E-COMMERCE WEBSITE REPORT

Thesis submitted by

SACHIN
2018CS50418

under the guidance of

Prof.Vireshwar Kumar ,

Indian Institute of Technology Delhi

Master of Technology

Department Of COMPUTER SCIENCE ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI

APRIL 2024

0.1 React Components

0.1.1 AddToCart.js

This file manages the functionality for adding items to the shopping cart. It likely includes event handlers to capture user interactions such as clicking an "Add to Cart" button and updating the cart state accordingly.

0.1.2 CartAmountToggle.js

Responsible for handling the functionality related to adjusting the quantity of items in the shopping cart. It may include buttons or input fields for increasing or decreasing the quantity of items and updating the cart state accordingly.

0.1.3 CartItem.js

Represents a component that displays individual items in the shopping cart. It likely includes information such as the item name, price, quantity, and options for removing the item from the cart.

0.1.4 FeatureProduct.js

Contains functionality for showcasing featured products on the website. This component may display a curated selection of products that are highlighted for promotional purposes or based on user preferences.

0.1.5 FilterSection.js

This file likely contains components or functionality for filtering products based on certain criteria such as category, price range, or product features. It may include filter options and event handlers to update the displayed products based on user selections.

0.1.6 Footer.js

Represents the footer component for the website. It typically contains links to important pages such as the About Us, Contact, and Terms of Service pages, as well as social media icons, contact details, and copyright information.

0.1.7 GridView.js

This component displays products in a grid view layout. It likely renders individual product items in a grid format, allowing users to view multiple products at once with thumbnail images, product names, and prices.

0.1.8 Header.js

Contains the header component for the website, which typically includes the site logo, navigation menu, search bar, and possibly additional elements such as a shopping cart icon or user account options.

0.1.9 HeroSection.js

This component represents the hero section of the website, which is often located at the top of the page and showcases important content or images. It may include a large banner image, headline text, and call-to-action buttons.

0.1.10 ListView.js

Displays products in a list view layout, allowing users to view product details in a linear format. It may include product names, prices, descriptions, and options for viewing additional product details or adding products to the cart.

0.1.11 MyImage.js

This file likely contains functionality related to handling images, such as displaying images, resizing images, or implementing image carousels or sliders.

0.1.12 Nav.js

Contains the navigation component for the website, providing links to different sections or pages. It may include the main navigation menu, dropdown menus for navigating subcategories, and possibly a mobile-friendly navigation menu for smaller screens.

0.1.13 PageNavigation.js

Includes a component for displaying page navigation elements such as breadcrumbs or links to different pages within the website. It helps users navigate between different sections or pages of the website easily.

0.1.14 Product.js

Represents a component for displaying individual product items. It may include product images, names, prices, descriptions, and options for adding products to the cart or viewing additional product details.

0.1.15 ProductList.js

Contains the main component for displaying a list of products. It likely integrates features such as filtering, sorting, and pagination to help users navigate through a large catalog of products efficiently.

0.1.16 Services.js

This file contains components or functionality related to displaying services offered by the website or company. It may include descriptions of different services, icons or images representing each service, and possibly links to learn more about each service.

0.1.17 Sort.js

This file includes functionality for sorting products based on certain criteria such as price, popularity, or alphabetical order. It may provide dropdown menus or other UI elements for users to select their preferred sorting option.

0.1.18 Star.js

Contains components or functionality related to displaying star ratings for products or services. It may render star icons based on a numerical rating and display the average rating along with the number of reviews.

0.1.19 Trusted.js

This file likely contains components or functionality for displaying logos of trusted companies or partners. It may include a carousel or grid layout to showcase logos of various partners or clients that the website has worked with or is affiliated with.

0.2 React Context Files

0.2.1 cart_context.js

Manages the state and functionality related to the shopping cart. Uses createContext, useReducer, and useContext hooks from React. Defines initial state for the cart, including items, total item count, total amount, and shipping fee. Provides a CartProvider component to wrap the application with the cart context. Exports a custom hook useCartContext to access the cart context and its state values within other components.

0.2.2 filter_context.js

Manages the state and functionality related to filtering products. Uses createContext, useReducer, and useContext hooks from React. Defines initial state for filtering, including filter products, all products, grid/list view, sorting value, and filter parameters. Provides a FilterContextProvider component to wrap the application with the filter context. Exports a custom hook useFilterContext to access the filter context and its state values within other components.

0.2.3 productcontex.js

Manages the state and functionality related to product data. Uses createContext, useReducer, useEffect, and useContext hooks from React. Utilizes Axios for making HTTP requests to fetch product data. Defines initial state for product-related data, including loading states, products, featured products, and single product details. Provides an AppProvider component to wrap the application with the product context. Exports a custom hook useProductContext to access the product context and its state values within other components.

0.3 FormatPrice.js

The FormatPrice component takes a price value in cents and returns a formatted currency string in Indian Rupees using the Intl.NumberFormat API.

0.4 Reducers

0.4.1 cartReducer.js

Manages the state related to the shopping cart. Handles actions such as adding items to the cart and removing items from the cart. When adding items to the cart, it creates a new cart item object and adds it to the existing cart array. When removing items from the cart, it filters out the item with the specified ID from the cart array. If the action type does not match any of the defined cases, it returns the current state unchanged.

0.4.2 filterReducer.js

Manages the state related to product filtering and sorting. Handles actions such as loading filtered products, setting the view (grid or list), sorting products, updating filter values, and clearing filters. When loading filtered products, it sets the filter_products array with the payload data and updates filter-related properties such as maxPrice based on the loaded products. When sorting products, it sorts the filter_products array based on the selected sorting value. When updating filter values, it updates the corresponding property in the filters object with the provided value. If the action type does not match any of the defined cases, it returns the current state unchanged.

0.4.3 productReducer.js

Manages the state related to product data and loading status. Handles actions such as setting loading status, setting API data (products and featured products), setting loading status for a single product, setting single product data, and handling errors. When setting API data, it extracts featured products from the payload and updates the products and featureProducts arrays in the state. When setting single product data, it updates the singleProduct property with the payload data. If the action type does not match any of the defined cases, it returns the current state unchanged.

0.5 About.js

0.5.1 Import Statements:

Imports the HeroSection component from the ./components/HeroSection file. Imports the useProductContext hook from the ./context/productcontex file.
0.6 App.js

0.5.2 About Component:

Defines a functional component named About. Inside the component, it utilizes the useProductContext hook to access the myName value from the product context. Creates a data object with a name property set to "E–commerce".

0.5.3 Return Statement:

Returns a fragment <> containing:

- The myName value obtained from the product context.
- The HeroSection component, passing the myData prop with the data object.

0.5.4 Export Statement:

Exports the About component as the default export of the file.

0.6 App.js

0.6.1 Import Statements:

Imports necessary components and dependencies from various files: React components like About, Home, Products, Contact, Cart, SingleProduct, ErrorPage, Header, and Footer. Required utilities like ThemeProvider from styled-components for theme management.
Routing components like BrowserRouter, Routes, and Route from react-router-dom. Global styles defined in GlobalStyle.

0.6.2 Theme Definition:

Defines a theme object containing various color and media properties used for styling throughout the application.

0.6.3 App Component:

Defines a functional component named App. Inside the component, it wraps the entire application with the ThemeProvider, providing the theme object defined earlier.

0.6.4 Router Setup:

Uses BrowserRouter to set up routing for the application. Includes Routes component to define the route configuration. Sets up individual routes using the Route component, each with a specific path and corresponding component to render:

- "/" for the Home page (<Home />).
- "/about" for the About page (<About />).
- "/products" for the Products page (<Products />).
- "/contact" for the Contact page (<Contact />).
- "/singleproduct/:id" for displaying a single product (<SingleProduct />).
- "/cart" for the Cart page (<Cart />).
- "/*" for all other paths, which renders the ErrorPage component (<ErrorPage />).

0.6.5 Global Styles:

Applies global styles using the GlobalStyle component.

0.6.6 Header and Footer:

Includes the Header and Footer components, ensuring they are displayed on every page.

0.6.7 Export Statement:

Exports the App component as the default export of the file.

0.7 File Descriptions

0.7.1 Cart.js

This file contains a React component responsible for rendering the shopping cart page. It utilizes styled-components for styling and the useCartContext hook to access cart-related data from the context. It maps over the items in the cart and renders CartItem components for each item.

0.7.2 Contact.js

This file defines a React component for the contact page. It includes a contact form and a Google Maps iframe for location visualization. The form uses Formspree for form submission and styled-components for styling.

0.7.3 ErrorPage.js

This file contains a React component for displaying a 404 error page. It provides a message indicating that the requested page does not exist and includes a button to navigate back to the homepage. The page utilizes styled-components for styling.

0.7.4 GlobalStyle.js

This file defines global styles using styled-components. It sets up default styling for various HTML elements, including font sizes, colors, and scrollbar styles. It also includes media queries for responsive design.

0.7.5 Home.js

This file contains a React component for the home page. It includes sections such as HeroSection, FeatureProduct, Services, and Trusted. Each section may contain different content, such as images, text, or icons.

0.7.6 index.js

This file is the entry point of the application. It renders the root component (App) inside the ReactDOM.createRoot function and wraps it with context providers (AppProvider, FilterContextProvider, CartProvider). It also imports and renders the GlobalStyle component and calls reportWebVitals for performance measurement.

0.7.7 Products.js

This file defines a React component for the products page. It includes sections for filtering products, sorting products, and displaying product lists. It uses styled-components for styling and the useFilterContext hook to access filter-related data from the context.

0.7.8 reportWebVitals.js

This file defines a function to report web vital metrics such as CLS, FID, FCP, LCP, and TTFB. It imports web-vitals and uses its functions to measure and report these metrics.

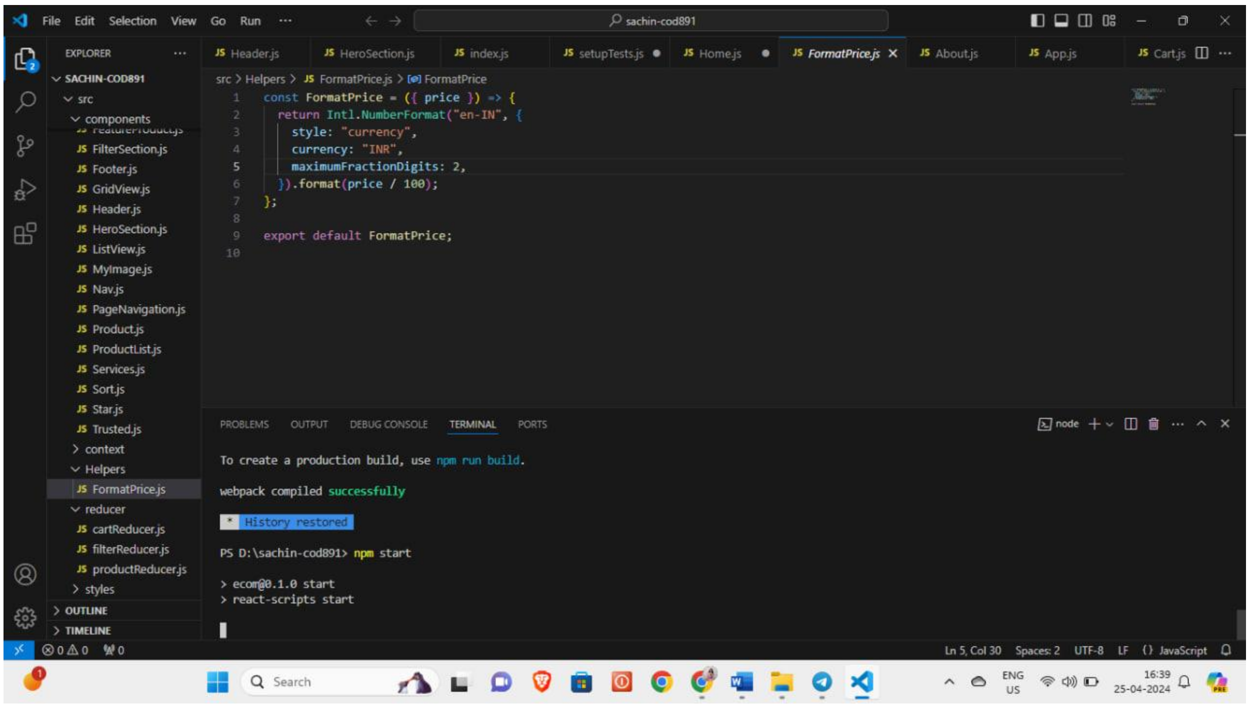
0.7.9 setupTests.js

This file configures the testing environment for Jest. It imports ’@testing-library/jest-dom’ to extend Jest’s expect function with DOM testing utilities.

0.7.10 SingleProduct.js

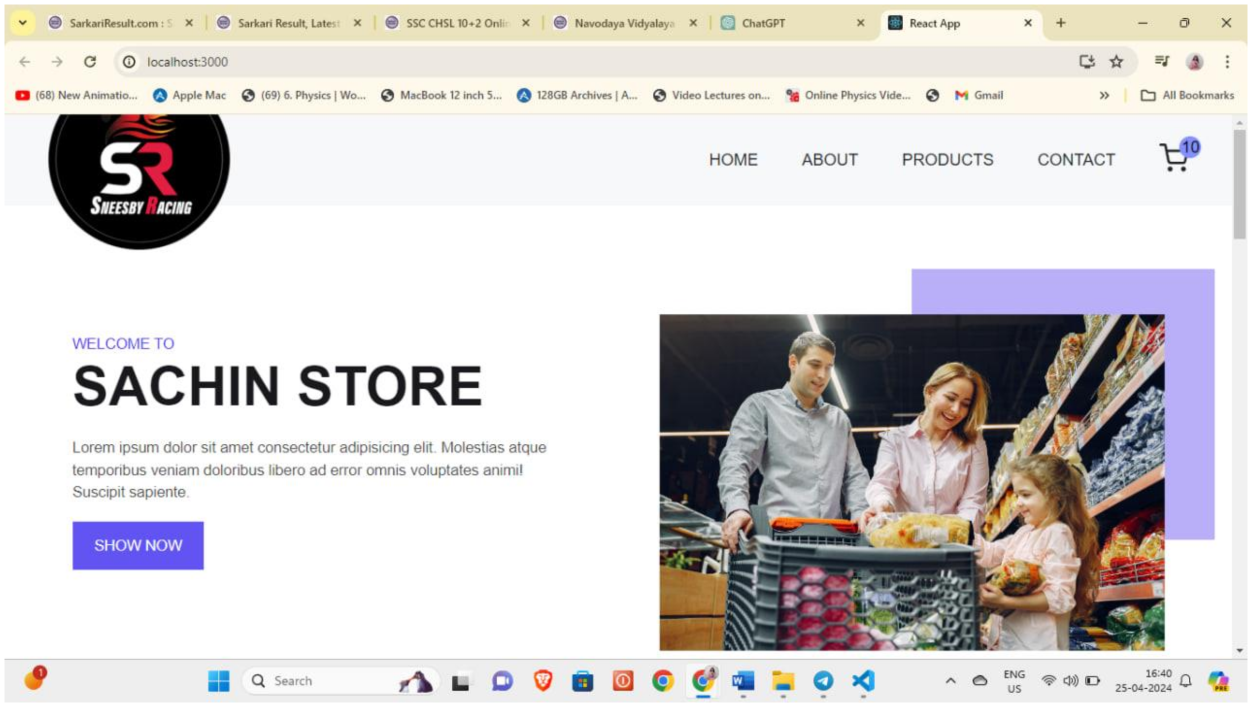
This file contains a React component for displaying detailed information about a single product. It fetches product data based on the product ID from the API and renders the product’s name, image, description, price, availability, and additional details. It also includes an "Add to Cart" button if the product is in stock.

GITHUB LINK: https://github.com/sachinsinghRathoreji/cod891project



```
const FormatPrice = ({ price }) => {
  return Intl.NumberFormat("en-IN", {
    style: "currency",
    currency: "INR",
    maximumFractionDigits: 2,
  }).format(price / 100);
};

export default FormatPrice;
```

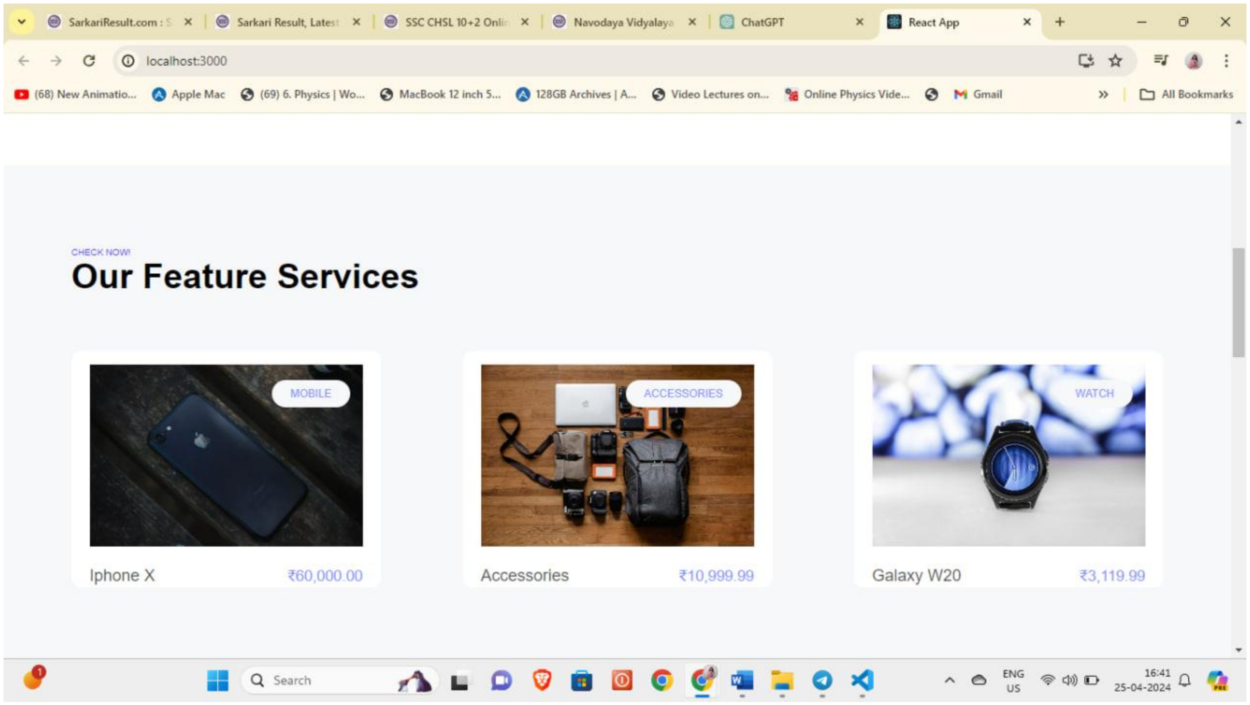
```
To create a production build, use npm run build.

webpack compiled successfully

 * History restored

PS D:\sachin-cod891> npm start

> ecom@0.1.0 start
> react-scripts start
```



WELCOME TO

SACHIN STORE

Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias atque temporibus veniam doloribus libero ad error omnis voluptates animi! Suscipit sapiente.



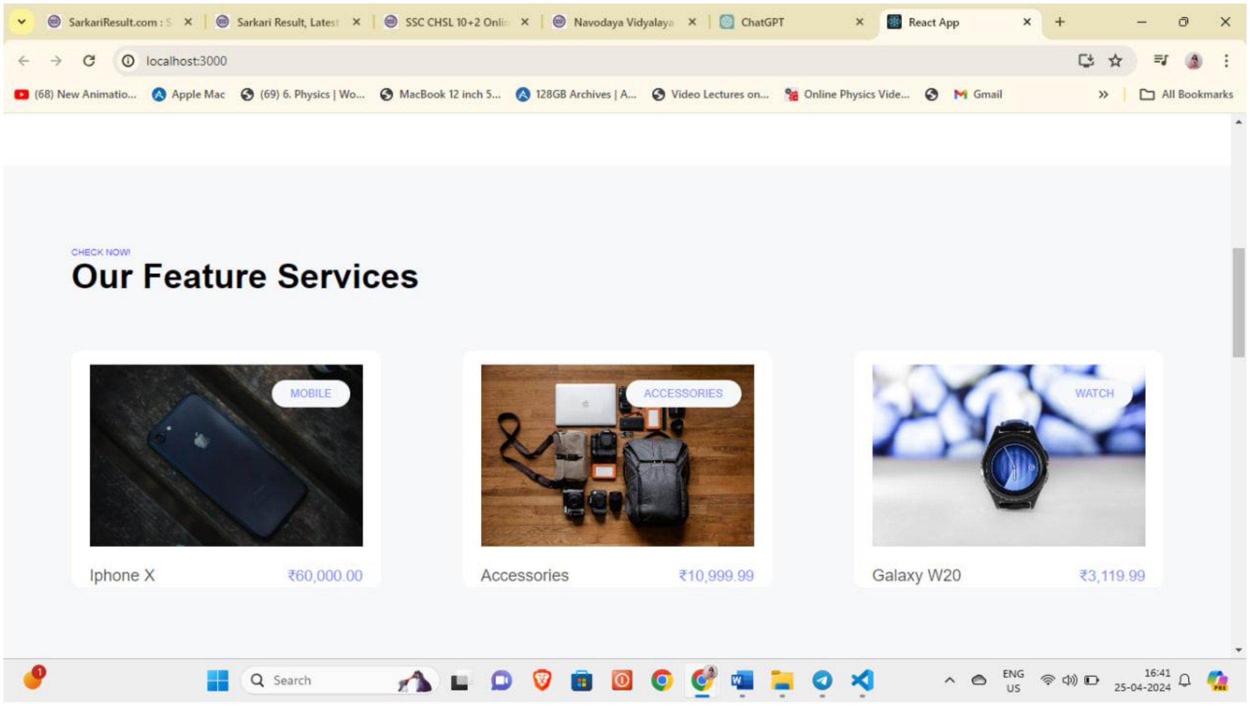
CHECK NOW!

Our Feature Services

Iphone X ₹60,000.00

Accessories ₹10,999.99

Galaxy W20 ₹3,119.99



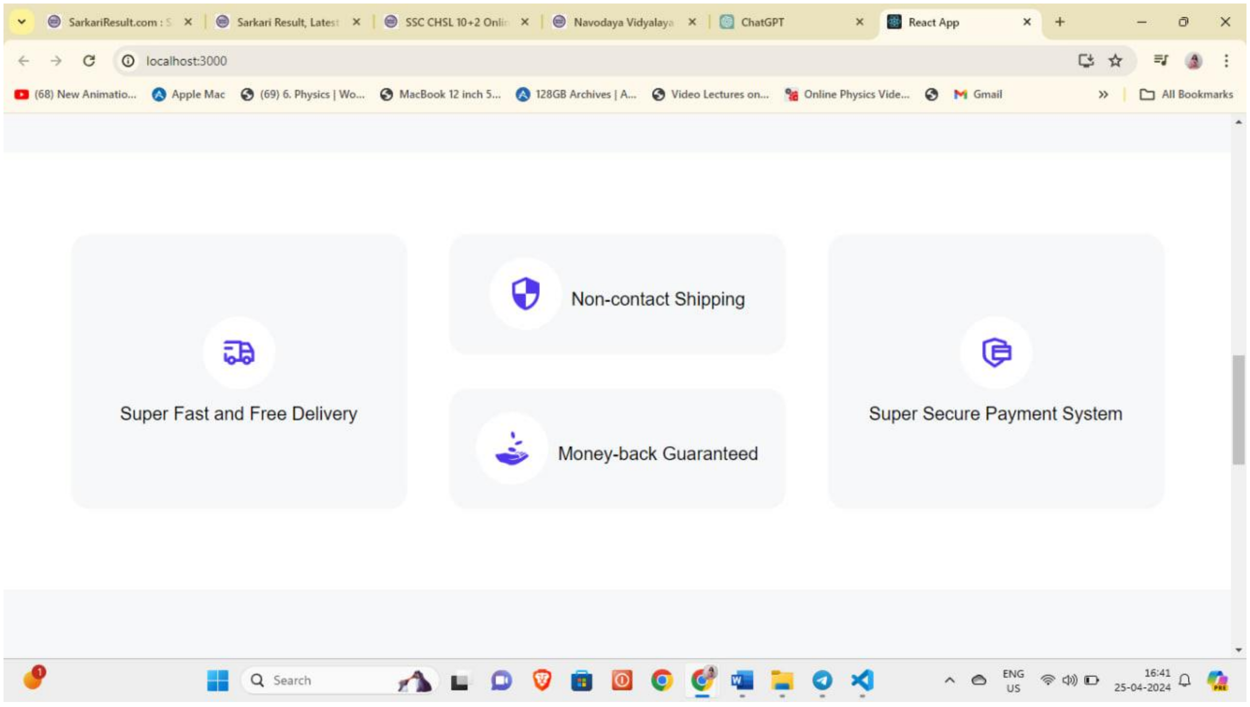
CHECK NOW!

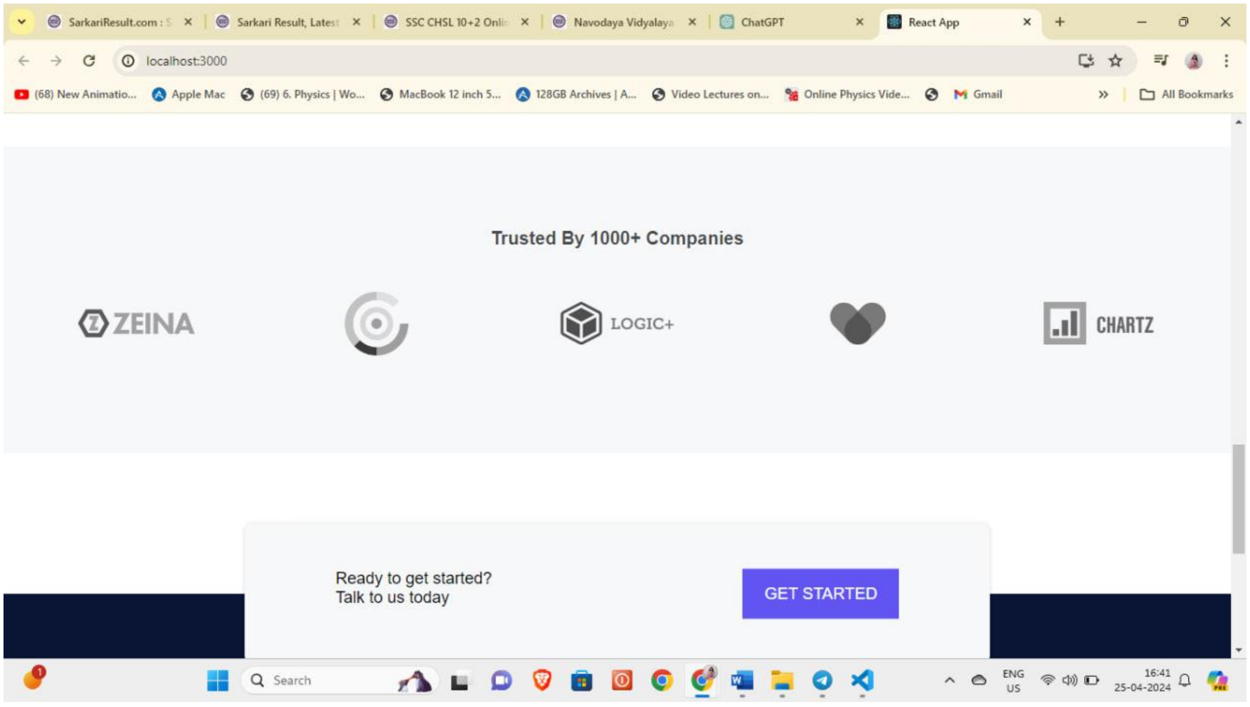
Our Feature Services

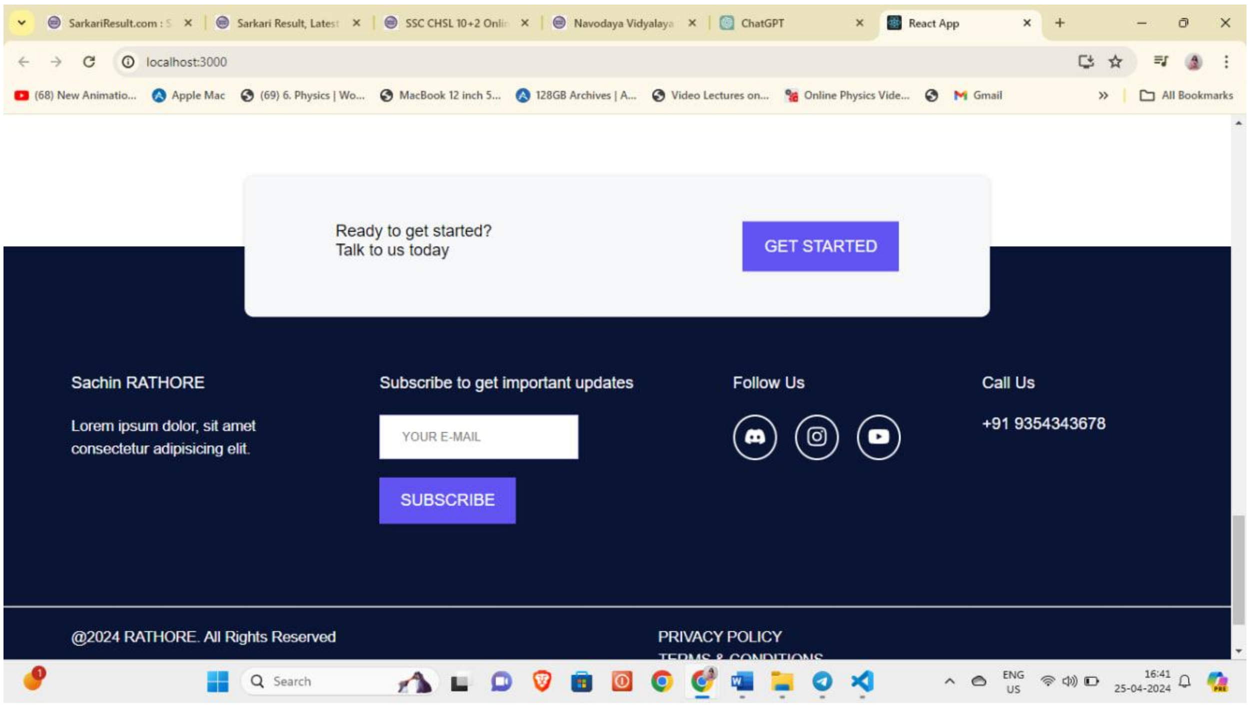
Iphone X ₹60,000.00

Accessories ₹10,999.99

Galaxy W20 ₹3,119.99


localhost:3000
Super Fast and Free Delivery
Non-contact Shipping
Money-back Guaranteed
Super Secure Payment System


localhost:3000
Trusted By 1000+ Companies
ZEINA
LOGIC+
CHARTZ
Ready to get started?
Talk to us today
GET STARTED



Ready to get started?
Talk to us today

GET STARTED

Sachin RATHORE

Lorem ipsum dolor, sit amet consectetur adipisicing elit.

Subscribe to get important updates

YOUR E-MAIL

SUBSCRIBE

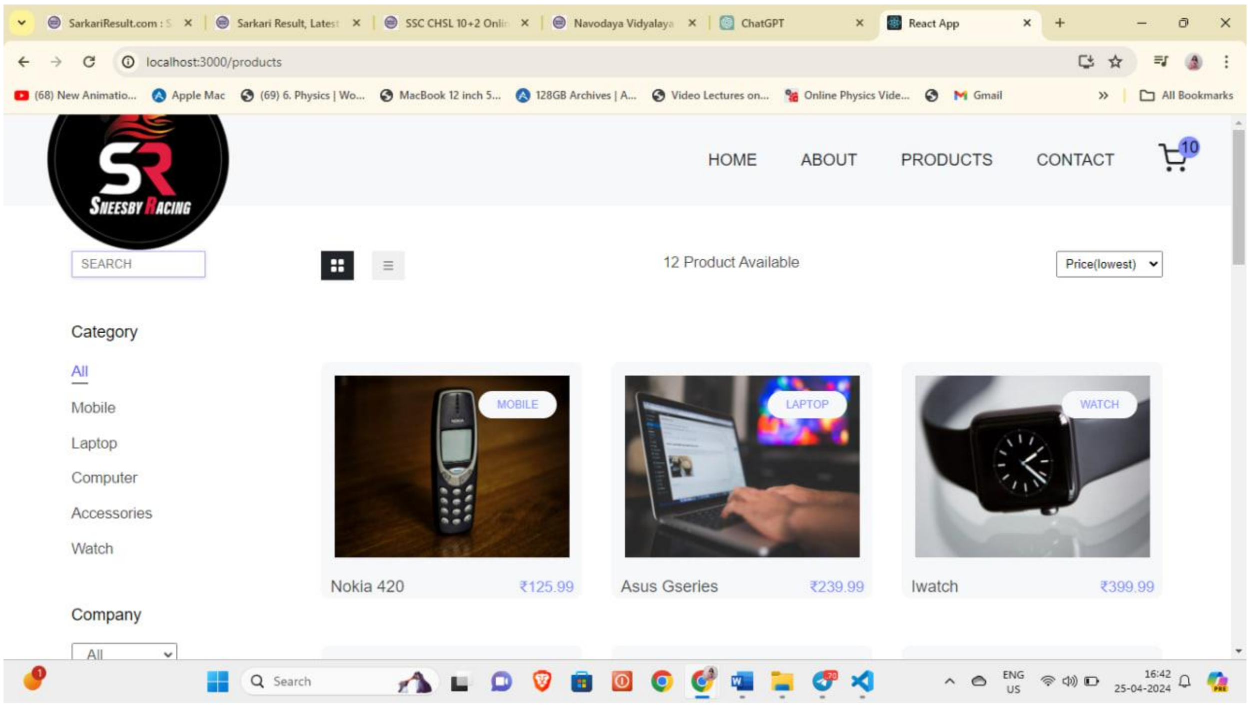
Follow Us

Call Us

+91 9354343678

@2024 RATHORE. All Rights Reserved

PRIVACY POLICY



12 Product Available

Price(lowest)

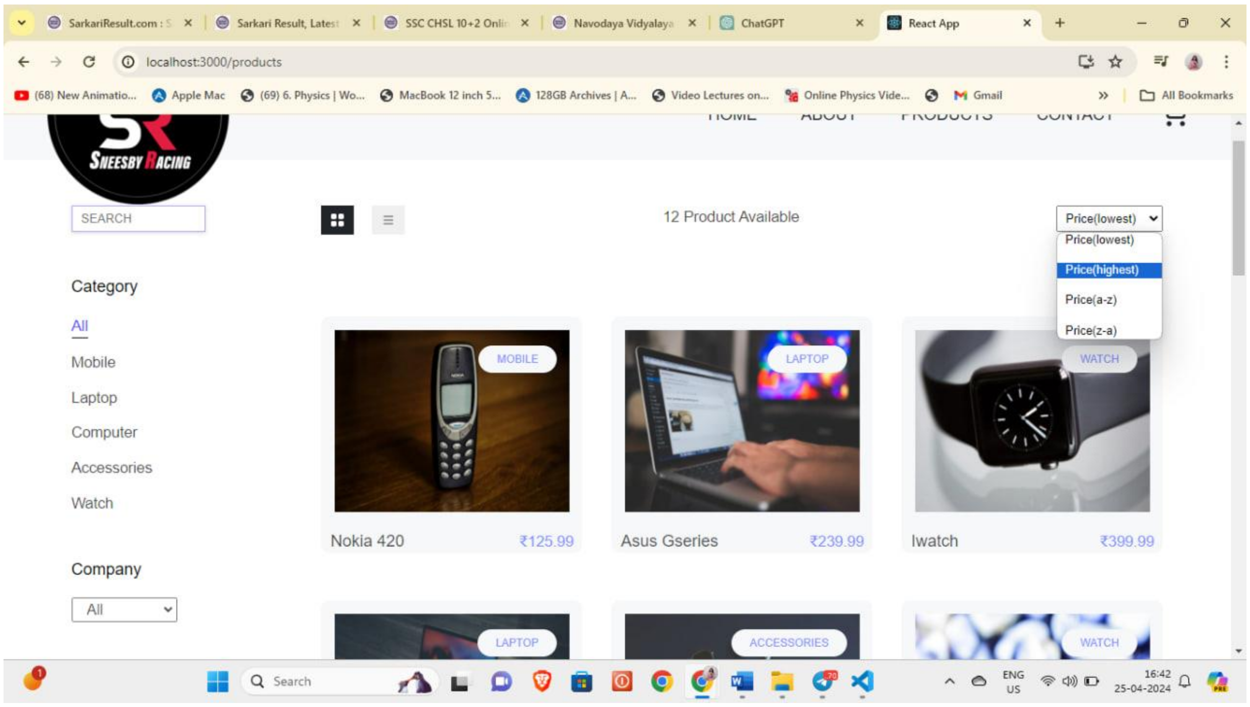
Category

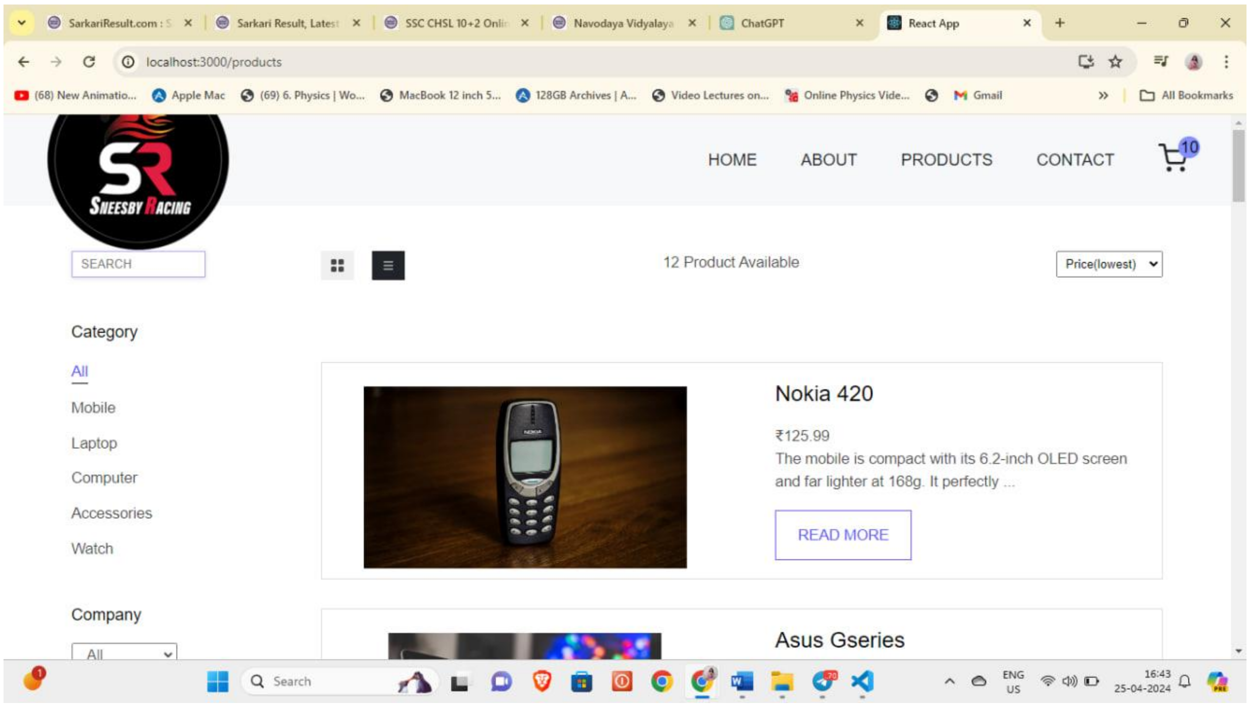
All
Mobile
Laptop
Computer
Accessories
Watch

Company

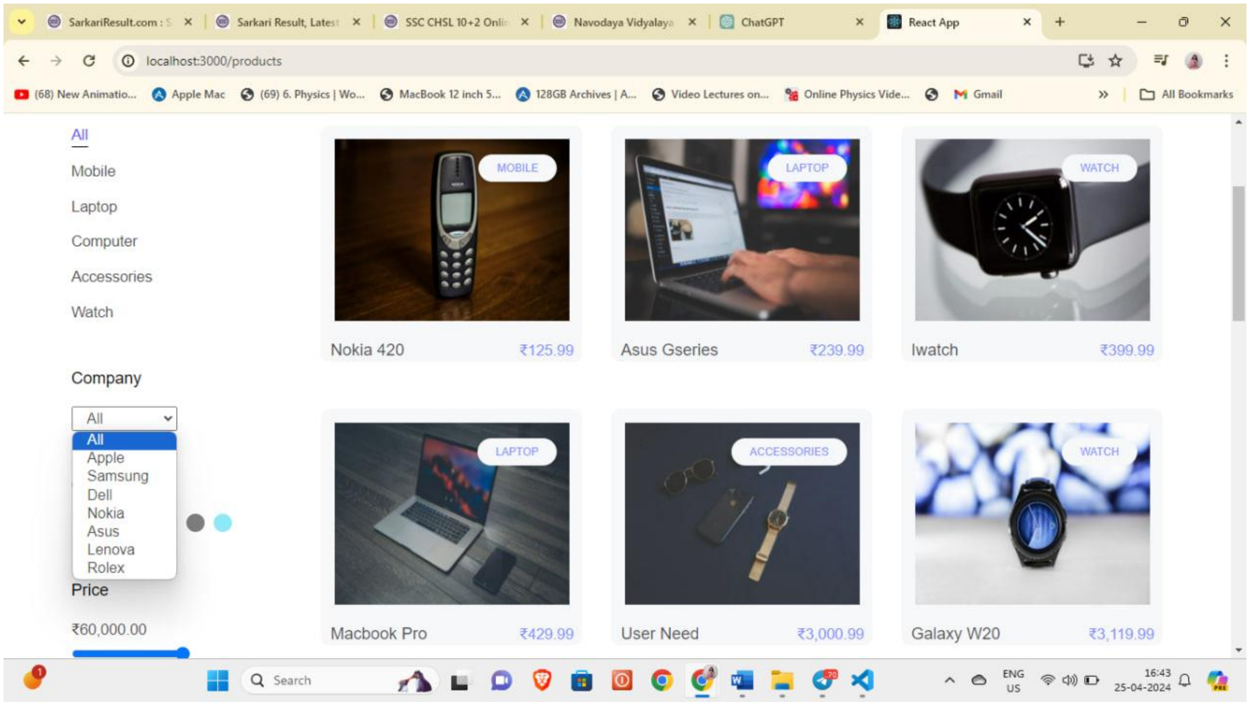
All

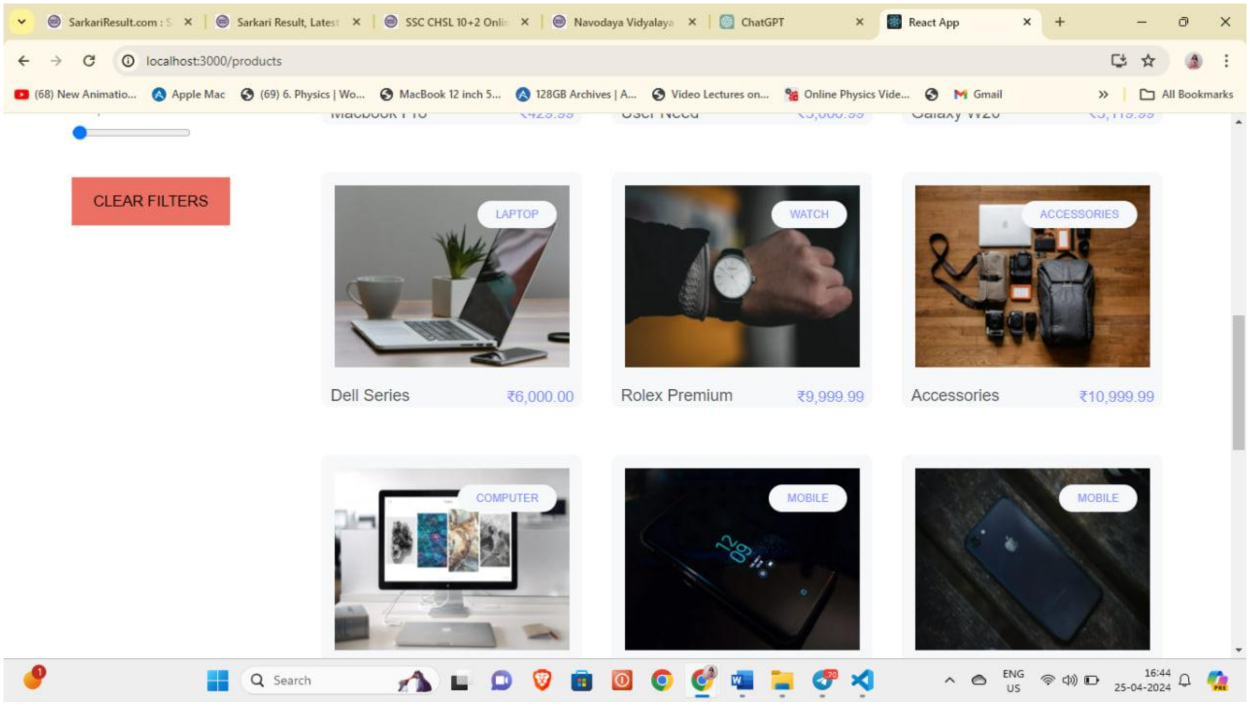
Nokia 420	₹125.99
Asus Gseries	₹239.99
Iwatch	₹399.99





The mobile is compact with its 6.2-inch OLED screen and far lighter at 168g. It perfectly ...


localhost:3000/products
All
Mobile
Laptop
Computer
Accessories
Watch
Company
All
Apple
Samsung
Dell
Nokia
Asus
Lenova
Rolex
Price
₹60,000.00
MOBILE
Nokia 420 ₹125.99
LAPTOP
Asus Gseries ₹239.99
WATCH
Iwatch ₹399.99
LAPTOP
Macbook Pro ₹429.99
ACCESSORIES
User Need ₹3,000.99
WATCH
Galaxy W20 ₹3,119.99


localhost:3000/products
CLEAR FILTERS
LAPTOP
Dell Series ₹6,000.00
WATCH
Rolex Premium ₹9,999.99
ACCESSORIES
Accessories ₹10,999.99
COMPUTER
MOBILE
MOBILE