EasyShop

A React.js E-commerce Application

Submitted By,

Deependra M B

8606882759

ABSTRACT

The EasyShop E-commerce project is a React.js application designed and implemented to showcase fundamental skills in web development. The application focuses on creating an intuitive and user-friendly online shopping experience. It incorporates features such as user authentication, product listing, search functionality, and a shopping cart. This report outlines the key aspects of the project, the steps involved in running it, and provides a conclusion on the overall development experience.

Running Steps

Step1 : git clone https://github.com/DeependraMB/EasyShop---Ecommerce.git

Step2 : cd EasyShop---Ecommerce

Step3 : npm install

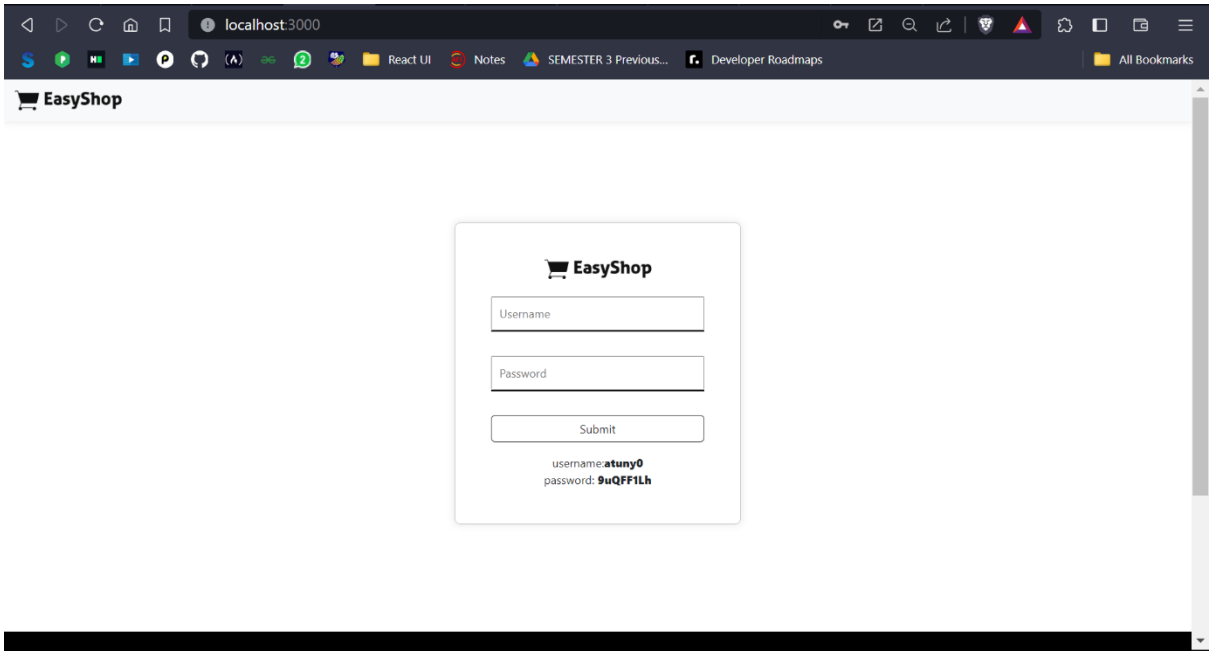
Step4 : npm start

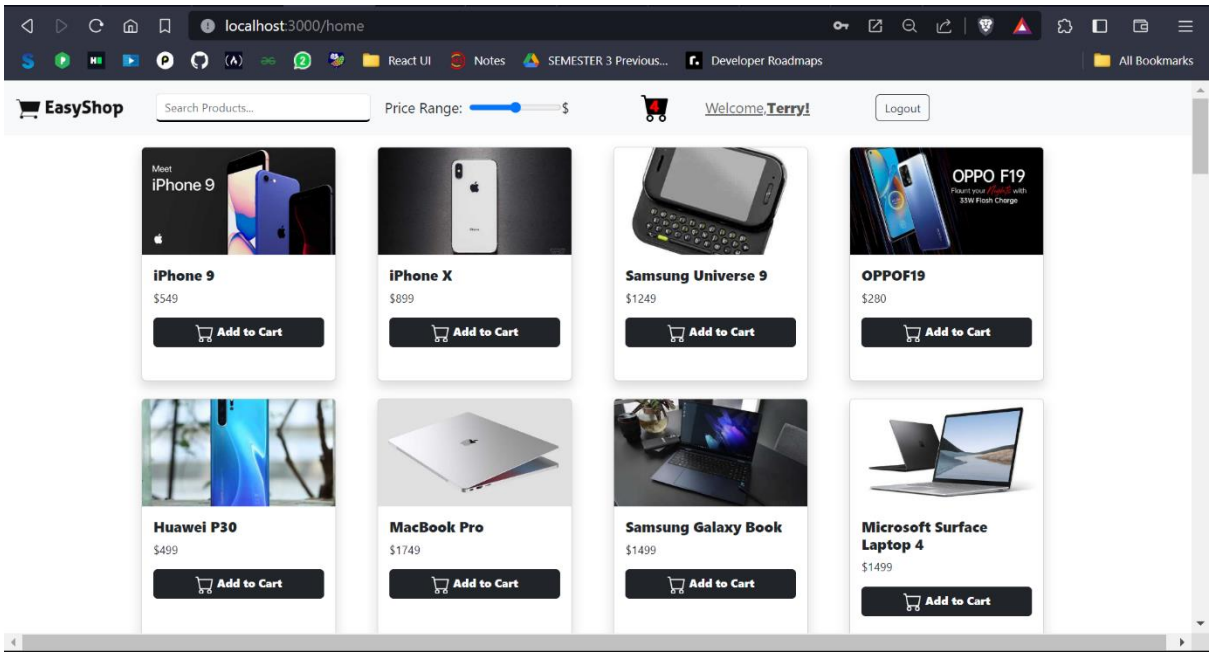
Step 5 : Open your web browser and visit http://localhost:3000 to explore the EasyShop E-commerce application.

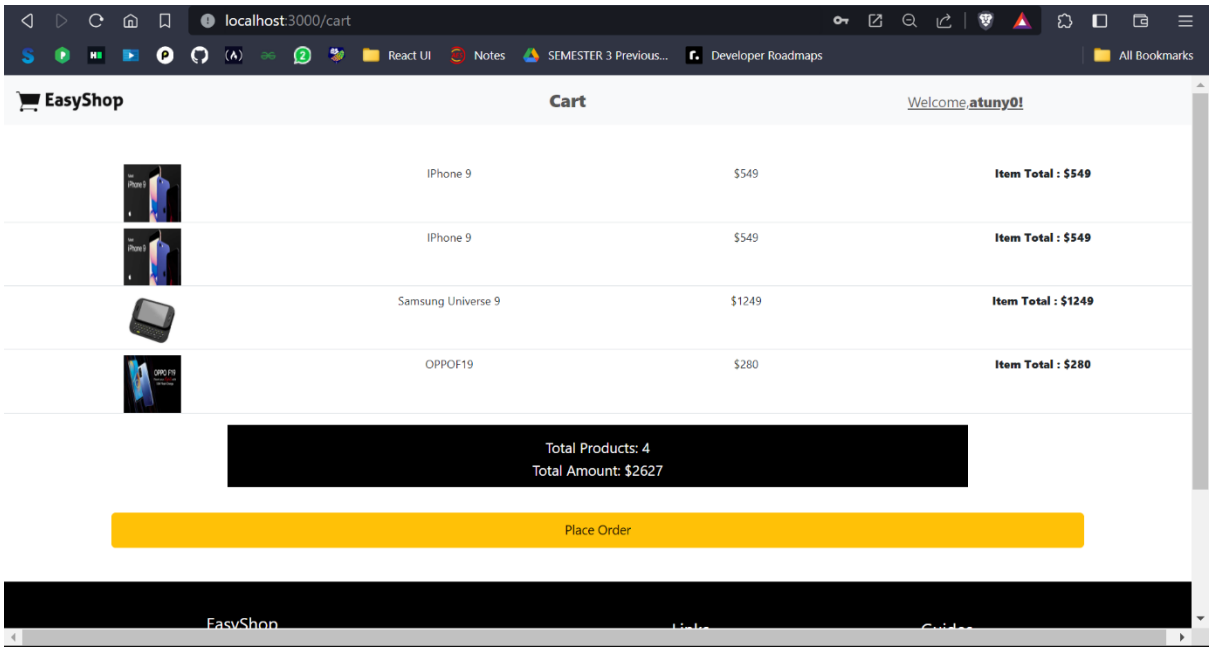
CONCLUSION

In conclusion, the EasyShop project has been an enriching experience in developing a React.js-based e-commerce application. The implementation of key features, such as user authentication, product listing, and shopping cart functionality, has deepened my understanding of front-end web development. The challenges faced during the development process have enhanced problem-solving skills, and the project serves as a valuable addition to my portfolio.

SCREENSHOTS







SAMPLE CODE

```
import "bootstrap/dist/css/bootstrap.css";
import {
  Navigate,
  Route,
  BrowserRouter as Router,
  Routes,
} from "react-router-dom";
import LoginPage from "./Pages/LoginPage";
import HomePage from "./Pages/HomePage";
import CartPage from "./Pages/CartPage";
import { useEffect, useState } from "react";
import { CartProvider } from "./Context/CartContext";

function App() {
  const [token, setToken] = useState("");
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const checkAuthentication = async () => {
      try {
        const authToken = localStorage.getItem("auth");

        if (authToken) {
          await new Promise((resolve) => setTimeout(resolve, 1000));
          setToken(authToken);
        }
      } finally {
```

```
        setLoading(false);
      }
    };

    checkAuthentication();
  }, []);

  if (loading) {
    return <div>Loading...</div>;
  }

  const isAuthenticated = Boolean(token);

  return (
    <div className="App">
      <CartProvider>
        <Router>
          <Routes>
            <Route path="/" element={<LoginPage />} />
            <Route
              path="/home"
              element={isAuthenticated ? <HomePage /> : <Navigate to="/" />}
            />
            <Route
              path="/cart"
              element={isAuthenticated ? <CartPage /> : <Navigate to="/" />}
            />
          </Routes>
        </Router>
      </CartProvider>
    </div>
  );
}

export default App;
```

LOGINPAGE

```
import React, { useState } from "react";
import Logo from "../Logo/Logo";
import "./LoginForm.css";
import { useNavigate } from "react-router-dom";

function LoginForm() {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const navigate = useNavigate();

  function handleSubmit(e) {
    e.preventDefault();

    console.log("handlesubmit Login Called", username, password);
```

```
    try {
      fetch("https://dummyjson.com/auth/login", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({
          username: username,
          password: password,
        }),
      })
        .then((res) => {
          if (!res.ok) {
            throw new Error(`HTTP error! Status: ${res.status}`);
          }
          return res.json();
        })
        .then((data) => {
          if (data && data.token) {
            console.log(data);
            localStorage.setItem("auth", JSON.stringify(data));
            navigate("/home");
          } else {
            console.error("Login failed:", data);
          }
        })
        .catch((error) => {
          console.error("Error during login:", error);
        });
    } catch (error) {
      console.log(error);
    }
  }

  return (
    <div style={{ marginTop: "200px", marginBottom: "150px" }}>
      <div className="login-container">
        <form method="post">
          <center>
            <Logo />
          </center>
          <br />
          <div className="mui-input">
            <input
              type="text"
              value={username}
              placeholder="Username"
              onChange={(e) => {
                setUsername(e.target.value);
              }}
            />
          </div>
          <br />
          <div className="mui-input">
            <input
```

```
              type="password"
              value={password}
              placeholder="Password"
              onChange={(e) => {
                setPassword(e.target.value);
              }}
            />
          </div>
          <br />
          <button
            type="submit"
            className="btn btn-outline-dark"
            onClick={handleSubmit}
          >
            Submit
          </button>
          <div className="login-info mt-3">
            <span>
              <center>
                {" "}
                username:<strong>atuny0</strong>{" "}
              </center>

              <center>
                {" "}
                password: <strong>9uQFF1Lh</strong>
              </center>
            </span>
          </div>
        </form>
      </div>
    </div>
  );
}

export default LoginForm;
```

OTHER PROJECTS

- **Placement Cell Management System** *MERN STACK*

Developing an Online Placement Cell Management System,improving educational institution placements using React,Node.js, MongoDB, and innovative modules for students,alumni, teachers, and administrators

GithubLink:

https://github.com/DeependraMB/PlacementCellManagementSystem.git

- **OLX CLONE** *React JS, Firebase, JSX, CSS*

OLX Clone using React JS and Firebase web app for onlineclassified ads, including signup ,login, search, selling , and real-time updates.

GithubLink: https://github.com/DeependraMB/Olx_Clone_ReactJS.git

- **NETFLIX CLONE** *React JS, JSX, CSS, TMDB API*

Netflix clone using React.js, CSS, and JSX, showcasing prociency in front-end web development. Integrated with a moviedatabase API to display movie poster,category anddescriptions.Demonstrated skills in React.js, API integration,and UI development.

- **WEATHER APP** *React JS,JSX,CSS*

The React JS-based Weather App seamlessly integrates with APIs to deliver real-time weather data. Users can effortlessly retrieve accurate information on temperature for any location.

GithubLink: https://github.com/DeependraMB/weatherApp.git

In conclusion, if there are any queries or additional information needed, please feel free to reach out to me. I am more than happy to provide any further clarification or address any concerns you may have. Your consideration of my application is greatly appreciated, and I am eager to discuss the opportunity in more detail. Looking forward to the possibility of joining the [Company Name] team and contributing to the success of future projects. Thank you once again for considering my application.

DEEPENDRA M B

Email : deependramundalil2001@gmail.com

Ph No : 8606882759

Github : https://github.com/DeependraMB

LinkedIn : https://www.linkedin.com/in/deependramb/