
                      <button
                        className="btn btn-info ms-
1"
                        onClick={() =>
navigate(`/product/${p.slug}`)}
                      >
                        More Details
                      </button>
                      {/* <button
                    className="btn btn-dark ms-1"
                    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>
            {/* <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 CategoryProduct;
```

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 { SearchProvider } from
"./context/Search";
import { CartProvider } from
"./context/cart";
import "antd/dist/reset.css";
const root =
ReactDOM.createRoot(document.getEle
mentById("root"));
root.render(
  <AuthProvider>
    <SearchProvider>
      <CartProvider>
        <BrowserRouter>
          <App />
<React.StrictMode></React.StrictMode>
        </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();
```

. ENV BACKEND

```
 PORT = 8000
DEV_MODE = development
MONGO_URL = mongodb+srv://deepika246:deepika246@cluster0.qd1qur5.mongodb.net/watch-shop
JWT_SECRET = HAHDWJDK123445
BRAINTREE_MERCHANT_ID = q7f7vrkz94x5cvj5
BRAINTREE_PUBLIC_KEY = 9t965w989yrxxp5f
BRAINTREE_PRIVATE_KEY = c123588bbfa4e3ff0ec8166818b72eaf
```

Chapter 6

SNAPSHOTS

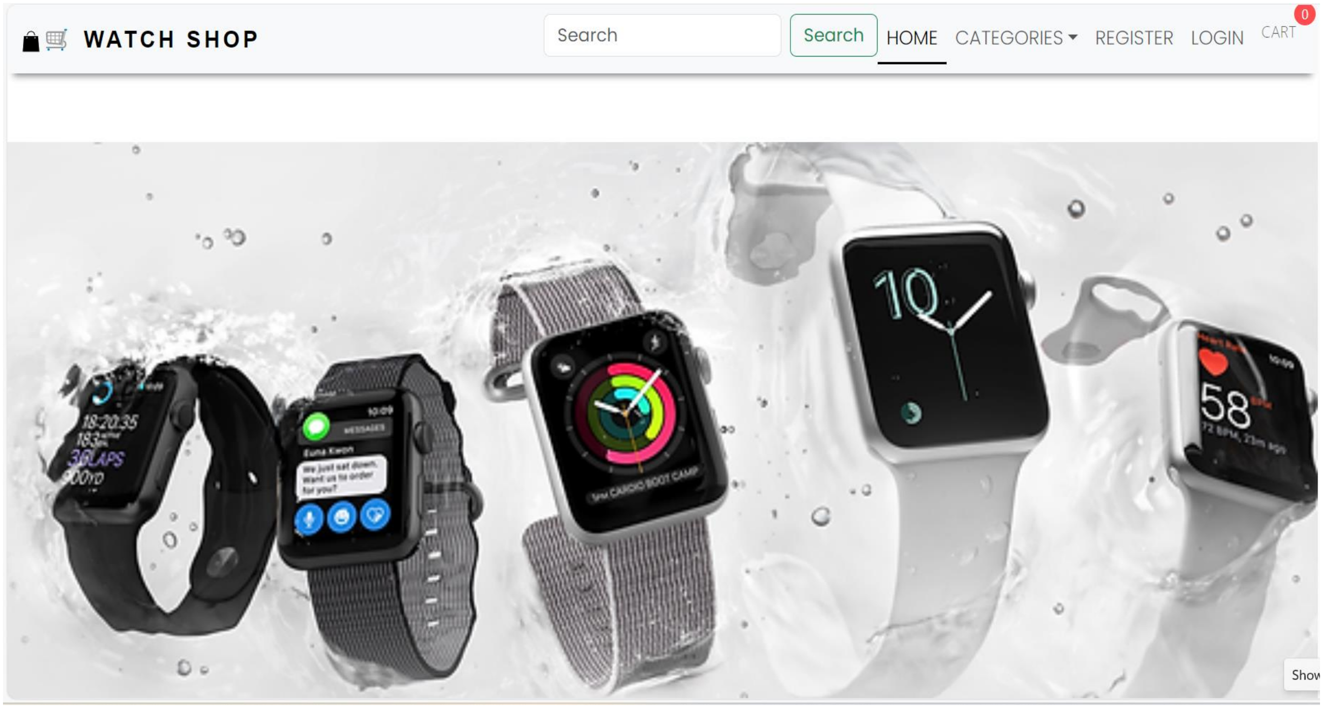
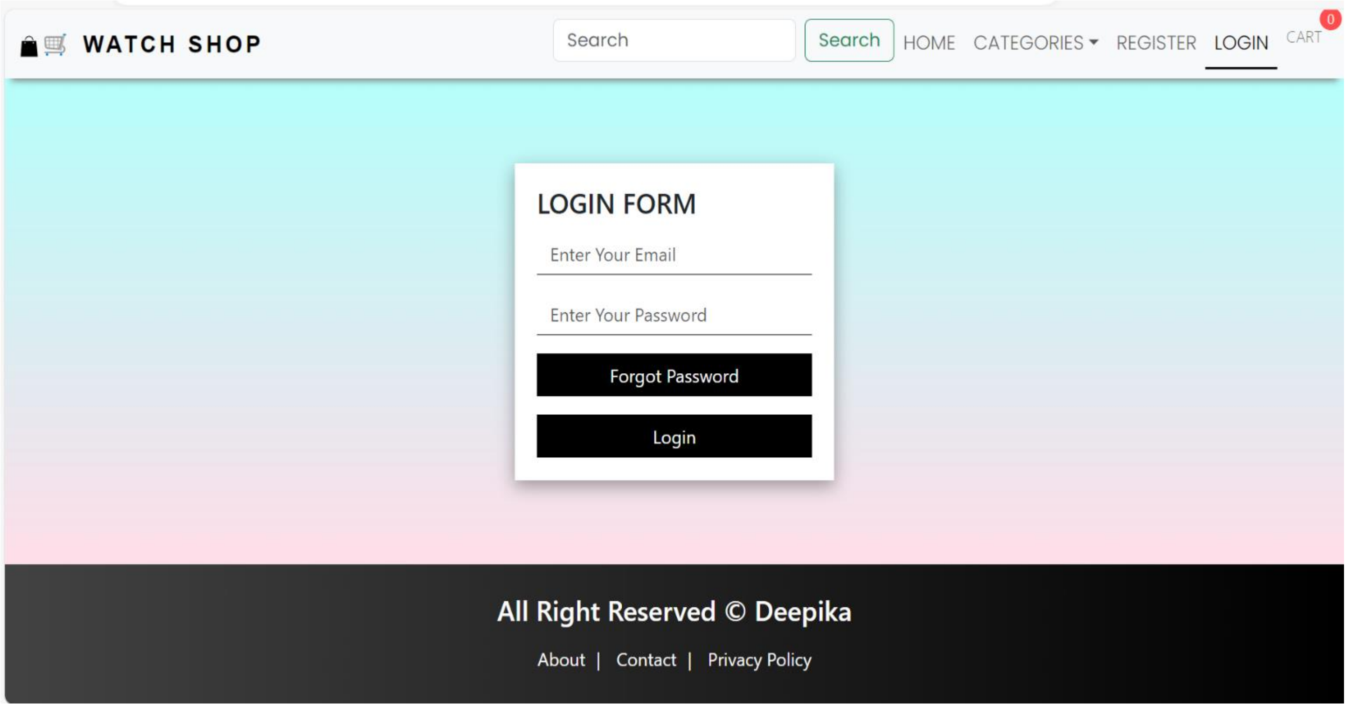


Figure 6.1: Home Page



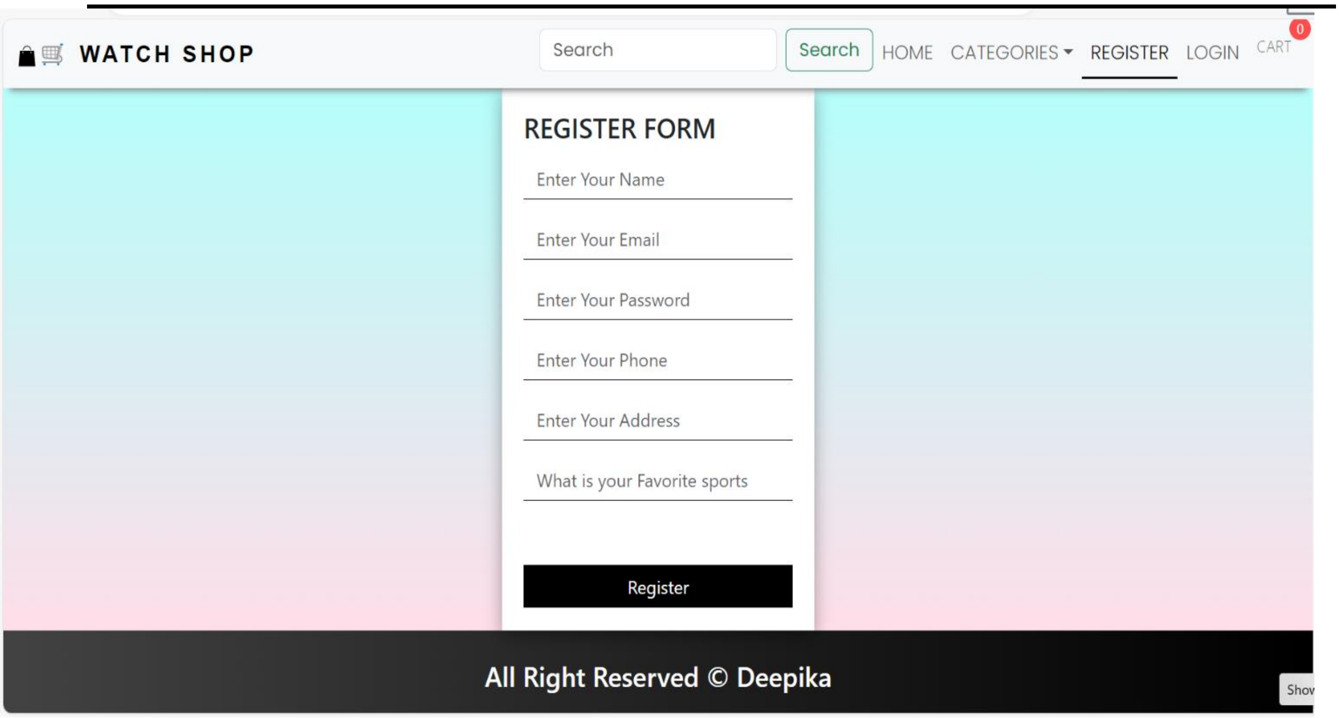
LOGIN FORM

Enter Your Email

Enter Your Password

All Right Reserved © Deepika

Figure 6.2: Login Page



REGISTER FORM

Enter Your Name

Enter Your Email

Enter Your Password

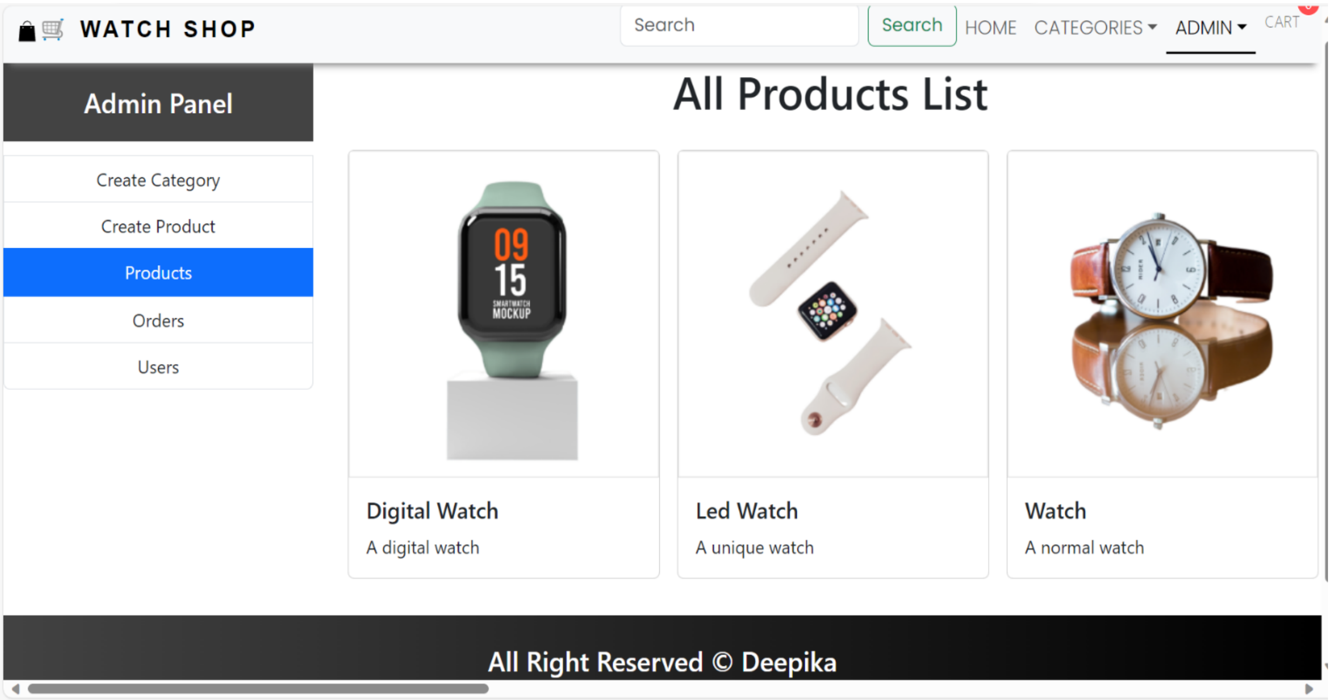
Enter Your Phone

Enter Your Address

What is your Favorite sports

All Right Reserved © Deepika

Figure 6.3: Register Page



All Products List

Digital Watch

A digital watch

Led Watch

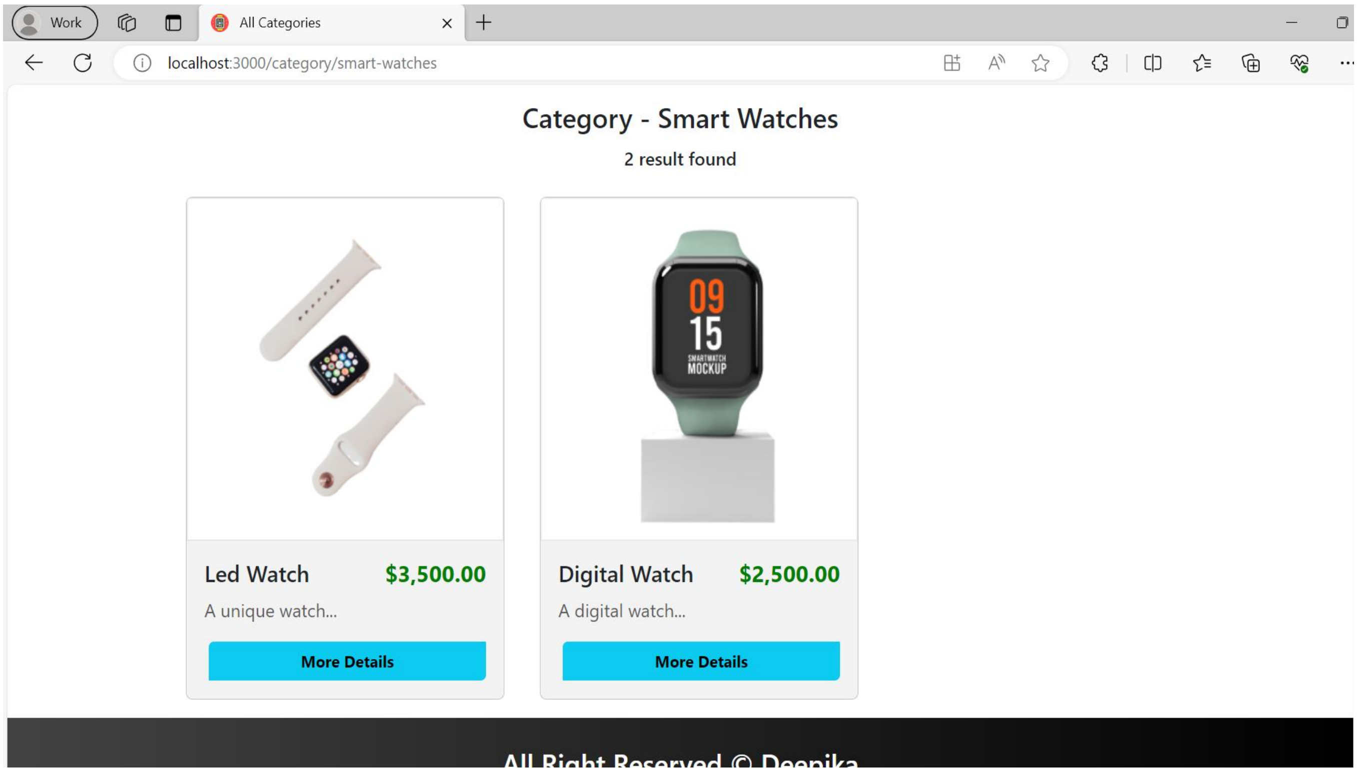
A unique watch

Watch

A normal watch

All Right Reserved © Deepika

Figure 6.4: Products Page



Category - Smart Watches

2 result found

Led Watch $3,500.00

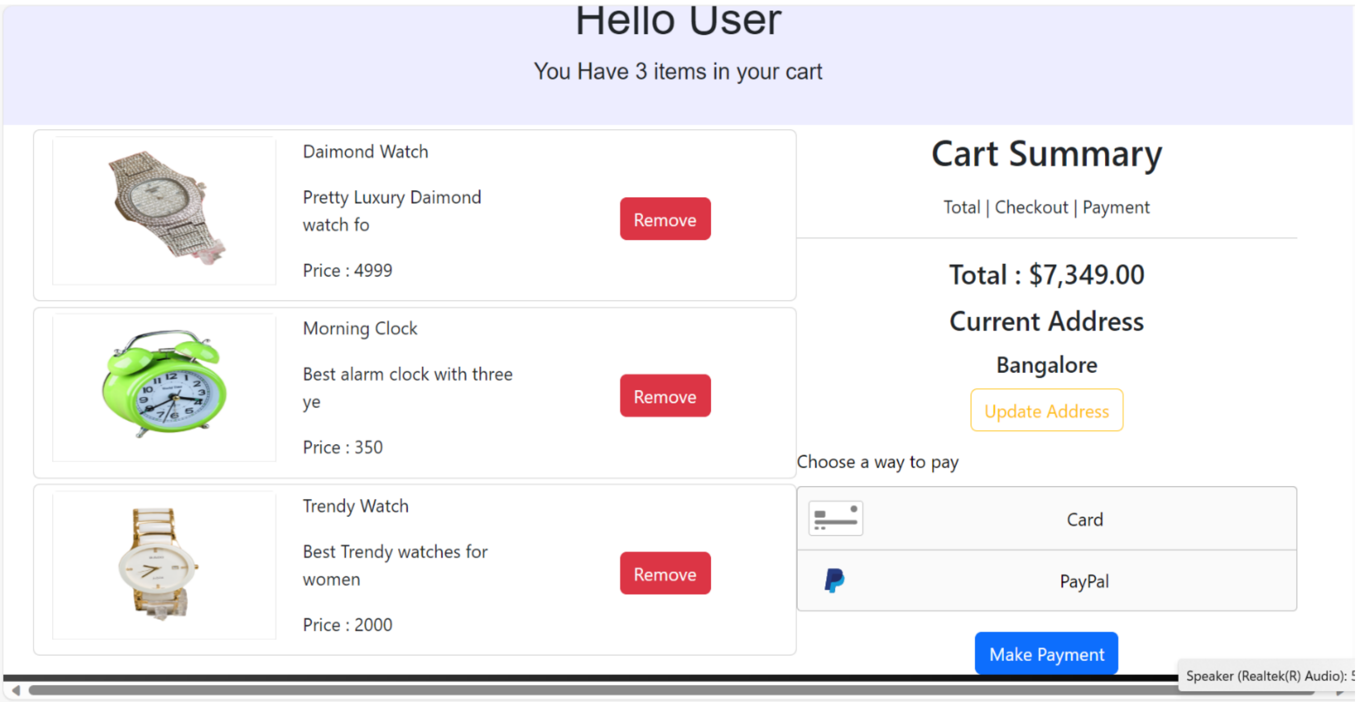
A unique watch...

Digital Watch $2,500.00

A digital watch...

All Right Reserved © Deepika

Figure 6.5: Category Page



Hello User

You Have 3 items in your cart

Daimond Watch

Pretty Luxury Daimond watch fo

Price : 4999

Morning Clock

Best alarm clock with three ye

Price : 350

Trendy Watch

Best Trendy watches for women

Price : 2000

Cart Summary

Total | Checkout | Payment

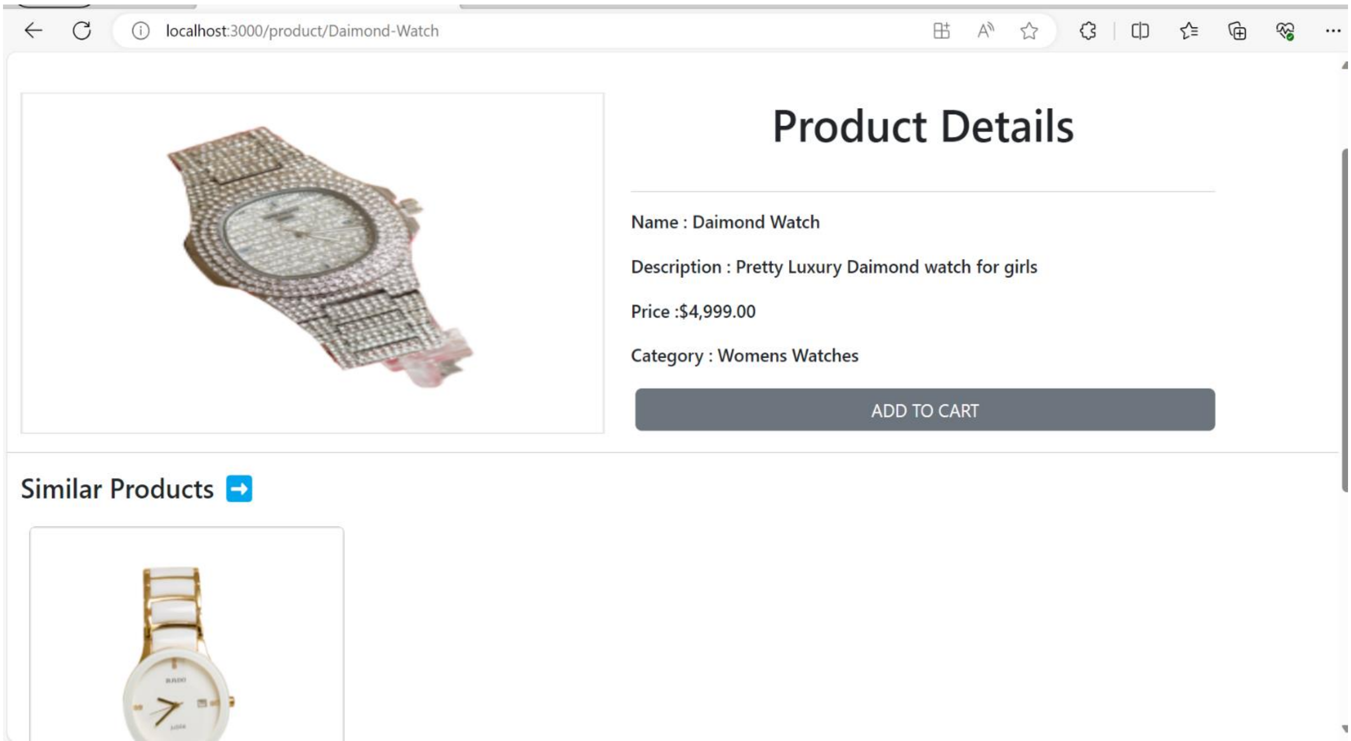
Total : $7,349.00

Current Address

Bangalore

Choose a way to pay

Figure 6.6: Shopping Cart Page



localhost:3000/product/Daimond-Watch

Product Details

Name : Daimond Watch

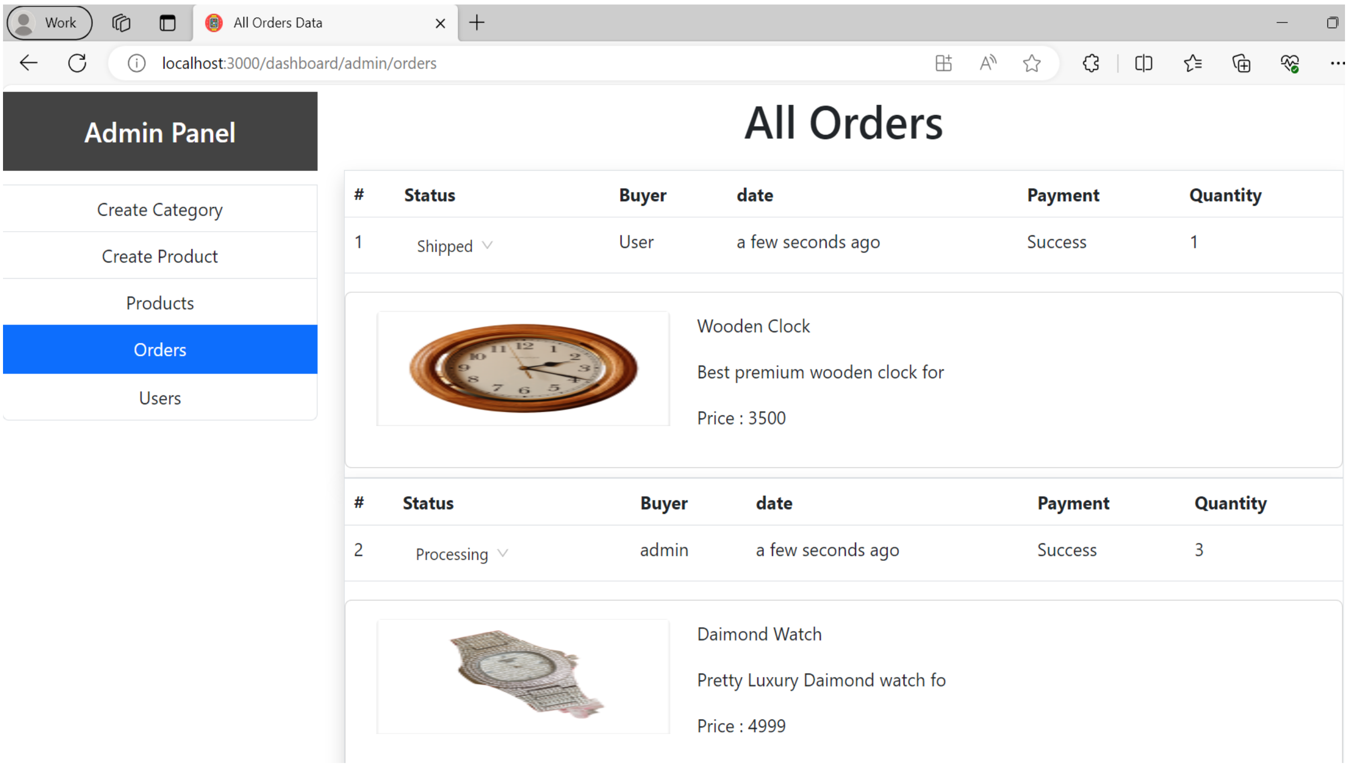
Description : Pretty Luxury Daimond watch for girls

Price :$4,999.00

Category : Womens Watches

Similar Products

Figure 6.7: New Product Page



localhost:3000/dashboard/admin/orders

All Orders

#	Status	Buyer	date	Payment	Quantity
1	Shipped	User	a few seconds ago	Success	1

Wooden Clock

Best premium wooden clock for

Price : 3500

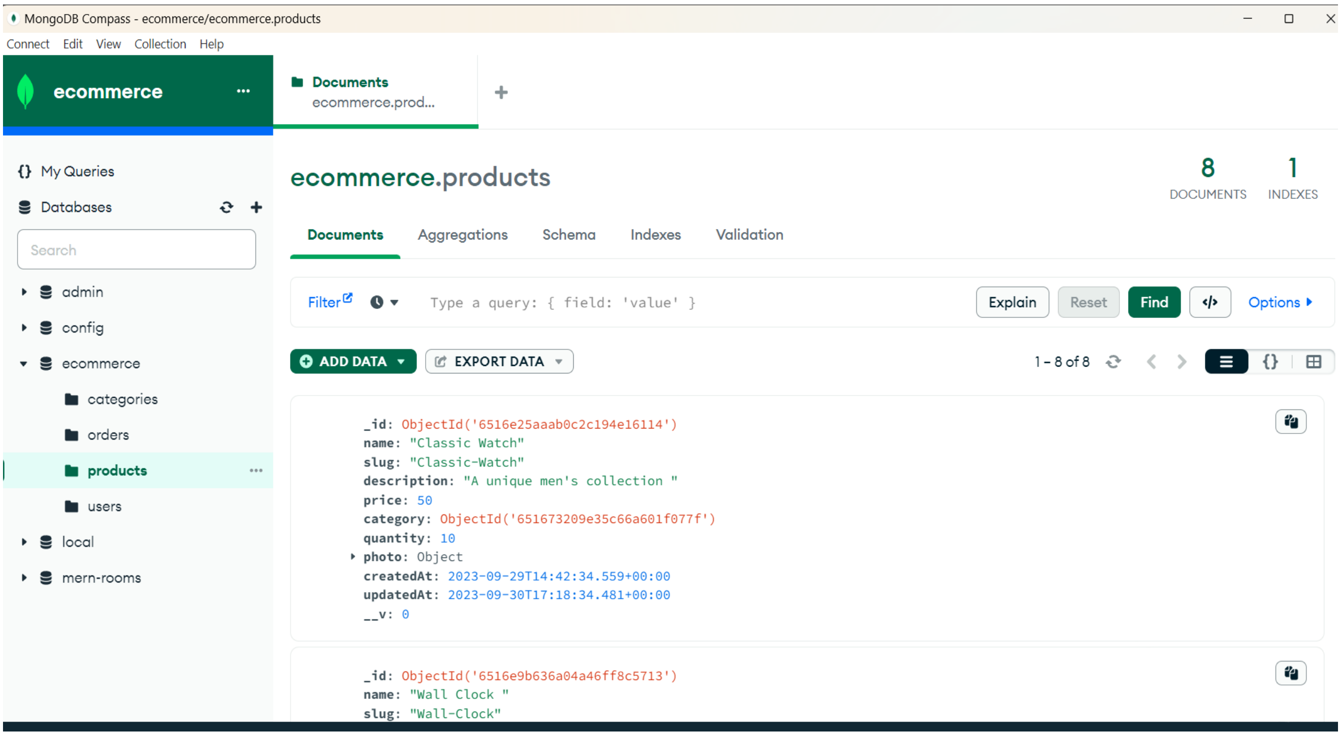
#	Status	Buyer	date	Payment	Quantity
2	Processing	admin	a few seconds ago	Success	3

Daimond Watch

Pretty Luxury Daimond watch fo

Price : 4999

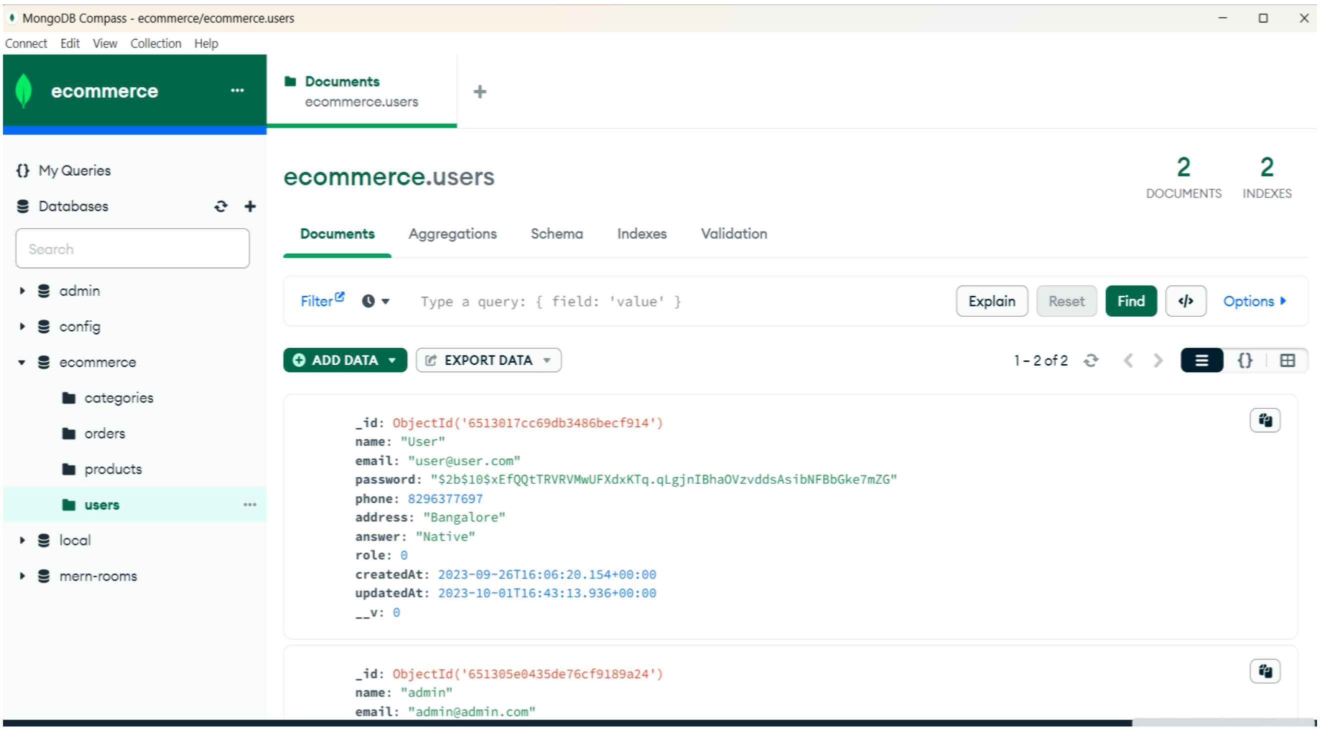
Figure 6.8: Order details Page



```
_id: ObjectId('6516e25aaab0c2c194e16114')
name: "Classic Watch"
slug: "Classic-Watch"
description: "A unique men's collection "
price: 50
category: ObjectId('651673209e35c66a601f077f')
quantity: 10
photo: Object
createdAt: 2023-09-29T14:42:34.559+00:00
updatedAt: 2023-09-30T17:18:34.481+00:00
__v: 0
```

```
_id: ObjectId('6516e9b636a04a46ff8c5713')
name: "Wall Clock "
slug: "Wall-Clock"
```

Figure 6.10: Products Database Page



```
_id: ObjectId('6513017cc69db3486becf914')
name: "User"
email: "user@user.com"
password: "$2b$10$xEfQQtTRVRVMwUFXdxKTq.qLgjnIBhaOVzvddsAsibNFBbGke7mZG"
phone: 8296377697
address: "Bangalore"
answer: "Native"
role: 0
createdAt: 2023-09-26T16:06:20.154+00:00
updatedAt: 2023-10-01T16:43:13.936+00:00
__v: 0
```

```
_id: ObjectId('651305e0435de76cf9189a24')
name: "admin"
email: "admin@admin.com"
```

Figure 6.11: Users Database Page

DECLARATION

We students of 7th semester BE, Information Science and Engineering department, Don Bosco Institute of Technology hereby declare that internship project work entitled "**Online Watch Shop**" has been carried out by us at Prinston Smart Engineers, Bengaluru and submitted in partial fulfilment of the course requirement for the award of the degree of **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belgaum,** during the academic year 2023-2024.We also declare that, to the best of our knowledge and belief, the work reported here does not from the part of dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Date: 05/10/2023

Place: Bangalore

Deepika M L [1DB20IS043]