

Local E-Commerce Platform - Golden Mobile Shop Documentation

Overview

Local E-Commerce Platform is an e-commerce platform designed to showcase and sell mobile phones and accessories. The application features product listings, a shopping cart, a slider on the home page, and a user-friendly interface.

Key Features: -

- **Product Listing:** Displays a variety of mobile phones and accessories.
- **Shopping Cart:** Allows users to add items to their cart and view their selections.
- **Home Slider:** Features promotional content and links to product categories.
- **Responsive Design:** Ensures the application is usable on both desktop and mobile devices.

Installation Guide: -

Prerequisites:

1. **Node.js:** Ensure Node.js is installed on your machine. [Download Node.js](#).
2. **npm:** Comes bundled with Node.js. Use npm to manage project dependencies.

Cloning the Repository:

1. Clone the project repository from GitHub:

```
git clone https://github.com/your-username/golden-mobile-shop.git
```

2. Navigate to the project directory:

```
cd Local E-Commerce Platform
```

Installing Dependencies:

1. Install the required Node.js packages:

```
npm install
```

2. Install Tailwind CSS and its dependencies:

```
npm install tailwindcss
```

Running the Application:

1. Start the development server:

```
npm start
```

This will launch the application in development mode and open it in your default browser.

2. For production build, use:

```
npm run build
```

This will create a build directory with optimized production files.

Running Tests:

1. To run tests, use:

```
npm test
```

Project Structure:

src/

- **Components/**: Contains reusable React components.
 - Card.js: Displays individual product cards.
 - Footer.js: Footer component with contact information.
 - Navbar.js: Navigation bar with links and cart counter.
 - Slider.js: Container for the slide components.
 - Slide.js: Displays individual slides in the slider.
- **pages/**: Contains React components for different pages.
 - Cart.js: Displays the shopping cart.
 - Home.js: Displays the home page with the slider and product list.
 - NotFound.js: Displays a 404 page for undefined routes.
 - Products.js: Displays the list of products.
- **Utilities/**: Contains utility functions.
 - currencyFormatter.js: Utility function to format prices.
- **app/**: Contains Redux-related files.
 - store.js: Configures the Redux store.

- **index.js:** Entry point of the application, sets up routing and Redux provider.
- **App.js:** Main application component with routing setup.
- **index.css:** Global styles including Tailwind CSS imports.

Styling:

- **Tailwind CSS:** Utility-first CSS framework used for styling.
- **Fonts:** Imported from Google Fonts (Poppins and Space Grotesk).

Components:

Navbar.js

- **Function:** Displays the navigation bar with links to home, products, and cart, and shows the number of items in the cart.
- **Usage:** Navbar is used in the main App component and displayed on every page.

Footer.js

- **Function:** Displays the footer with contact information and copyright notice.
- **Usage:** Footer is used in the main App component and displayed on every page.

Slider.js and Slide.js

- **Function:** Provides a carousel of promotional slides with links to product categories.
- **Usage:** Slider contains multiple Slide components and is used on the home page.

Card.js

- **Function:** Displays product details in a card format with an image, name, price, and a link to view more details.
- **Usage:** Card is used in the Products page to list all available products.

Pages:

Home.js

- **Function:** Displays the home page with a slider and a section for product listings.
- **Usage:** The default route (/) renders the Home component.

Products.js

- **Function:** Displays a list of products with Card components.
- **Usage:** The /products route renders the Products component.

Cart.js

- **Function:** Displays the shopping cart with selected items.
- **Usage:** The /cart route renders the Cart component.

NotFound.js

- **Function:** Displays a 404 error message for undefined routes.
- **Usage:** The * route renders the NotFound component.

Screenshots: -

Code:

```
src > JS Appjs > [App] App
1  import { Route, Routes } from "react-router-dom";
2  import { ToastContainer } from "react-toastify";
3  import Footer from "../Components/Footer";
4  import Navbar from "../Components/Navbar";
5  import Cart from "../pages/Cart";
6  import Home from "../pages/Home";
7  import NotFound from "../pages/NotFound";
8  import Products from "../pages/Products";
9  import "react-toastify/dist/ReactToastify.css";
10
11  const App = () => {
12    return (
13      <>
14        <div className="app min-h-screen bg-gray-50 text-gray-700">
15          <Navbar />
16          <ToastContainer />
17          <Routes>
18            <Route path="/" element={<Home />} />
19            <Route path="/products" element={<Products />} />
20            <Route path="/products/:id" element={<Products />} />
21            <Route path="/cart" element={<Cart />} />
22            <Route path="*" element={<NotFound />} />
23          </Routes>
24        </div>
```

src > features > products > JS cartSlice.js > [0] cartSlice

```
1 import { createSlice } from "@reduxjs/toolkit";
2 import { toast } from "react-toastify";
3
4 const initialState = {
5   cartItems: localStorage.getItem("cartItem")
6     ? JSON.parse(localStorage.getItem("cartItems"))
7     : [],
8   cartTotalAmount: 0,
9 };
10
11 const cartSlice = createSlice({
12   name: "cart",
13   initialState,
14   reducers: {
15     addToCart(state, action) {
16       //check if the item is already in the cart
17       const existedItemIndex = state.cartItems.findIndex(
18         (item) => item.id === action.payload.id
19       );
20       //if exist
21       if (existedItemIndex >= 0) {
22         //increase quantity
23         state.cartItems[existedItemIndex].cartQuantity += 1;
24       } else {
25         //add new item to the cart
26       }
27     }
28   }
29 });
```

ar.js JS Slidejs X JS Sliderjs JS cartSlicejs JS productSlicejs JS Cartjs JS Homejs JS NotFoundjs JS Productsjs JS currencyFormatterjs

src > Components > JS Slidejs > [0] Slide

```
1 import { Link } from "react-router-dom";
2 const Slide = ({ image }) => {
3   return (
4     <div className="slide" style={{ backgroundImage: `url(${image.src})` }}>
5       <div className="slide-texts container mx-auto flex flex-col items-start justify-center h-full gap-5 text-violet-50">
6         <h1 className="text-7xl font-bold uppercase space-font w-3/5">
7           {image.headline}
8         </h1>
9         <p className="w-3/5">{image.body}</p>
10        <Link
11          to={`/${products}/${image.category}`}
12          className="slide-btn uppercase border border-violet-50 h-14 w-72 hover:text-orange-50 duration-300 font-medium mt-5">
13          <span className="z-[2] absolute top-[50%] left-[50%] -translate-x-1/2 -translate-y-1/2 w-full text-center">
14            {image.cta}
15          </span>
16        </Link>
17      </div>
18    </div>
19  );
20 }
21 export default Slide;
```

JS Navbarjs X JS Slidejs JS Sliderjs JS cartSlicejs JS productSlicejs JS Cartjs JS Homejs JS NotFoundjs JS Productsjs JS currencyFormatterjs JS Appjs JS

src > Components > JS Navbarjs > [0] Navbar

```
1 import { BsCart3 } from "react-icons/bs";
2 import { useSelector } from "react-redux";
3 import { Link } from "react-router-dom";
4 const Navbar = () => {
5   const { cartItems } = useSelector((state) => state.cart);
6   return (
7     <div className="navban-bg bg-violet-700 text-violet-50 h-20 flex justify-center items-center">
8       <div className="navban container mx-auto flex items-center justify-between">
9         <div className="left">
10           <Link to="/" className="text-xl font-semibold">
11             Golden <span className="text-yellow-500"> Mobile Shop </span>
12           </Link>
13         </div>
14         <div className="right flex items-center gap-5">
15           <Link to="/" className="nav-link">
16             Home
17           </Link>
18           <Link to="/products" className="nav-link">
19             Products
20           </Link>
21           <Link to="/cart">
22             <span className="cart-icon relative">
23               <BsCart3 />
24               <span className="cart-count absolute z-[1] top-3 right-3 text-xs bg-orange-600 h-5 w-5 rounded-full flex items-center justify-center font-medium">
25                 {cartItems.length}
26               </span>
27             </span>
28           </Link>
29         </div>
30       </div>
31     </div>
32  );
33 }
```

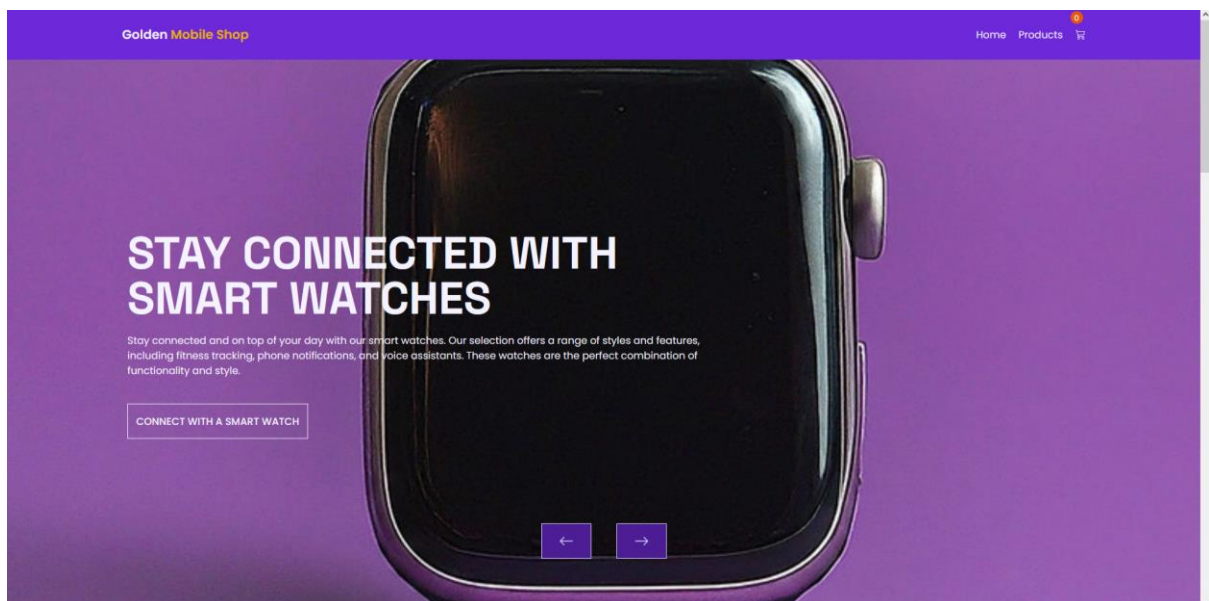
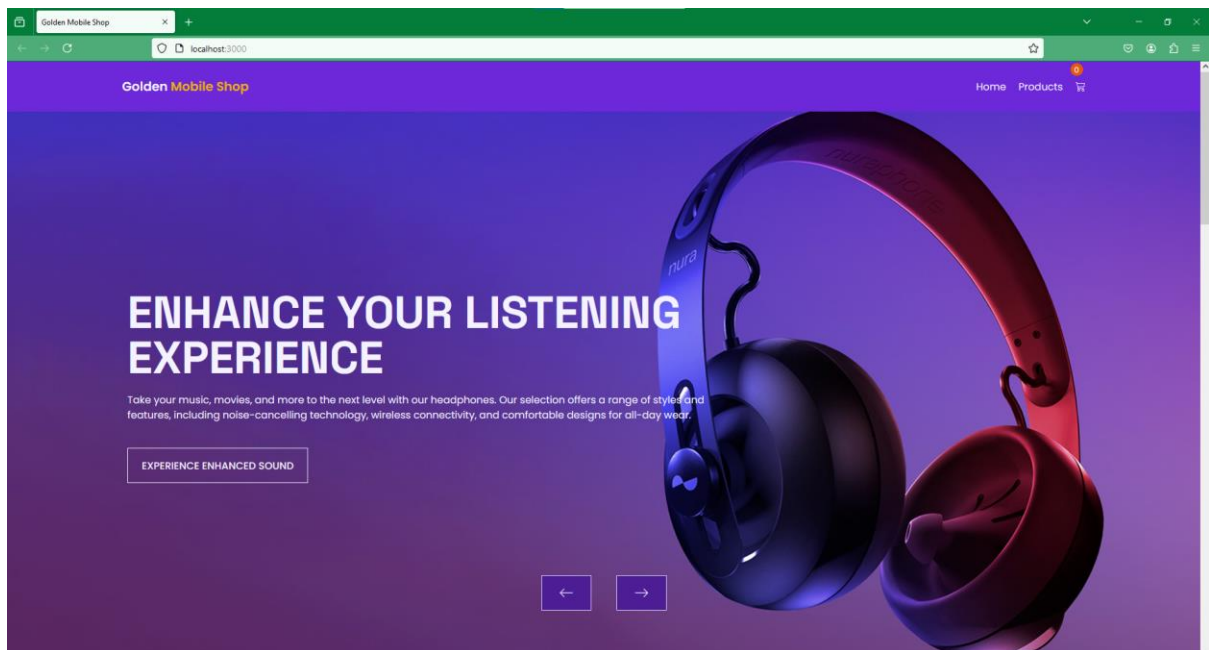
```
JS Navbar.js JS Slide.js JS Slider.js JS cartSlice.js JS productSlice.js JS Cart.js JS Home.js JS NotFound.js JS Products.js JS currencyFormatter.js X JS App.js
src > Utilities > JS currencyFormatter.js > ...
1 export const currencyFormatter = (price) => {
2   if (!price) return;
3   return price.toLocaleString("en-US", { style: "currency", currency: "INR" });
4 };
5
```

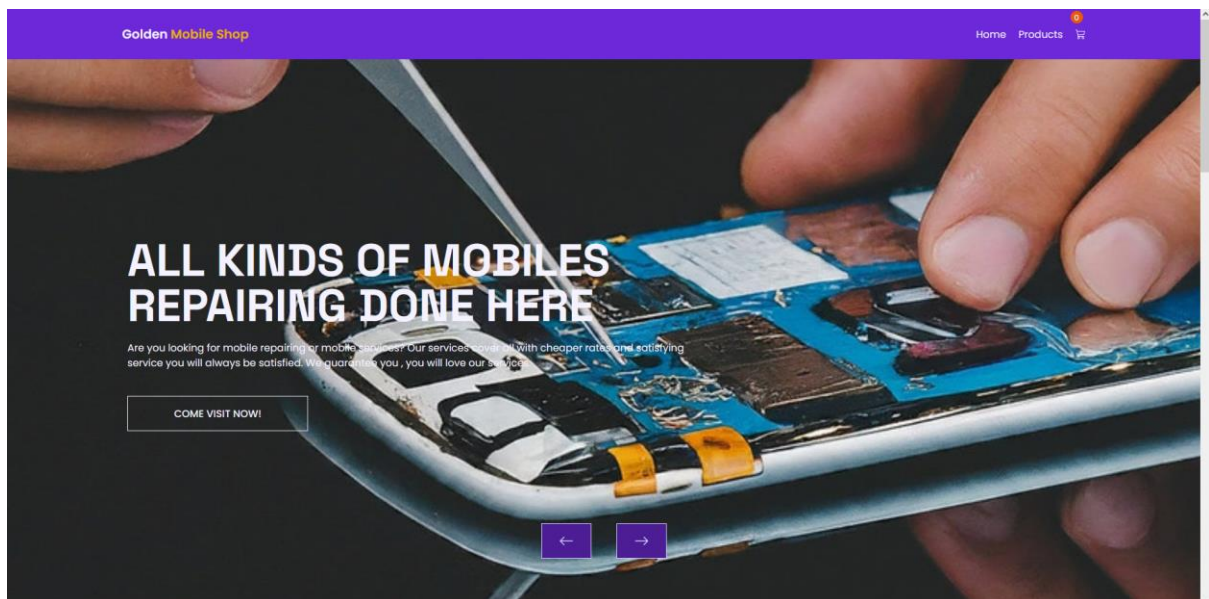
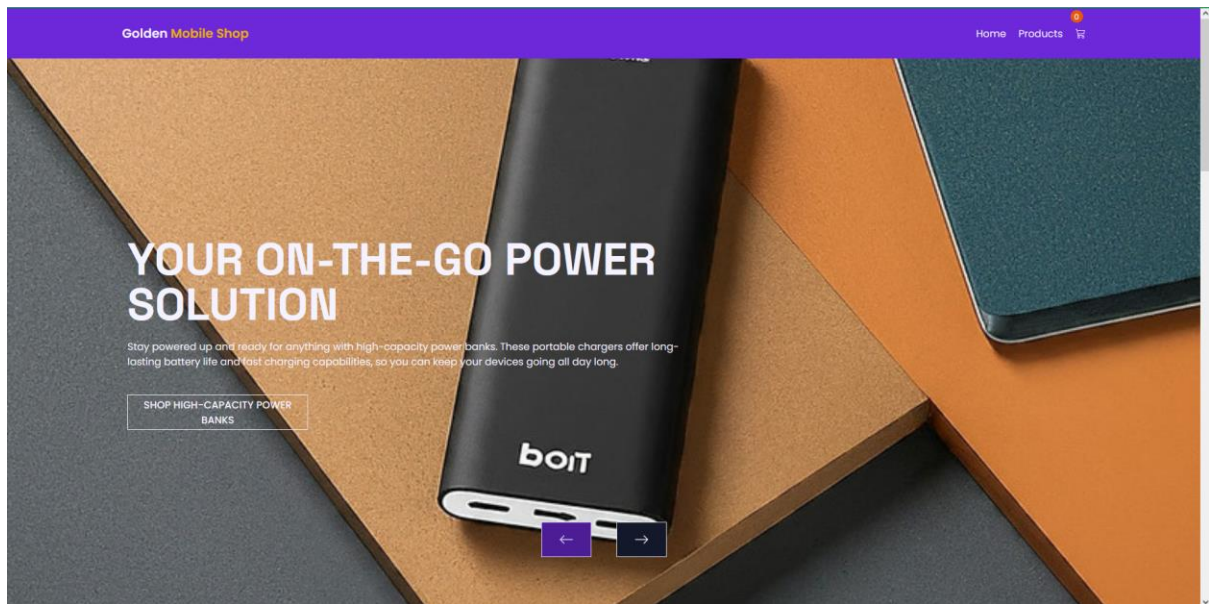
```
JS Navbar.js JS Slide.js JS Slider.js JS cartSlice.js JS productSlice.js
```

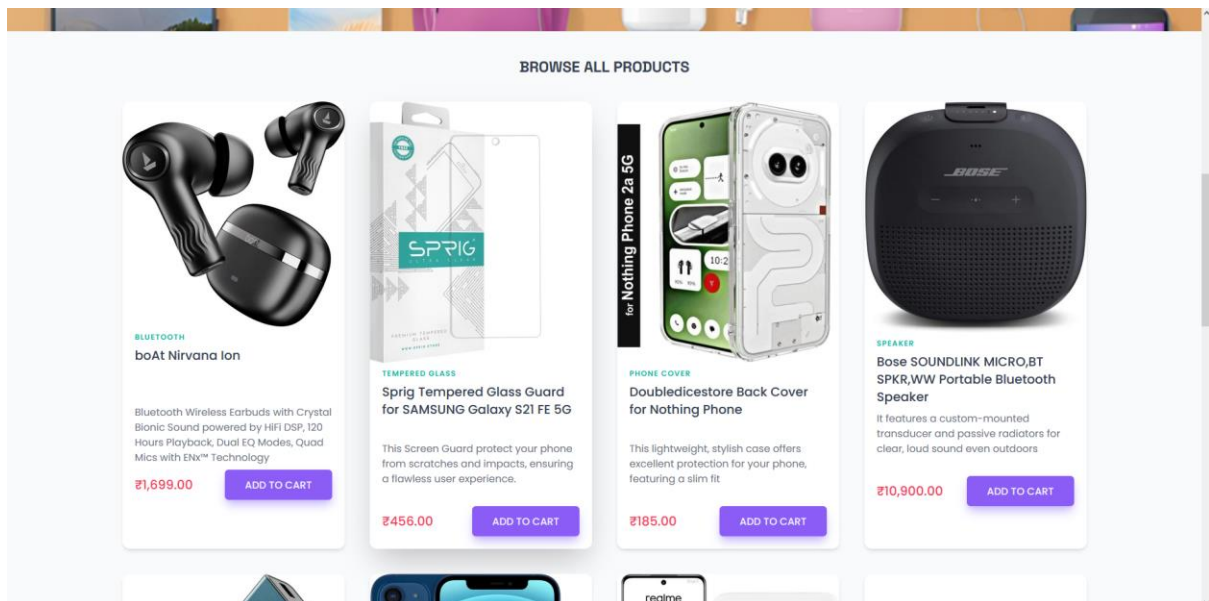
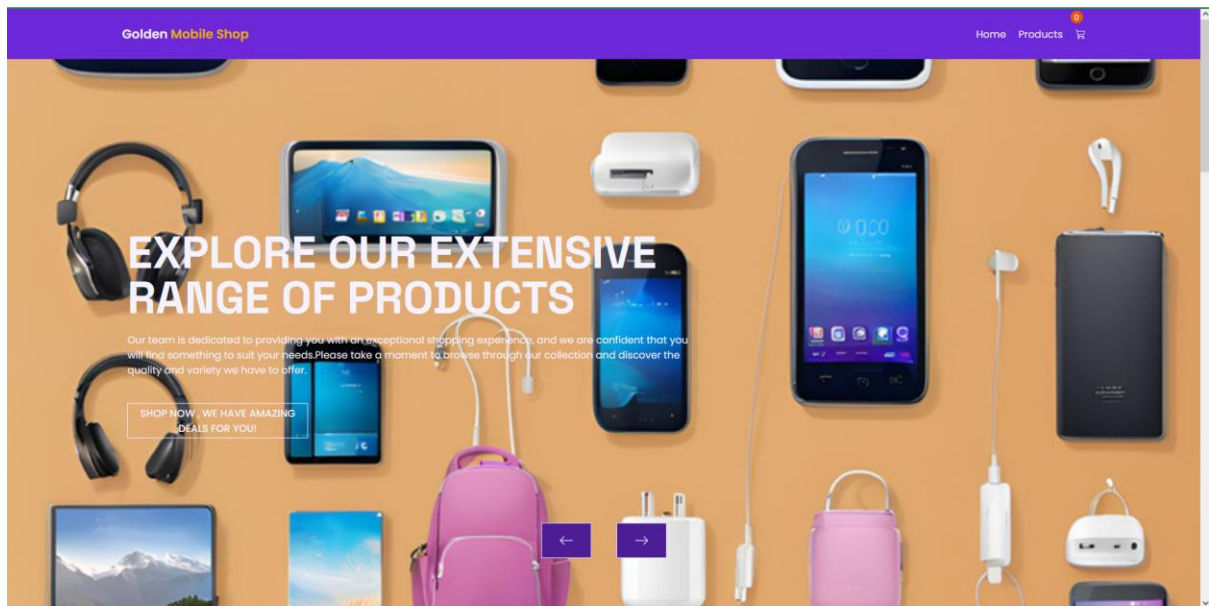
```
src > JS index.js > ...
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import { BrowserRouter } from "react-router-dom";
4 import App from "../App";
5 import "../index.css";
6
7 import { store } from "../app/store";
8 import { Provider } from "react-redux";
9
10 const root = ReactDOM.createRoot(document.getElementById("root"));
11 root.render(
12   <BrowserRouter>
13     <React.StrictMode>
14       <Provider store={store}>
15         <App />
16       </Provider>
17     </React.StrictMode>
18   </BrowserRouter>
19 );
20
```


```
JS tailwind.config.js > ...
1 /** @type {import('tailwindcss').Config} */
2 module.exports = {
3   content: ["./src/**/*.{js,jsx,ts,tsx}"],
4   theme: {
5     extend: {},
6   },
7   plugins: [],
8 };
9
```

Output:










POWER BANK

boAt 20000 mAh 22.5 W Power Bank

Max 22.5W Output, 20W for 2 USB-C port and 22.5W for 1 USB-A port

₹479.99 [ADD TO CART](#)




SMARTPHONE

Apple iPhone 12 (Blue, 64 GB)

It comes with Super Retina XDR Display, A14 Bionic chip to make for blazing-fast performance speeds

₹37,999.00 [ADD TO CART](#)




CHARGER

RoarX 33 W VOOC 6 A Wall Charger for Mobile - White

Offers exceptional charging speeds and QC 3.0 technology ensure efficient charging

₹411.00 [ADD TO CART](#)




MEMORY CARD

SanDisk Ultra 256 GB MicroSDXC Class 10 140 MB/s Memory Card 4.1

It boasts impressive transfer speeds of 150MB/s and an AI-rated performance

₹1,849.00 [ADD TO CART](#)




SanDisk

HP 2 IN1 OTG V220N 128 GB Pen Drive - Silver

Good Quality products Direct connect Mobile with otg

₹480.00 [ADD TO CART](#)




CAR CHARGER

Portronics 3 Amp Turbo Car Charger - Black

90W Fast Charger offers a fast, efficient, ensuring seamless and safe charging on-the-go

₹629.00 [ADD TO CART](#)



EARPHONE

boAt Bassheads 100 Wired Headset - Black

Come with a unique design that is inspired by the hawk and looks extremely trendy


₹329.00 [ADD TO CART](#)

Golden Mobile Shop

[Home](#) [Products](#)

1

YOU'VE ADDED 1 ITEM

PRODUCT	UNIT PRICE	QUANTITY	TOTAL PRICE
<div><div><div><div>REMOVE</div></div></div><div>Doublelicestore Back Cover for Nothing Phone</div></div>	₹185.00	<div>- 1 +</div>	₹185.00

CLEAR CART

Subtotal

₹185.00

Taxes and shipping costs are calculated at the checkout

CHECKOUT

CONTINUE SHOPPING

✔ Item added

Golden Mobile Shop

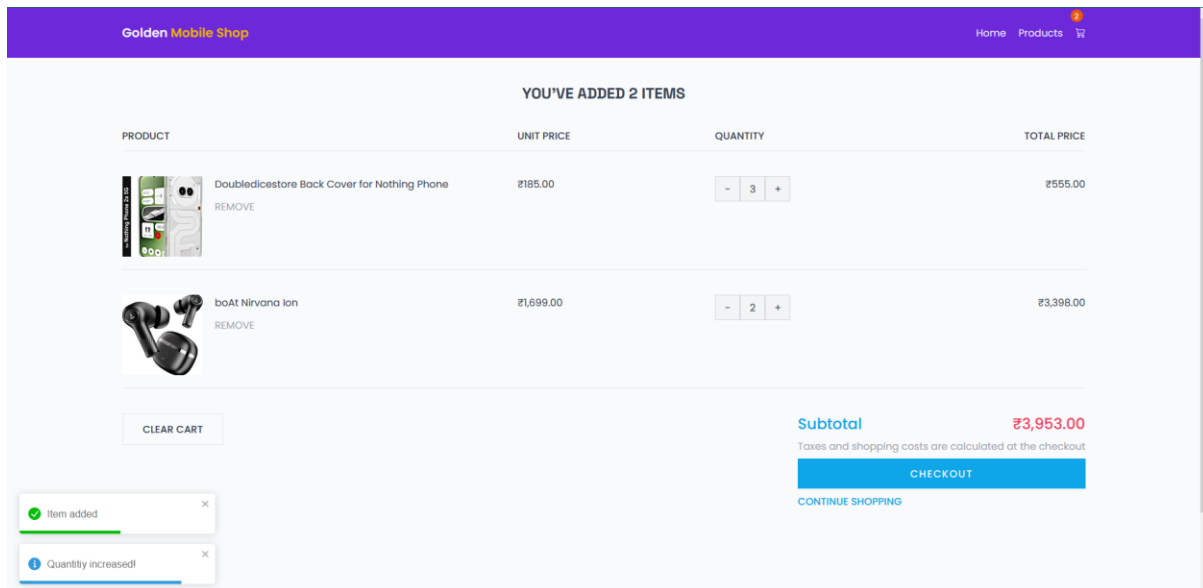
[Home](#) [Products](#)

0

CART IS EMPTY

[Start shopping now](#)

❗ Item removed!



Installation Impact:

- **Ease of Setup:** The installation process is straightforward, involving cloning the repository, installing dependencies, and running the development server.
- **Dependencies:** Includes popular libraries like React, Redux, Tailwind CSS, and Axios, which are well-supported and widely used.

Project Workings:

- **State Management:** Redux manages the state for the shopping cart and product list, ensuring consistency across the application.
- **Routing:** React Router DOM is used for client-side routing, allowing navigation between different pages without full-page reloads.
- **Styling:** Tailwind CSS provides a responsive and customizable styling framework, enhancing the application's UI and UX.

Findings:

- **Component Reusability:** The project structure promotes component reusability, with common components like Navbar and Footer used across multiple pages.
- **Responsive Design:** Tailwind CSS ensures that the application is responsive and looks good on various devices.
- **Scalability:** The use of Redux and React Router DOM makes the application scalable and maintainable.

Conclusion:

The Local E-Commerce Platform is a well-structured and modern application with a clean UI, responsive design, and efficient state management. It is easy to install and set up, and its modular design allows for easy maintenance and scalability.

For any further questions or issues, please refer to the project's GitHub repository.