A Major Project

on

E.Comm Store

Submitted in partial fulfilment of the requirementfor the award of the degree of

B. Tech in Computer Science & Engineering from Punjab

IKG Technical University, JALANDHAR

Submitted by:

Lovkush (2008085)

Guided by:

Ms. Poonam Mam

Session 2020-2023

Department of Computer Science & Engineering

SRI SUKHMANI INSTITUTE ENGINEERING&TECHNOLOGY

DERA BASSI, 140057

DECLARATION

Department of Computer Science & Engineering

I Lovkush, having Roll No. 2008085 declare that the project entitled "E.com Store" submitted for the B.Tech Degree in Computer Science & Engineering Department is our original work and the project has not formed the basis for the award of any degree, associate sip, fellowship, or any other similartitles. I further declare that in case of any violation of intellectual property rightsor copyright, I as the candidate will be fully responsible for the same. Our head, supervisor and institute should not be held for full or partial violation of copyright if found at any stage of our degree.

Signature of Students: Place: Chandigarh

Lovkush (2008085) Date:

The B.Tech. Major Project Viva-Voce examination of Lovkush,8th SEM,Computer Science & Engineering, has been held on

.

Sign. of Supervisor(s) Sign. of External Examiner

CERTIFICATE

Department of Computer Science & Engineering

This is to certify that the project entitled “E.com Store” is the bonafide work carried out Lovkush student of B. Tech, IKG Punjab Technical University, Jalandhar, during the year 2022-23, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering and that the project has not been formed thebasis for the award previously of any degree, diploma, associate ship, fellowship or any other similar title.

Signature of Students:
Lovkush (2008085)

Signature of the Guide:
Ms. Poonam

Signature of the H.O.D.: Prof. Ranju Marwaha

Date:

Place:

ACKNOWLDEGEMENT

I highly grateful to the **S.S.I.E.T** for providing this opportunity to carry out the 6 Months industrial training at Excellence Technology Mohali.

The constant guidance and encouragement received from Ms. Poonam, Project Incharge at Excellence Technology Mohali has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I would like to express a deep sense of gratitude and thanks to ID CARD MANUFACTURE project guide, without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

I express gratitude to other faculty members of Excellence Technology Mohali. for their intellectual support throughout the course of this work.

I would also like to thank my Internal Guide **Mrs, Ranju Marwaha** for his valuable time.I am also indebted to the entire **Department of Computer Science Eng..S.S.I.E.T** for their esteemed guidance, support and valuable suggestions.

Lovkush(2008085)

COMPANY PROFILE

Excellence technology provides best software industrial training in Chandigarh which helps you to reboot your professional career to excellent point to achieve your goal. Get an All types of software testing training which helps you to increase your testing software ability with knowledgeable and experienced faculty members, a transparent teaching methodology is applied. The advanced training is few of the things that make Excellence Technology a best Industrial institute in Chandigarh. We Are Providing Digital Marketing Services to Local & Global Clients. Our Main Services Are Website Designing & Development, Search Engine Optimization (SEO), Search Engine Marketing (SMM Or PPC), Lead Generation, Online Branding Etc. Our aim is to provide the best learning environment to our students. We help them to discover their potential to build their professional career. We believe that our students should receive best training. so that this led to effective careers growth and development.

ABSTRACT

The project "E.comStore" is a comprehensive web-based solution developed using MERN. The primary objective of this project is to create an efficient and user-friendly online platform for selling products. The website aims to bridge the gap between sellers and buyers by providing a convenient and secure platform for conducting transactions.

The project encompasses various features and functionalities that enhance the overall user experience. The website enables users to browse through a wide range of products, including Mobile, Earphones etc. Each product is accompanied by detailed descriptions, high-resolution images, and pricing information.

Registered users can create accounts and customize their profiles, enabling them to save their favorite products, track orders, and manage their purchase history. The website also incorporates a secure payment gateway, allowing users to make purchases using various payment methods, such as credit/debit cards or online wallets.

To ensure a seamless shopping experience, the website incorporates a search functionality, allowing users to quickly locate specific products based on categories, price ranges, or keywords. Additionally, a responsive design approach has been adopted, ensuring optimal viewing and usability across various devices, including desktops, laptops, tablets, and smartphones.

The development of this project involved the utilization of Node for server-side scripting, HTML and CSS for front-end development, MongoDB for database management, and jQuery, JavaScript, and the JS framework for creating dynamic and interactive elements. These technologies, when combined, provide a robust and scalable solution that meets the requirements of an ecommerce website.

In conclusion, the E.com Store project serves as a valuable platform for Electronics sellers and buyers to connect and transact online. It offers a user-friendly interface, secure payment options, and a wide range of Electronics products, catering to the diverse needs of customers. The successful implementation of this project demonstrates the effectiveness of the chosen technologies and showcases the potential of ecommerce in the electronics industry.

Table of Contents

Chapter 1:

Introduction

The E.com Store Website project aims to develop a comprehensive and user-friendly online platform for selling Electronics products. In today's digital age, ecommerce websites have gained immense popularity as they provide convenience and accessibility to customers who can browse and purchase products from the comfort of their homes. This project focuses specifically on the electronics industry, catering to the growing demand for online Electronics shopping.

The primary objective of the Electronics Ecommerce Website is to create a seamless and engaging shopping experience for customers looking to buy electronics items. The website will offer a wide range of Electronics products, including earrings, necklaces, bracelets, rings, and more. Customers will have the flexibility to explore various categories, view product details, make purchases, and track their orders.

Features such as product categorization, search functionality, user registration and authentication, shopping cart management, and secure payment options will be integrated into the website to enhance user experience and streamline the purchasing process. The use of NODE, HTML, MongoDB, CSS, jQuery, JavaScript, and the JS framework will ensure a responsive and visually appealing interface.

The E.com Store Website aims to bridge the gap between traditional brick-and- E.com Store and online shopping by providing a platform that offers convenience, a wide selection of products, and a secure and trustworthy shopping experience. By leveraging the power of technology and ecommerce, this project aims to cater to the evolving needs and preferences of customers in the electronics industry.

The successful implementation of the E.com Store Ecommerce Website will not only benefit customers by providing them with a convenient shopping platform but also offer business opportunities for Electronics vendors and manufacturers. The website will act as a digital

storefront, expanding their reach and potential customer base. Additionally, the project will contribute to the overall growth of the ecommerce industry and pave the way for future innovations and advancements in the field of online electronics retail.

Purpose:

The main purpose of developing the electronics Ecommerce Website is to address the limitations of the traditional brick-and-mortar electronics stores and provide customers with a convenient and accessible platform to browse, select, and purchase electronics items from the comfort of their homes. This project aims to bridge the gap between customers and electronics retailers, enhancing the overall shopping experience for both parties involved.

Problem with the Existing System:

The existing system of purchasing electronics primarily relies on physical stores, which poses several limitations and challenges. Some of the key problems with the existing system are:

a. Limited Accessibility: Physical electronics stores have limited operating hours and are often located in specific geographic areas, making it inconvenient for customers who are unable to visit the stores during those hours or reside far away.

b. Limited Variety: Physical stores have limited display space, restricting the range of electronics products available for customers to choose from. This limitation reduces the chances of finding unique or customized pieces.

c. Time-Consuming Process: Purchasing electronics from physical stores involves spending significant time in travelling, searching for the desired products, and waiting for assistance from sales personnel. This process can be tedious and time-consuming.

d. Lack of Information: Customers may not have access to detailed information about the electronics products, such as their specifications, materials used, or pricing. This lack of information can make it challenging for customers to make informed decisions.

e. Security Concerns: Customers may have concerns regarding the security of their valuable

purchases, such as the risk of theft or loss during transit.

Technologies used in the project.

- NODE
- HTML
- CSS
- Bootstrap
- MongoDB Server

Software Requirements

- DBMS Software
- Windows XP
- XamppServer
- Dreamweaver cc
- Visual code

Hardware Requirements

- Hard Disk – 2 GB.
- RAM – 1 GB.
- Processor – Dual Core or Above.
- Mouse.
- Keyboard.
- Monitor.

- Printer.

CHAPTER 2:

LITRATURE SURVEY

In conclusion, the electronics Ecommerce Website project aims to create a dynamic and efficient platform for online electronics shopping. By harnessing the power of various technologies and incorporating user-friendly features, the website will revolutionize the way customers explore, purchase, and experience the world of electronics.

In order to design and develop the electronics Ecommerce Website, a comprehensive literature survey was conducted to gain insights into existing ecommerce platforms and best practices in the field of web development. The following sources were referred to:

"Ecommerce Website Design: 10 Best Practices" by HubSpot: This article provided valuable insights into the key elements and design principles of successful ecommerce websites. It discussed the importance of clear navigation, appealing visuals, mobile responsiveness, and optimized checkout processes.

"The Impact of User Interface Design on Ecommerce Conversion Rates" by Smashing Magazine: This research paper examined the impact of user interface design on the conversion rates of ecommerce websites. It highlighted the significance of intuitive navigation, persuasive product descriptions, and effective call-to-action buttons.

"Database Design for Ecommerce Websites" by Shopify: This resource focused on the importance of database design for ecommerce websites. It discussed the various database models, such as relational and NoSQL, and their suitability for different types of ecommerce platforms.

"Secure Payment Systems for Ecommerce" by Worldpay: This document provided insights into secure payment systems and their implementation in ecommerce websites. It discussed the

importance of encryption, tokenization, and PCI compliance for protecting customer payment information.

"Responsive Web Design for Ecommerce" by Nielsen Norman Group: This research report explored the significance of responsive web design in ecommerce websites. It discussed the benefits of creating a consistent user experience across different devices and screen sizes.

CHAPTER 4:

SOFTWARE DETAILS

The IDE used in this project is Visual Studio Code. All the NODE files were created in VSC, and all thenecessary packages were easily installable in this IDE. For this project following technologies were used i.e.,NODE, MongoDB, JS, jQuery, HTML, CSS & Bootstrap. We have created a live GUI for interacting with the Academy as it gives a design and interesting look while having the conversation.

Visual Studio Code :

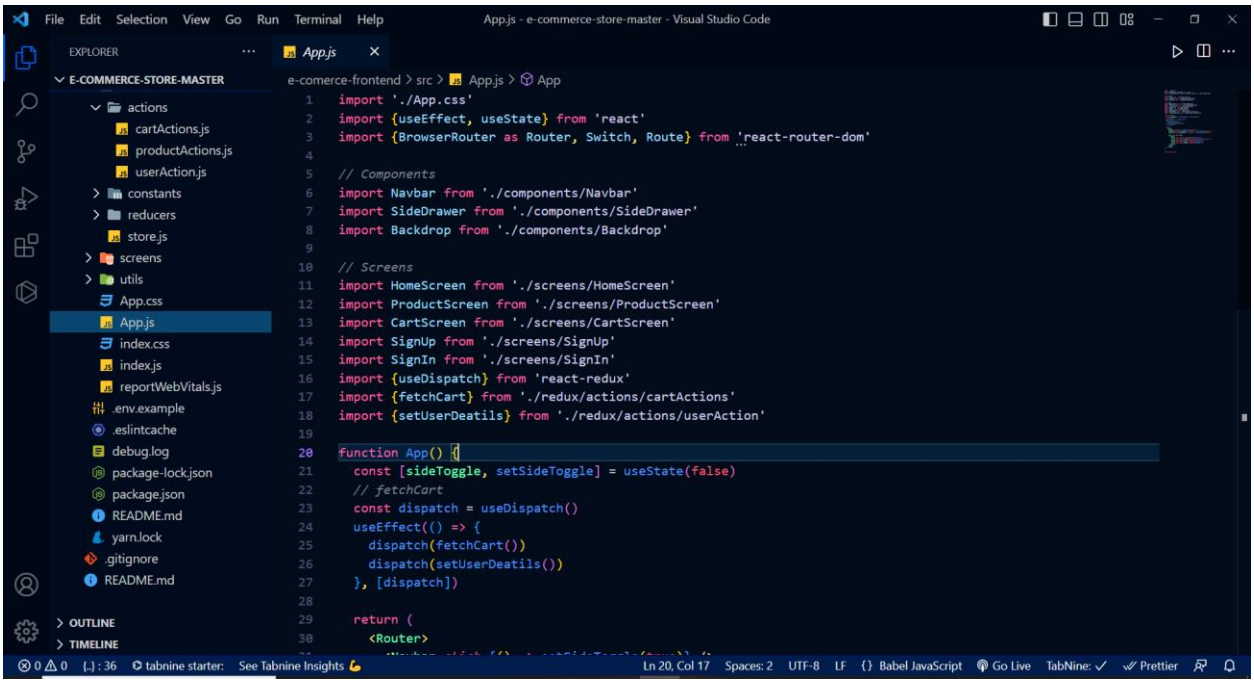


Figure 4.1 Visual Studio Code ID

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including PHP, C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component used in Azure

DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

ReactJS Keys

After answering what is ReactJs, let us know what are keys.

While dealing with components that are produced periodically in React, keys are essential. Your component will continue to be uniquely identifiable after the modification if the key value is set. They aid React in determining which elements have changed, been eliminated, or been added.

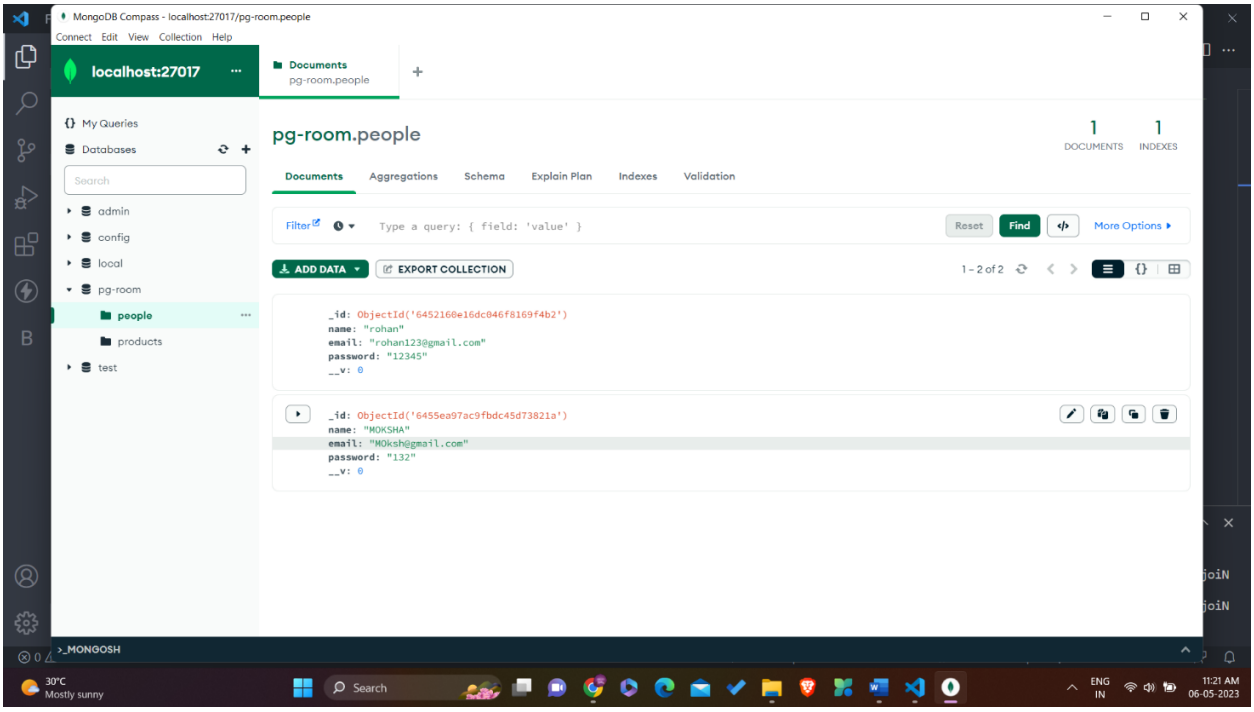
Figure 4.2 React-Js

MongoDB:

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as **Foreign Keys**.



```
import './App.css'
import {useEffect, useState} from 'react'
import {BrowserRouter as Router, Switch, Route} from 'react-router-dom'

// Components
import Navbar from './components/Navbar'
import SideDrawer from './components/SideDrawer'
import Backdrop from './components/Backdrop'

// Screens
import HomeScreen from './screens/HomeScreen'
import ProductScreen from './screens/ProductScreen'
import CartScreen from './screens/CartScreen'
import SignUp from './screens/SignUp'
import SignIn from './screens/SignIn'
import {useDispatch} from 'react-redux'
import {fetchCart} from './redux/actions/cartActions'
import {setUserDeatils} from './redux/actions/userAction'
```

Figure 4.3 Imported Modules

CHAPTER 5:

SOURCE CODE AND COMMAND

```
import './App.css'
import {useEffect, useState} from 'react'
import {BrowserRouter as Router, Switch, Route} from 'react-router-dom'

// Components
import Navbar from './components/Navbar'
import SideDrawer from './components/SideDrawer'
import Backdrop from './components/Backdrop'

// Screens
import HomeScreen from './screens/HomeScreen'
import ProductScreen from './screens/ProductScreen'
import CartScreen from './screens/CartScreen'
import SignUp from './screens/SignUp'
import SignIn from './screens/SignIn'
import {useDispatch} from 'react-redux'
import {fetchCart} from './redux/actions/cartActions'
import {setUserDeatils} from './redux/actions/userAction'

function App() {
  const [sideToggle, setSideToggle] = useState(false)
  // fetchCart
  const dispatch = useDispatch()
  useEffect(() => {
    dispatch(fetchCart())
    dispatch(setUserDeatils())
  }, [dispatch])

  return (
    <Router>
      <Navbar click={() => setSideToggle(true)} />
      <SideDrawer show={sideToggle} click={() => setSideToggle(false)} />
      <Backdrop show={sideToggle} click={() => setSideToggle(false)} />

      <main className="app">
        <Switch>
          <Route exact path="/" component={HomeScreen} />
          <Route exact path="/product/:id" component={ProductScreen} />
          <Route exact path="/cart" component={CartScreen} />
          <Route exact path="/signup" component={SignUp} />
          <Route exact path="/signin" component={SignIn} />
```

```
        </Switch>
      </main>
    </Router>
  )
}

export default App
```

```
import * as actionTypes from '../constants/cartConstants'
import axios from 'axios'
import {Api} from '../../utils/Api'
import {convertToCartData} from '../../utils/utils.function'

export const addToCart = (id, qty) => async dispatch => {
  const {data} = await Api.getRequest(`/api/products/${id}`)
  const product = JSON.parse(data)
  // console.log(product)
  dispatch({
    type: actionTypes.ADD_TO_CART,
    payload: {
      product: product._id,
      name: product.name,
      imageUrl: product.imageUrl,
      price: product.price,
      countInStock: product.countInStock,
      qty,
    },
  })

  Api.postRequest('/api/cart', {productId: id, count: qty})
}

export const removeFromCart =
  ({pId, _id}) =>
  dispatch => {
    dispatch({
      type: actionTypes.REMOVE_FROM_CART,
      payload: pId,
    })
    Api.DeleteRequest('/api/cart/' + _id)
```

```
  }

export const fetchCart = () => async dispatch => {
  try {
    const {data: strigifyData} = await Api.getRequest(`/api/cart/`)
    // console.log({strigifyData})
    const {carts} = JSON.parse(strigifyData)
    // console.log(carts)

    dispatch({
      type: actionTypes.FETCH_MY_CART,
      payload: {
        carts: convertToCartData(carts),
      },
    })
  } catch (e) {
    console.log('ERORROR :  ', e)
  }
}

// x = [
//   {
//     product: '615ac2b036ecb5ed71497630',
//     name: 'Cannon EOS-1D',
//     imageUrl:
//       'https://images.unsplash.com/photo-1519183071298-
a2962feb14f4?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-
1.2.1&auto=format&fit=crop&w=1350&q=80',
//     price: 1300,
//     countInStock: 5,
//     qty: '3',
//   },
//   {
//     product: '615ac2b036ecb5ed71497631',
//     name: 'Amazon Alexa',
//     imageUrl:
//       'https://images.unsplash.com/photo-1518444065439-
e933c06ce9cd?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-
1.2.1&auto=format&fit=crop&w=1267&q=80',
//     price: 50,
//     countInStock: 25,
//     qty: '3',
//   },
// ]
```

```
import * as actionTypes from '../constants/productConstants'
import axios from 'axios'
import {Api} from '../../utils/Api'

export const getProducts = () => async dispatch => {
  try {
    dispatch({type: actionTypes.GET_PRODUCTS_REQUEST})

    const {data} = await Api.getRequest('/api/products')

    dispatch({
      type: actionTypes.GET_PRODUCTS_SUCCESS,
      payload: JSON.parse(data),
    })
  } catch (error) {
    dispatch({
      type: actionTypes.GET_PRODUCTS_FAIL,
      payload:
        error.response && error.response.data.message
          ? error.response.data.message
          : error.message,
    })
  }
}

export const getProductDetails = id => async dispatch => {
  try {
    dispatch({type: actionTypes.GET_PRODUCT_DETAILS_REQUEST})

    const {data} = await Api.getRequest(`/api/products/${id}`)
    const p = JSON.parse(data)
    dispatch({
      type: actionTypes.GET_PRODUCT_DETAILS_SUCCESS,
      payload: {
        ...p,
      },
    })
  } catch (error) {
    dispatch({
      type: actionTypes.GET_PRODUCT_DETAILS_FAIL,
```

```
      payload:
        error.response && error.response.data.message
          ? error.response.data.message
          : error.message,
    })
  }
}

export const removeProductDetails = () => dispatch => {
  dispatch({type: actionTypes.GET_PRODUCT_DETAILS_RESET})
}

import {Api} from '../../utils/Api'
import * as actionTypes from '../constants/userContants'

export const setUserDeatils = () => async dispatch => {
  const {statusCode, data} = await Api.getRequest(`/api/user/me`)
  // console.log({statusCode, data})
  if (statusCode === 400 || statusCode === 500) {
    dispatch({
      type: actionTypes.SET_INITIAL_STATE,
    })
    return
  }
  const {user} = JSON.parse(data)
  dispatch({
    type: actionTypes.SET_USER,
    payload: {
      isLogin: true,
      details: {...user},
    },
  })
}

export const setInitialState = () => async dispatch => {
  dispatch({
    type: actionTypes.SET_INITIAL_STATE,
  })
}

.cartitem {
  width: 100%;
  padding: 1rem;
  display: grid;
  grid-template-columns: 1fr 4fr 1fr 1fr 1fr;
```

```
  gap: 8px;
  background: #fff;
  border-radius: 2px;
  place-items: center;
  margin-bottom: 8px;
}

.cartItem__name {
  text-decoration: none;
  color: #171717;
}

.cartItem__name:hover {
  color: #dd219e;
}

.cartItem__select {
  padding: 10px 17px;
}

.cartItem__deleteBtn {
  padding: 10px 17px;
  color: red;
  background: #f4f4f4;
  border: 1px solid #171717;
  cursor: pointer;
  transition: all 0.3s ease-out;
}

.cartItem__deleteBtn:hover,
.carItem__deleteBtn:active,
.carItem__deleteBtn:focus {
  background: #171717;
  transform: scale(1.2);
}

@media (max-width: 700px) {
  .cartItem__name {
    font-size: 0.8rem;
  }

  .cartItem__select,
  .cartItem__deleteBtn {
    padding: 8px 13px;
  }
```

```
  .cartitem__price {
    font-size: 0.8rem;
  }
}

@media (max-width: 700px) {
  .cartItem__name {
    font-size: 0.6rem;
  }

  .cartItem__select,
  .cartItem__deleteBtn {
    padding: 5px 8px;
  }

  .cartitem__price {
    font-size: 0.6rem;
  }
}

import "./CartItem.css";
import { Link } from "react-router-dom";

const CartItem = ({ item, qtyChangeHandler, removeHandler }) => {
  return (
    <div className="cartitem">
      <div className="cartitem__image">
        <img src={item.imageUrl} alt={item.name} />
      </div>
      <Link to={`/product/${item.product}`} className="cartItem__name">
        <p>{item.name}</p>
      </Link>
      <p className="cartitem__price">${item.price}</p>
      <select
        value={item.qty}
        onChange={(e) => qtyChangeHandler(item.product, e.target.value)}
        className="cartItem__select"
      >
        {[...Array(item.countInStock).keys()].map((x) => (
          <option key={x + 1} value={x + 1}>
            {x + 1}
          </option>
        ))}
      </select>
```

```
      <button
        className="cartItem__deleteBtn"
        onClick={() => removeHandler(item.product)}
      >
        <i className="fas fa-trash"></i>
      </button>
    </div>
  );
};

export default CartItem;

import './Navbar.css'
import {Link, useHistory} from 'react-router-dom'
import {useDispatch, useSelector} from 'react-redux'
import {useMemo} from 'react'
import {logout} from '../utils/localstorage'
import {setInitialState} from '../redux/actions/userAction'

const Navbar = ({click}) => {
  const cart = useSelector(state => state.cart)
  const history = useHistory()
  const user = useSelector(state => state.user)
  const dispatch = useDispatch()
  // console.log({user})

  const {cartItems} = cart

  const getCartCount = () => {
    return cartItems.reduce((qty, item) => Number(item.qty) + qty, 0)
  }

  const _handleLogout = () => {
    // console.log('click')
    dispatch(setInitialState())
    logout()
    history.push('/')
  }

  return (
    <nav className="navbar">
      <div className="navbar__logo">
        <h2>JSOM-E-COMERCE</h2>
      </div>
```

```
      <ul className="navbar__links">
        <li>
          <Link to="/cart" className="cart__link">
            <i className="fas fa-shopping-cart"></i>
            <span>
              Cart <span className="cartlogo__badge">{getCartCount()}</span>
            </span>
          </Link>
        </li>

        <li>
          <Link to="/">Shop</Link>
        </li>

        {!user.userInfo.isLogin ? (
          <li>
            <Link to="/signin">Login</Link>
          </li>
        ) : (
          <li>
            <p onClick={_handleLogout}>Logout</p>
          </li>
        )}
      </ul>

      <div className="hamburger__menu" onClick={click}>
        <div></div>
        <div></div>
        <div></div>
      </div>
    </nav>
  )
}

export default Navbar
```

CHAPTER 6:

INPUT/OUTPUT SCREENSHOT

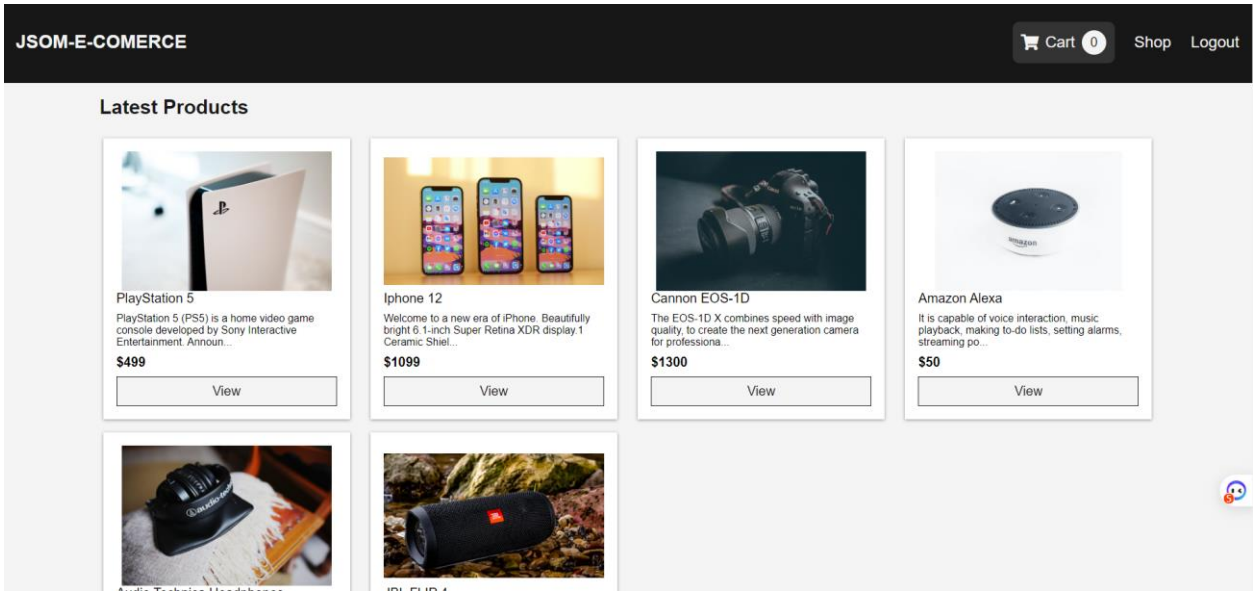


Figure 6.1 E.com Store Website

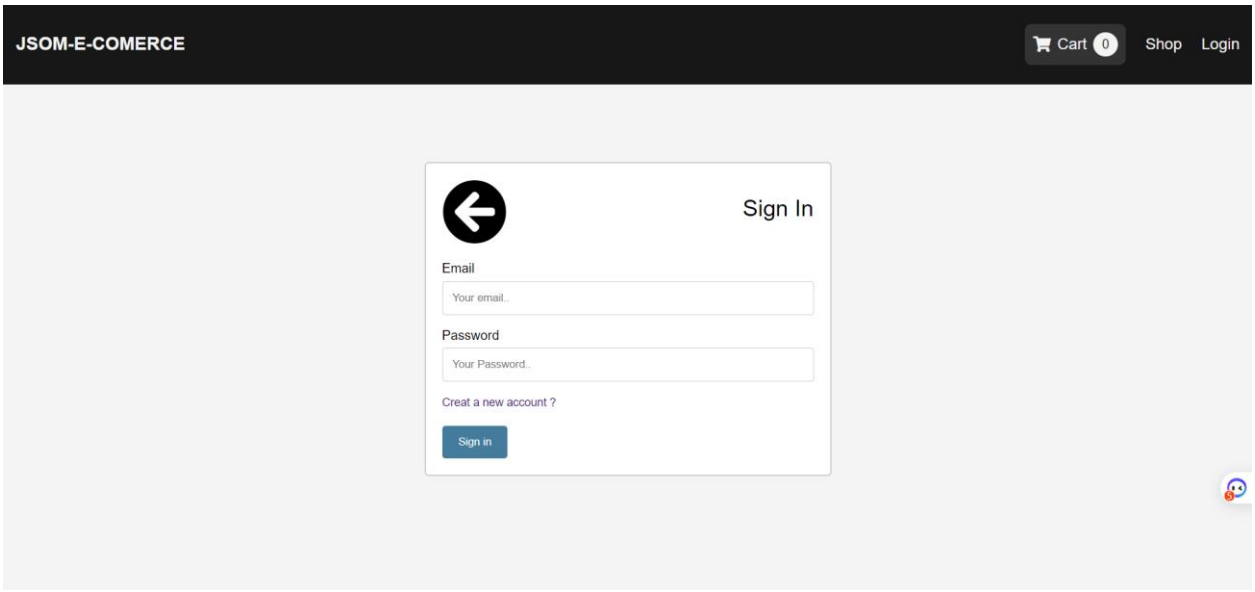


Figure 6.2 login Admin: The authority who operates the application, manages theuser's panel, etc.

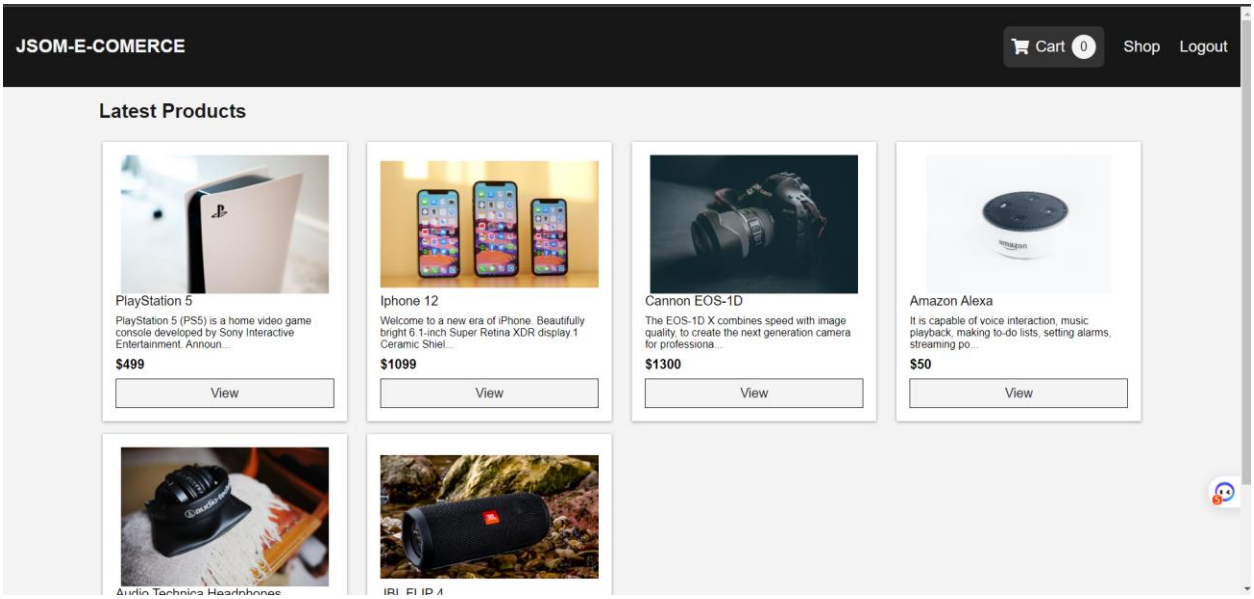
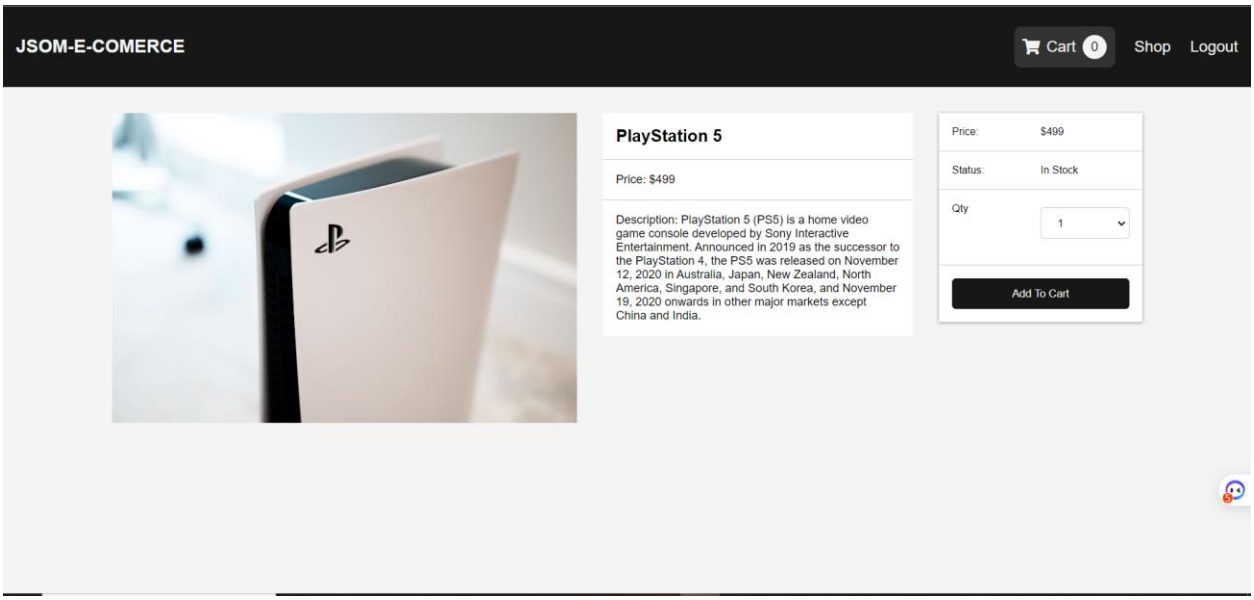


Figure 6.3: Dashboard



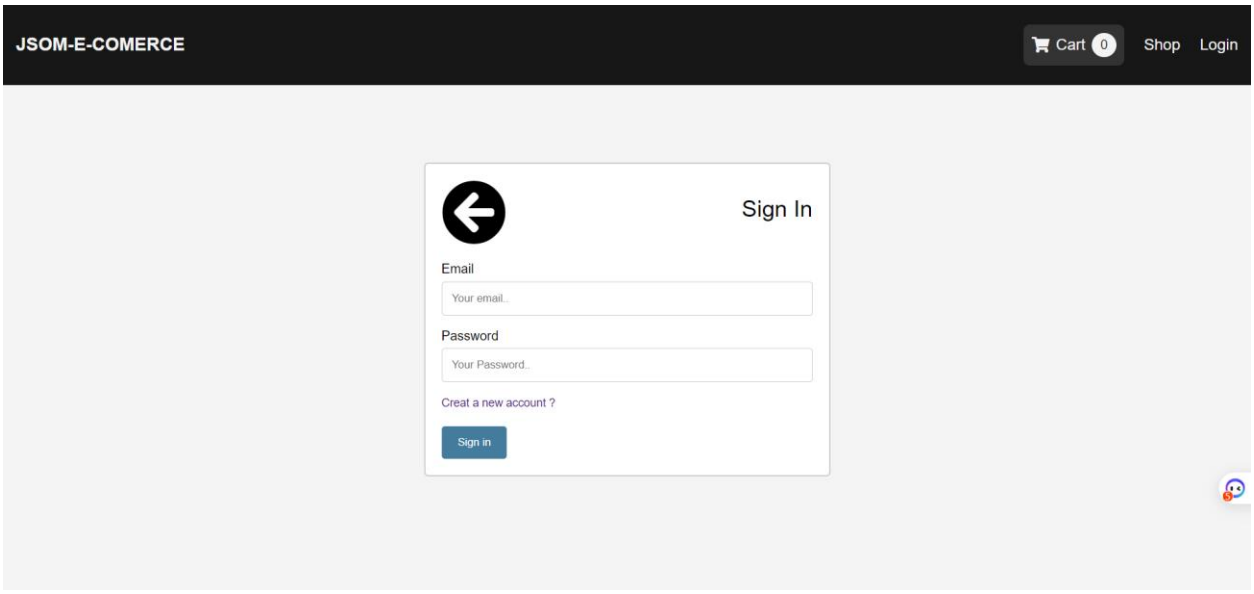


Figure 6.5: User Account

CHAPTER 7

CONCLUSION

In conclusion, the development of the Ecom Store Website has been a significant achievement. The project successfully implemented a user-friendly and secure online platform for customers to browse and purchase E.com products. The website incorporates various technologies such as NODE, HTML, MongDB, CSS, jQuery, JavaScript, and the Bootstrap framework to provide an efficient and visually appealing user experience.

Throughout the development process, several key objectives were achieved. The website allows customers to easily navigate through different categories of E.com Store items, search for specific products, add items to their shopping cart, and proceed with secure payment options. The user registration and authentication system ensure the privacy and security of customer information.

However, like any project, there were certain limitations encountered during the development process. These limitations include:

Scalability: The current implementation of the website may face challenges in scaling up to accommodate a large number of concurrent users. Additional measures would be required to ensure optimal performance under high traffic conditions.

Mobile Responsiveness: While the website is designed to be responsive, further improvements could be made to enhance the mobile browsing experience and ensure compatibility with a wide range of devices.

User Feedback and Reviews: Incorporating a feedback and review system would provide valuable insights for both customers and the business, enabling continuous improvement and building trust among users.

CHAPTER 8:

LIMITATIONS & FUTURE WORK

LIMITATIONS:

Scalability: The scalability of the website may be limited as the project does not incorporate advanced techniques like load balancing or server clustering. This may result in performance issues and difficulties in handling a large number of concurrent users.

Mobile Responsiveness: The website may not be fully optimized for mobile devices, which could lead to a suboptimal user experience for customers accessing the site from smartphones or tablets.

Security: While the project addresses basic security measures, such as user authentication and secure payment options, it may still have potential vulnerabilities that could be exploited by malicious users. Advanced security measures like encryption, secure coding practices, and regular vulnerability assessments were not extensively implemented.

Limited Payment Options: The website currently supports only a limited number of payment options. Future enhancements could include integrating popular payment gateways to provide more flexibility for customers.

FUTURE WORK:

Enhancing User Experience: The website can be improved by implementing a more intuitive and user-friendly interface. This could involve enhancing navigation, optimizing page loading speed, and incorporating features such as product recommendations based on user preferences.

Mobile Responsiveness: The website should be made fully responsive to ensure a seamless user experience across all devices, including smartphones and tablets. This could involve adopting a responsive web design approach or developing a dedicated mobile application.

Advanced Search Functionality: Future development could focus on enhancing the search functionality to allow customers to filter and sort products based on specific criteria, such as price range, material, or design.

Social Media Integration: Integrating social media platforms can help increase brand visibility and customer engagement. Adding features like social sharing, customer reviews, and social login options can enhance the overall user experience and facilitate organic marketing.

Advanced Security Measures: To ensure the security of user data and transactions, future work could include implementing advanced security measures such as SSL/TLS encryption, two-factor authentication, and regular security audits to identify and address any vulnerabilities.

Performance Optimization: The website's performance can be improved by implementing techniques like caching, optimizing database queries, and leveraging content delivery networks (CDNs) to reduce server load and enhance page loading speed.

Multilingual Support: Adding multilingual support to the website can help expand the customer base by catering to users from different regions and languages.

Inventory Management: Integrating an inventory management system can help streamline the process of tracking and managing product stock levels, ensuring accurate product availability information for customers.

By addressing these limitations and incorporating the suggested future work, the Electronics Ecommerce Website can become a more robust and feature-rich platform, providing an enhanced shopping experience for customers.

CHAPTER 9

REFERENCES

W3Schools: https://www.w3schools.com

This website provided valuable resources and tutorials for learning HTML, CSS, JavaScript, and NODE.
React Documentation: https://react.dev/doc.

Stack Overflow: https://stackoverflow.com/

This platform was used to find solutions to specific coding problems and errors encountered during the development process.
GitHub: https://github.com/

GitHub was utilized for version control and collaboration during the development of the project.
Adobe Photoshop: https://www.adobe.com/products/photoshop.html

Adobe Photoshop was used for creating and editing images and graphics used on the website.
Sublime Text: https://www.sublimetext.com/

Sublime Text was the text editor used for writing and editing code.
XAMPP: https://www.apachefriends.org/index.html

XAMPP was used as the local development environment, providing Apache web server, MySQL database.
Google Fonts: https://fonts.google.com/

Google Fonts provided a wide range of fonts that were used for typography on the website.
Unsplash: https://unsplash.com/