

Introduction

I have developed an E-Commerce website as my final project, specializing in Music Gadgets. The core objective of this platform is to enable individuals to sell their products online and generate revenue. E-Commerce, fundamentally, entails the buying and selling of goods and services, as well as the transmission of funds or data, facilitated primarily over electronic networks like the internet. Through this website, users can leverage the digital marketplace to showcase their music gadgets, reach a broader audience, and conduct transactions seamlessly.

User-Friendly Interface: Navigating through our website is a breeze. Our intuitive interface allows you to effortlessly browse through categories, compare products, read reviews, and make informed decisions, all from the comfort of your home.

Experience music like never before, with headphones that deliver crystal-clear highs, deep bass, and immersive soundscapes. Whether you're an audiophile, a traveller, or a daily commuter, our website is your gateway to audio nirvana. Elevate your listening experience today.

Objective

Our objective at our headphone products e-commerce website is to become the go-to destination for customers seeking high-quality headphones online. We aim to offer a wide selection of headphones from leading brands, ensuring that customers can find the perfect product to suit their needs and preferences. Our goal is to provide a seamless and convenient shopping experience, with user-friendly navigation, detailed product descriptions, and secure payment options. We strive to deliver exceptional customer service, including prompt assistance with inquiries, fast shipping, and hassle-free returns. Ultimately, we aim to build trust and loyalty with our customers by consistently exceeding their expectations and delivering top-notch headphone products and service.

- This site gives all the information about the e-shopping to provide better service for the customer.
- It provides the facility to the customers who want to shop on-line due to lack of time.
- It provides facility to the customer to payment by the Cash, UPI and Card also.
- It's providing the full details about the product and related information about the product like cost, etc.
- With the help of it we can save the time and money also.
- It provides better security and good delivery service to the customer.

Needs:

Convenience: Consumers increasingly prefer the convenience of shopping online, where they can browse, compare, and purchase products from the comfort of their homes or on-the-go. A headphone e-commerce project fulfils this need by providing a convenient platform for customers to explore and buy headphones anytime, anywhere.

Wide Selection: The headphone market offers a vast array of products with diverse features, styles, and price points. An e-commerce platform allows customers to access a wide selection of headphones from various brands, making it easier for them to find the perfect product that suits their preferences and budget.

Information Access: Before making a purchase decision, customers often seek detailed information about headphones, including specifications, reviews, and recommendations. An e-commerce project provides customers with access to comprehensive product descriptions, user reviews, and expert insights, helping them make informed choices.

Accessibility: Not everyone has easy access to physical retail stores that offer a wide range of headphone options. An e-commerce project addresses this accessibility issue by providing a platform where customers from remote or underserved areas can access and purchase headphones online.

Benefits:

Global Reach: By establishing an online presence, a headphone e-commerce project can reach a global audience, breaking down geographical barriers and tapping into new markets beyond traditional retail footprints.

24/7 Availability: Unlike physical stores with fixed operating hours, an e-commerce platform is accessible 24/7, allowing customers to shop for headphones at their convenience, regardless of time zones or holidays.

Cost-effectiveness: Operating an online store typically incurs lower overhead costs compared to maintaining brick-and-mortar retail locations. With reduced expenses in areas such as rent, utilities, and staffing, an e-commerce project can achieve greater cost-effectiveness and profitability.

Customer Insights: E-commerce platforms generate valuable data about customer behaviour, preferences, and purchasing patterns. By analysing this data, businesses can gain insights into customer trends and preferences, enabling them to tailor marketing strategies, optimize product offerings, and enhance the overall customer experience.

Scalability: As the business grows, an e-commerce project can easily scale its operations to accommodate increasing demand, expand product offerings, and reach new customer segments. This scalability allows businesses to adapt and evolve in response to market dynamics and changing consumer preferences.

Customer Engagement: Through features such as live chat support, personalized recommendations, and loyalty programs, an e-commerce project can foster greater engagement and interaction with customers, building brand loyalty and driving repeat purchases over time.

Software and hardware

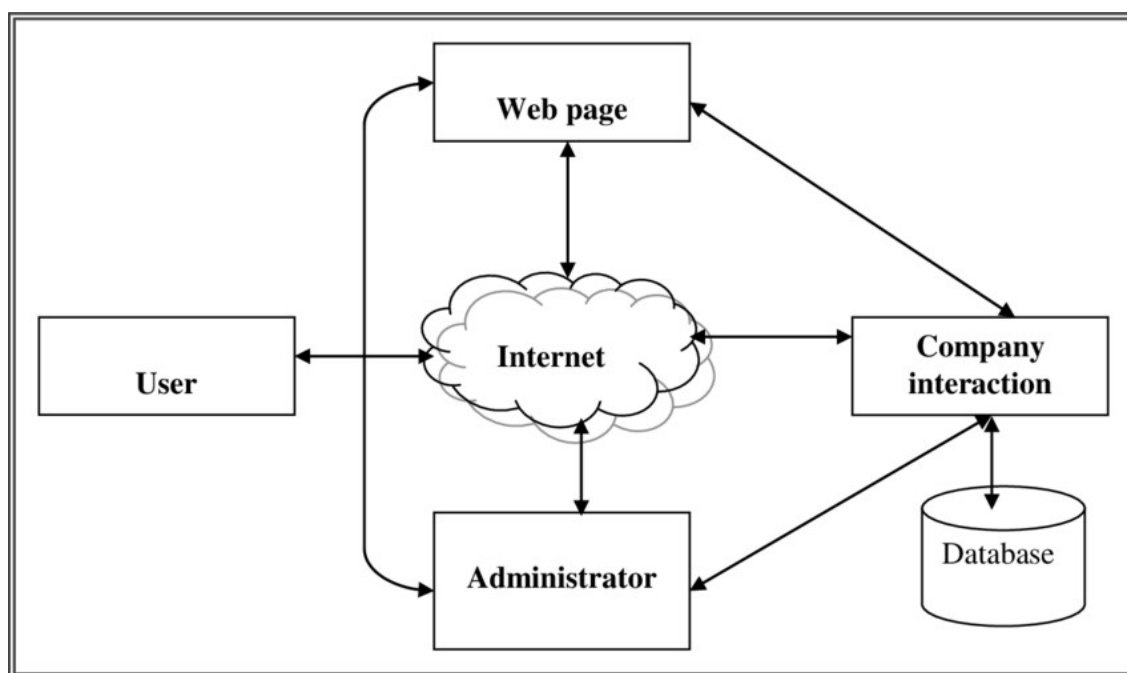
Software:

- **E-commerce Platform:** Dedicated e-commerce platforms such as Shopify, Flipkart, or Amazon offer features specifically tailored for online retail, including shopping cart functionality, payment gateways integration, and order management.
- **Frontend Development:** React.js are used to create the user interface (UI) and ensure responsive design for compatibility across various devices.
- **Backend Development:** Strapi allows you to easily build customizable backend services.
- **Database Management:** Cloudinary and PostgreSQL are commonly used to store product information, customer data, and transaction records securely.
- **Payment Gateway Integration:** APIs provided by payment gateways Stripe are integrated to facilitate secure online transactions.

Hardware:

- **Processor** : Ryzen 5
- **Ram** : 8 GB
- **Storage** : 100 GB SSD
- **Network Adapter** : Ethernet Adapter
- **Others** : Display, Printer etc.

System Design



Implementation and maintenance

Implementation:

Select a Technology Stack: Choose the appropriate technologies for frontend, backend, and database based on your requirements, team expertise, and scalability needs.

Design Database Schema: Create a database schema to store product information, user data, orders, and other relevant entities. Normalize the database structure to ensure efficiency and data integrity.

Develop Frontend: Build the user interface using HTML, CSS, and JavaScript along with a modern frontend framework like React.js, Angular, or Vue.js. Implement features such as product listing, search, filtering, and cart functionality.

Implement Backend Logic: Develop server-side logic using a backend framework like Django, Express.js, or Spring Boot. Create RESTful APIs to handle user authentication, product management, cart management, order processing, and payment integration.

Integrate Payment Gateway: Integrate with a payment gateway provider like Stripe, PayPal, or Square to enable secure online transactions. Implement payment processing logic to handle payments, refunds, and order fulfillment.

Optimize Performance: Optimize website performance by minimizing load times, reducing server requests, caching static assets, and employing content delivery networks (CDNs) for efficient content delivery.

Test the Website: Conduct thorough testing to ensure functionality, usability, compatibility, and security across different devices, browsers, and screen sizes. Perform unit tests, integration tests, and end-to-end tests to identify and fix bugs.

Deploy to Production: Deploy the website to a production environment using a reliable hosting provider or cloud platform. Set up monitoring tools to track website performance, uptime, and user activity.

Maintenance:

Regular Updates: Keep the website up-to-date by applying security patches, bug fixes, and feature enhancements to both frontend and backend components.

Monitor Performance: Continuously monitor website performance metrics such as page load times, server response times, and error rates. Use monitoring tools to detect and address performance issues promptly.

Backup Data Regularly: Implement regular backups of the website's database and files to prevent data loss in case of hardware failures, security breaches, or accidental deletions.

Security Audits: Conduct periodic security audits and vulnerability assessments to identify and mitigate potential security risks. Stay informed about the latest security threats and industry best practices.

Scale Infrastructure: Scale website infrastructure as needed to accommodate growing traffic and ensure high availability during peak periods. Consider horizontal scaling by adding more servers or vertical scaling by upgrading server resources.

Customer Support: Provide responsive customer support to address user inquiries, resolve issues, and assist with order management, returns, and refunds.

SEO and Marketing: Optimize the website for search engines (SEO) to improve visibility and attract organic traffic. Implement digital marketing strategies such as email campaigns, social media promotions, and affiliate partnerships to drive sales and customer engagement.

User Feedback: Gather feedback from users through surveys, reviews, and analytics data to identify areas for improvement and enhance the user experience.

Testing

1. Functional Testing:

User Flows: Test various user flows such as browsing products, adding items to the cart, proceeding to checkout, making payments, and viewing order history.

Search Functionality: Verify that the search feature returns accurate results for different queries, including product names, categories, and attributes.

Product Details: Ensure that product details pages display accurate information such as descriptions, images, prices, and specifications.

Cart and Checkout: Test the functionality of the shopping cart, including adding/removing items, updating quantities, applying discounts, and calculating totals. Verify the checkout process, including shipping options, payment methods, and order confirmation.

User Accounts: Test user registration, login, password recovery, and profile management features.

2. Usability Testing:

Navigation: Evaluate the website's navigation structure, menu layout, and ease of finding products. Ensure intuitive navigation across different sections of the website.

Responsive Design: Test the website's responsiveness on various devices (desktops, laptops, tablets, smartphones) and screen sizes. Ensure that the layout and content adapt seamlessly to different resolutions.

Accessibility: Assess the website's accessibility for users with disabilities by testing keyboard navigation, screen reader compatibility, and color contrast.

Error Handling: Test error messages and validation prompts to ensure they are clear, descriptive, and help users resolve issues effectively.

3. Performance Testing:

Load Testing: Simulate heavy user traffic to evaluate the website's performance under peak load conditions. Measure server response times, throughput, and resource utilization to identify bottlenecks.

Page Load Speed: Measure and optimize page load times to enhance user experience and SEO rankings. Minimize image sizes, optimize code, and leverage caching mechanisms.

4. Security Testing:

Data Encryption: Ensure sensitive data such as user credentials, payment information, and personal details are encrypted during transmission and storage.

Authentication and Authorization: Test user authentication mechanisms to prevent unauthorized access to user accounts and sensitive data.

Payment Security: Verify that payment transactions are processed securely using HTTPS encryption and comply with PCI DSS standards.

5. Compatibility Testing:

Browser Compatibility: Test the website on different web browsers (Chrome, Firefox, Safari, Edge, etc.) and ensure consistent functionality and appearance.

Device Compatibility: Test the website on various devices (desktops, laptops, tablets, smartphones) and operating systems (Windows, macOS, iOS, Android) to ensure compatibility and responsiveness.

6. Content Testing:

Product Data Accuracy: Verify that product information such as descriptions, images, prices, and availability is accurate and up-to-date.

7. Performance Monitoring and Maintenance:

Implement monitoring tools to continuously monitor website performance, uptime, and user activity.

Regularly update and maintain the website to address security vulnerabilities, performance issues, and user feedback.

Coding

Cart

```
import React, { useContext } from "react";
import { MdClose } from "react-icons/md";
import { BsCartX } from "react-icons/bs";
import { Context } from "../../utils/context";
import CartItem from "../CartItem/CartItem";
import { loadStripe } from "@stripe/stripe-js";
import { makePaymentRequest } from "../../utils/api";

import "../Cart.scss";
const Cart = () => {
  const { cartItems, setShowCart, cartSubTotal } =
    useContext(Context);

  const stripePromise = loadStripe(
    process.env.REACT_APP_STRIPE_PUBLISHABLE_KEY
  );

  const handlePayment = async () => {
    try {
      const stripe = await stripePromise;
      const res = await makePaymentRequest.post("/api/orders", {
        products: cartItems,
      });
      await stripe.redirectToCheckout({
        sessionId: res.data.stripeSession.id,
      });
    } catch (err) {
      console.log(err);
    }
  };
};
```



```

return (
  <div className="cart-panel">
    <div
      className="opac-layer"
      onClick={() => setShowCart(false)}
    ></div>
    <div className="cart-content">
      <div className="cart-header">
        <span className="heading">Shopping Cart</span>
        <span
          className="close-btn"
          onClick={() => setShowCart(false)}
        >
          <MdClose />
          <span className="text">close</span>
        </span>
      </div>

      {!cartItems.length && (
        <div className="empty-cart">
          <BsCartX />
          <span>No products in the cart.</span>
          <button className="return-cta" onClick={() =>
            RETURN TO SHOP
          >
            </button>
          </div>
        )}

        {!!cartItems.length && (
          <>
            <CartItem />
            <div className="cart-footer">
              <div className="subtotal">

```

```

        <span
className="text">Subtotal:</span>
        <span className="text total">
            &#8377;{cartSubTotal}
        </span>
    </div>
    <div className="button">
        <button
            className="checkout-cta"
            onClick={handlePayment}
        >
            Checkout
        </button>
    </div>
</div>
</>
    )}
</div>
</div>
    );
};

export default Cart;

```

Cart SCSS

```
@import "../../../css-config/mixins.scss";

.cart-panel {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: 99;
  display: flex;
  justify-content: flex-end;

  .opac-layer {
    background: rgba(0, 0, 0, 0.5);
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
  }

  .cart-content {
    width: 100%;
    height: 100%;
    background: white;
    position: relative;
    z-index: 1;
    display: flex;
    flex-direction: column;
    transform: translateX(100%);
    animation: slideCartWindow 0.3s ease forwards;
    @include md {
      width: 340px;
    }
  }
}
```

```

.cart-header {
  display: flex;
  align-items: center;
  justify-content: flex-end;
  padding: 20px 15px;
  border-bottom: 1px solid rgba(0, 0, 0, 0.1);
  .heading {
    flex-grow: 1;
    margin-bottom: 0;
    font-size: 20px;
    font-weight: 700;
    text-transform: uppercase;
  }
  .close-btn {
    display: flex;
    align-items: center;
    gap: 5px;
    cursor: pointer;
    svg {
      font-size: 18px;
    }
    .text {
      text-transform: uppercase;
      font-size: 14px;
    }
    &:hover {
      opacity: 0.5;
    }
  }
}

.empty-cart {
  display: flex;

```

```

    flex-direction: column;
    align-items: center;
    gap: 20px;
    margin-top: 100px;
    svg {
        font-size: 120px;
        opacity: 0.1;
    }
    .return-cta {
        outline: 0;
        border: 0;
        height: 40px;
        width: 150px;
        display: flex;
        align-items: center;
        justify-content: center;
        cursor: pointer;
        font-size: 13px;
        color: white;
        background: #8e413f;
        border-bottom: 3px solid #662e2c;
        svg {
            margin-right: 10px;
        }
    }
}

.cart-footer {
    border-top: 1px solid rgba(0, 0, 0, 0.1);
    .subtotal {
        padding: 20px 15px;
        border-bottom: 1px solid rgba(0, 0, 0, 0.1);
        display: flex;
        justify-content: space-between;
    }
}

```

```

        .text {
            margin-bottom: 0;
            font-size: 20px;
            font-weight: 700;
            text-transform: uppercase;
            &.total {
                color: #8e413f;
            }
        }
    .button {
        padding: 20px 15px;
        .checkout-cta {
            outline: 0;
            border: 0;
            height: 50px;
            width: 100%;
            display: flex;
            align-items: center;
            justify-content: center;
            cursor: pointer;
            font-size: 16px;
            color: white;
            background: #8e413f;
            border-bottom: 3px solid #662e2c;
            text-transform: uppercase;
            svg {
                margin-right: 10px;}}}}}}
}

@keyframes slideCartWindow {
    0% {
        transform: translateX(100%);
    }
    100% {
        transform: translateX(0);}}

```

Cart Item

```
import React, { useContext } from "react";
import { Context } from "../../utils/context";
import { MdClose } from "react-icons/md";
import "./CartItem.scss";
const CartItem = () => {
  const { cartItems, handleRemoveFromCart, handleCartProductQuantity } =
    useContext(Context);
  return (
    <div className="cart-products">
      {cartItems?.map((item) => (
        <div
          className="search-result-item"
          key={item.id}
          onClick={() => {}}
        >
          <div className="image-container">
            <img
              src={
                process.env.REACT_APP_DEV_URL +
                item.attributes.image.data[0].attributes.url
              } alt=""
            />
          </div>
          <div className="prod-details">
            <span
              className="name">{item.attributes.title}</span>
            <MdClose
              className="close-btn"
              onClick={() => handleRemoveFromCart(item)}
            />
            <div className="quantity-buttons">
```

```

        <span
            onClick={() =>
                handleCartProductQuantity("dec",
item)
            }
        >
    </span>
    <span>{item.attributes.quantity}</span>
    <span
        onClick={() =>
            handleCartProductQuantity("inc",
item)
        }
    >
        +
    </span>
</div>
<div className="text">
    <span>{item.attributes.quantity}</span>
    <span>x</span>
    <span className="highlight">
        <span>&#8377;</span>
        {item.attributes.price *
            item.attributes.quantity}
    </span>
</div>
</div>
</div>
    )})
</div>
);
};

export default CartItem;

```


Cart Item SCSS

```
.cart-products {
  flex-grow: 1;
  .search-result-item {
    padding: 20px 15px;
    display: flex;
    gap: 10px;
    .image-container {
      background-color: rgba(0, 0, 0, 0.05);
      width: 60px;
      height: 60px;
      flex-shrink: 0;
      img {
        width: 100%;
        height: 100%;
      }
    }
  }
  .prod-details {
    overflow: hidden;
    position: relative;
    .name {
      text-overflow: ellipsis;
      white-space: nowrap;
      overflow: hidden;
      font-size: 16px;
      line-height: 1;
      margin-bottom: 10px;
      font-weight: 600;
      display: block;
      padding-right: 25px;
    }
    .text {
      display: flex;
```

```

        align-items: center;
        gap: 5px;
        font-size: 12px;
        font-weight: 600;
        .highlight {
            color: #8e413f;
        }
    }
    .quantity-buttons {
        width: fit-content;
        display: flex;
        border: 1px solid rgba(0, 0, 0, 0.1);
        height: 30px;
        margin-bottom: 8px;
        span {
            font-size: 14px;
            width: 30px;
            display: flex;
            align-items: center;
            justify-content: center;
            cursor: pointer;
            color: #6b6b6b;
            &:nth-child(1) {
                font-size: 18px;
                border-right: 1px solid rgba(0, 0, 0, 0.1);
            }
            &:nth-child(2) {
                width: 40px;
            }
            &:nth-child(3) {
                font-size: 18px;
                border-left: 1px solid rgba(0, 0, 0, 0.1);
            }
        }
    }
    &:hover {
        background-color: rgba(0, 0, 0, 0.05);
    }
}

```

Category

```
import { useParams } from "react-router-dom";
import useFetch from "../../hooks/useFetch";
import Products from "../../Products/Products";
import "../Category.scss";
const Category = () => {
  const { id } = useParams();
  const { data } = useFetch(
    `/api/products?populate=*${filters}[categories][id]=${id}`
  );
  return (
    <div className="category-main-content">
      <div className="layout">
        <div className="category-title">
          {
            data?.data?.[0]?.attributes?.categories?.data?.[0]
              ?.attributes?.title
          }
        </div>
        <Products innerPage={true} products={data} />
      </div>
    </div>
  );
};

export default Category;
```

Category SCSS

```
@import "../../css-config/mixins.scss";
```

```
.category-main-content{
  margin: 30px 0;
  @include md{
    margin: 75px 0;
  }
  .layout{
    max-width: calc(100% - 20px);
    margin: 0 auto;
    @include md {
      max-width: 1200px;
    }
  }
  .category-title{
    font-size: 24px;
    @include md {
      font-size: 34px;
    }
  }
  .products-container{
    margin: 20px 0;
    @include md {
      margin: 50px 0;
    }
  }
}
```

Footer

```
import React from "react";
import "./Footer.scss";
import { FaLocationArrow, FaMobileAlt, FaEnvelope } from "react-
icons/fa";
import Payment from "../../assets/payments.png";
const Footer = () => {
  return (
    <footer className="footer">
      <div className="footer-content">
        <div className="col">
          <div className="title">About</div>
          <div className="text">
            Lorem ipsum dolor sit amet consectetur adipisicing
elit. Harum quisquam vel possimus quaerat, facere repellat commodi
iusto unde eum cupiditate.
          </div>
        </div>
        <div className="col">
          <div className="title">Contact</div>
          <div className="c-item">
            <FaLocationArrow />
            <div className="text">
              Maharana Pratap Nagar, Bhopal, Madhya Pradesh (462011)
            </div>
          </div>
          <div className="c-item">
            <FaMobileAlt />
            <div className="text">
              Phone : +91 9876543210
            </div>
          </div>
          <div className="c-item">
            <FaEnvelope />

```

```

        <div className="text">
            Email : web@echo.com
        </div></div></div>
    <div className="col">
    <div className="title">Categories</div>
        <span className="text">Headphones</span>
        <span className="text">Smart Watches</span>
        <span className="text">Bluetooth Speakers</span>
        <span className="text">Wireless Earbuds</span>
        <span className="text">Earphones</span>
        <span className="text">Home Theatre</span>
    </div>
    <div className="col">
    <div className="title">Pages</div>
        <span className="text">Home</span>
        <span className="text">About</span>
        <span className="text">Privacy Policy</span>
        <span className="text">Returns</span>
        <span className="text">Terms & Conditions</span>
        <span className="text">Contact Us</span>
    </div>    </div>
    <div className="bottom-bar">
        <div className="bottom-bar-content">
            <div className="text">
                ECHO 2024 E-Commerce Website for Music Gadgets.
            </div>
            <img src={Payment} alt=""/>
        </div>    </div>
    </footer>
    );
};
export default Footer;

```

Footer SCSS

```
@import "../css-config/mixins.scss";

.footer{
  width: 100%;
  .footer-content{
    padding: 50px 20px;
    max-width: 1200px;
    margin: 0 auto;
    display: flex;
    flex-flow: wrap;
    gap: 20px;
    @include md {
      justify-content: space-between;
      padding: 50px 0;
    }
  }
  .col{
    max-width: 300px;
    &:nth-child(3) {
      width: 150px;
      @include md {
        width: auto;
      }
    }
  }
  .title{
    margin-bottom: 20px;
    font-size: 20px;
  }
  .text{
    color: rgba(0, 0, 0, 0.5);
    font-size: 14px;
  }
}
```

```

.c-item{
  display: flex;
  margin-bottom: 10px;
  svg{
    flex-shrink: 0;
    font-size: 14px;
    margin-right: 10px;
    margin-top: 4px;
    color: rgba(0, 0, 0, 0.5);}}
span{
  &.text{
    display: block;
    margin-bottom: 10px;
    cursor: pointer;}}}}
.bottom-bar{
  border-top: 1px solid rgba(0, 0, 0, 0.5);
  .bottom-bar-content{
    padding: 20px;
    display: flex;
    align-items: center;
    flex-direction: column;
    text-align: center;
    gap: 10px;
    @include md {
      padding: 0;
      height: 60px;
      max-width: 1200px;
      margin: 0 auto;
      flex-direction: row;
      justify-content: space-between;
      text-align: left;
    }
  }
}

```


Newsletter

```
import React from "react";
import { FaFacebookF, FaTwitter, FaInstagram, FaLinkedinIn, } from
"react-icons/fa";
import "./Newsletter.scss";
const Newsletter = () => {
  return <div className="newsletter-section">
    <div className="newsletter-content">
      <span className="small-text">Newsletter</span>
      <span className="big-text">Sign up for latest updates and
offers</span>
      <div className="form">
        <input type="text" placeholder="Email Address"/>
        <button>Subscribe</button>
      </div>
      <div className="text">Will be used in accordance with our
Privacy Policy</div>
      <div className="social-icons">
        <div className="icon">
          <FaFacebookF size={14} />
        </div>
        <div className="icon">
          <FaTwitter size={14} />
        </div>
        <div className="icon">
          <FaInstagram size={14} />
        </div>
        <div className="icon">
          <FaLinkedinIn size={14} />
        </div></div></div> </div>;
};
export default Newsletter;
```

Newsletter SCSS

```
@import "../../../css-config/mixins.scss";

.newsletter-section{
  width: 100%;
  height: 400px;
  display: flex;
  align-items: center;
  background: #f9f9f9 url("../../assets/newsletter-bg.jpg");
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;

  .newsletter-content{
    width: fit-content;
    display: flex;
    flex-direction: column;
    align-items: center;
    margin: 0 auto;
    text-align: center;
  }

  .small-text{
    margin-bottom: 15px;
    text-transform: uppercase;
    font-size: x-large;
    color: rgba(255, 255, 255, 0.5);
  }

  .big-text{
    margin-bottom: 20px;
    font-size: 22px;
    color: white;
    line-height: 30px;
    font-weight: 300;
  }
}
```

```

    text-transform: uppercase;
    padding: 0 30px;
    @include md {
        font-size: 28px;
    }
}
.form{
    display: flex;
    gap: 5px;
    margin-bottom: 10px;
    input{
        width: 200px;
        height: 40px;
        border-radius: 0;
        border: 1px solid rgba(0, 0, 0, 0.2);
        padding: 0 12px;
        font-size: 16px;
        outline: 0;
        @include md {
            width: 300px;
        }
    }
}
button{
    outline: 0;
    border: 0;
    height: 40px;
    width: 100px;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    font-size: 16px;
    color: white;

```

```

        background-color: #8e413f;
        border-bottom: 3px solid #662e2c;
        @include md {
            width: 120px;
        }
    }
}

.text{
    margin-bottom: 20px;
    font-size: 14px;
    color: rgb(255, 255, 255);
}

.social-icons{
    display: flex;
    gap: 10px;
    .icon{
        width: 30px;
        height: 30px;
        background-color: rgba(0, 0, 0, 0.8);
        border-radius: 50%;
        color: white;
        display: flex;
        align-items: center;
        justify-content: center;
        cursor: pointer;
    }
}
}
}

```

Header

```
import { useEffect, useState, useContext } from "react";
import { useNavigate } from "react-router-dom";
import { TbSearch } from "react-icons/tb";
import { CgShoppingCart } from "react-icons/cg";
import { AiOutlineHeart } from "react-icons/ai";
import "./Header.scss";
import Search from "../Search/Search";
import { Context } from "../../utils/context";
import Cart from "../Cart/Cart";

const Header = () => {
  const [scrolled, setScrolled] = useState(false);
  const [searchModal, setSearchModal] = useState(false);
  const navigate = useNavigate();
  const handleScroll = () => {
    const offset = window.scrollY;
    if (offset > 200) {
      setScrolled(true);
    } else {
      setScrolled(false);
    }
  };

  useEffect(() => {
    window.addEventListener("scroll", handleScroll);
  }, []);

  const { cartCount, showCart, setShowCart } = useContext(Context);

  return (
    <>
      <header
```

```

        className={`main-header ${scrolled ? "sticky-header" :
""}`}
        >
        <div className="header-content">
            <ul className="left">
                <li onClick={() => navigate("/")}>Home</li>
                <li onClick={() =>
navigate("/about")}>About</li>
                <li onClick={() =>
navigate("category/8")}>Categories</li>
            </ul>
            <div className="center" onClick={() =>
navigate("/")}>
                ECHO.
            </div>
            <div className="right">
                < TbSearch onClick={() => setSearchModal(true)}

                < AiOutlineHeart />
                <span
                    className="cart-icon"
                    onClick={() => setShowCart(true)}
                >
                    < CgShoppingCart />
                    {!!cartCount && <span>{cartCount}</span>}
                </span>
            </div></div>
        </header>
        {searchModal && <Search setSearchModal={setSearchModal}

        />}
        {showCart && <Cart />}
    </>
    );
};

export default Header;

```

Header SCSS

```
@import "../css-config/mixins.scss";
```

```
.main-header {  
  width: 100%;  
  padding: 0 20px;  
  z-index: 99;  
  background-color: #212121;  
  color: white;  
  border-bottom: 1px solid;  
  border-color: rgba(0, 0, 0, 0.1);  
  @include md {  
    padding: 0 40px;  
  }  
  .header-content {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    height: 50px;  
    max-width: 1200px;  
    margin: 0 auto;  
    @include md {  
      height: 80px;  
    }  
  }  
  .left {  
    list-style-type: none;  
    display: none;  
    gap: 25px;  
    @include md {  
      display: flex;  
    }  
  }  
}
```

```

    li {
        font-size: 14px;
        font-weight: 600;
        text-transform: uppercase;
        cursor: pointer;
        &:hover {
            opacity: 0.6;
        }
    }
}

.center {
    font-size: 22px;
    font-weight: 700;
    cursor: pointer;
    @include md {
        font-size: 34px;
        position: absolute;
        left: 50%;
        transform: translateX(-50%);
    }
}

.right {
    display: flex;
    align-items: center;
    gap: 20px;
    @include md {
        gap: 25px;
    }
    svg {
        font-size: 20px;
        cursor: pointer;
        @include md {

```



```

        font-size: 24px;
    }
    &:hover {
        opacity: 0.6;
    }
    .cart-icon {
        position: relative;
        span {
            min-width: 20px;
            text-align: center;
            background: #8e413f;
            padding: 2.5px;
            position: absolute;
            top: -5px;
            right: -12px;
            font-size: 12px;
            line-height: 1;
            border-radius: 10px;}}}}
    &.sticky-header {
        background-color: #212121;
        color: white;
        position: sticky;
        top: 0;
        transform: translateY(-60px);
        animation: StickyHeader 0.3s ease forwards;}}

@keyframes StickyHeader {
    0% {
        transform: translateY(-60px);
    }
    100% {
        transform: translateY(0);}}

```

Search

```
import React, { useState } from "react";
import { MdClose } from "react-icons/md";
import "./Search.scss";
import useFetch from "../../hooks/useFetch";
import { useNavigate } from "react-router-dom";

const Search = ({ setSearchModal }) => {
  const [query, setQuery] = useState("");
  const navigate = useNavigate();

  const onChange = (e) => {
    setQuery(e.target.value);
  };

  let { data } = useFetch(
    `/api/products?populate=*&filters[title][$contains]=${query}`
  );

  if (!query.length) {
    data = null;
  }

  return (
    <div className="search-modal">
      <div className="form-field">
        <input
          autoFocus
          type="text"
          placeholder="Search for products"
          value={query}
          onChange={onChange}
        />
      </div>
    </div>
  );
};
```

```

        />
        <MdClose
            className="close-btn"
            onClick={() => setSearchModal(false)}
        />
    </div>
    <div className="search-result-content">
        {!data?.data?.length && (
            <div className="start-msg">
                Start typing to see products you are looking
for.
            </div>
        )}
        <div className="search-results">
            {data?.data?.map((item) => (
                <div
                    className="search-result-item"
                    key={item.id}
                    onClick={() => {
                        navigate("/product/" + item.id);
                        setSearchModal(false);
                    }}
                >
                    <div className="img-container">
                        <img src={
+
                            process.env.REACT_APP_DEV_URL
+
                            item.attributes.image.data[0].attributes.url} alt=""
                        />
                    </div>
                    <div className="prod-details">
                        <span className="name">

```

```
        {item.attributes.title}
      </span>
      <span className="desc">
        {item.attributes.description}
      </span>
    </div>
  </div>
  )})
</div>
</div>
</div>
);
};

export default Search;
```

Search SCSS

```
@import "../../../css-config/mixins.scss";

.search-modal{
  position: fixed;
  width: 100%;
  height: 100%;
  z-index: 999;
  top: 0;
  left: 0;
  background-color: white;
  transform: translateY(100%);
  animation: slideSearchWindow 0.5s ease forwards;
  .form-field{
    width: 100%;
    display: flex;
    justify-content: center;
    padding: 10px 50px;
    border-bottom: 1px solid rgba(0, 0, 0, 0.1);
    position: relative;
    @include md{
      padding: 20px 0;
    }
    input{
      width: 100%;
      max-width: 1200px;
      height: 50px;
      text-align: center;
      font-size: 20px;
      font-weight: 600;
      color: #212121;
      outline: none;
      border: none;
    }
  }
}
```

```

        text-transform: uppercase;
        @include md {
            font-size: 48px;
            height: 80px;
        }
    }

    svg{
        position: absolute;
        font-size: 25px;
        right: 20px;
        top: 50%;
        transform: translateY(-50%);
        cursor: pointer;
        @include md{
            right: 40px;
            font-size: 50px;
        }
    }
}

.search-result-content{
    max-width: calc(100% - 20px);
    margin: 0 auto;
    @include md {
        max-width: 800px;
    }
    .start-msg {
        text-align: center;
        margin-top: 20px;
        color: rgba(0, 0, 0, 0.5);
    }
}

```

```

}
.search-results{
    height: calc(100vh - 110px);
    overflow: auto;
    margin: 20px 0;
    @include md {
        height: calc(100vh - 160px);
    }
    .search-result-item{
        display: flex;
        align-items: center;
        gap: 10px;
        padding: 10px 0;
        border-bottom: 1px solid rgba(0, 0, 0, 0.1);
        cursor: pointer;
        .img-container{
            background-color: rgba(0, 0, 0, 0.05);
            width: 60px;
            height: 60px;
            flex-shrink: 0;
            img{
                width: 100%;
                height: 100%;
            }
        }
    }
    .prod-details{
        overflow: hidden;

        .name{
            text-overflow: ellipsis;
            white-space: nowrap;
            overflow: hidden;

```

}

Home

```
import React, { useEffect, useContext } from "react";
import "./Home.scss";
import Banner from "./Banner/Banner";
import Category from "./Category/Category";
import Products from "../Products/Products";
import { fetchDataFromApi } from "../../utils/api";
import { Context } from "../../utils/context";

const Home = () => {
  const { products, setProducts, categories, setCategories } =
    useContext(Context);
  useEffect(() => {
    getProducts();
    getCategories();
  }, []);
  const getProducts = () => {
    fetchDataFromApi("/api/products?populate=*").then((res) => {
      console.log(res);
      setProducts(res);
    });
  };
  const getCategories = () => {
    fetchDataFromApi("/api/categories?populate=*").then((res) => {
      console.log(res);
      setCategories(res);
    });
  };
  return (
    <div>
      <Banner />
      <div className="main-content">
```

```
    <div className="layout">
      <Category categories={categories} />
      <Products
        headingText="Popular Products"
        products={products}
      /> </div></div></div>);};
export default Home;
```

Home SCSS

```
@import "../css-config/mixins.scss";
.main-content{
  max-width: calc(100% - 20px);
  margin: 0 auto;
  @include md {
    max-width: 1200px;
  }
}
```

Banner

```
import "./Banner.scss";
import BannerImg from "../../assets/banner-img.png";

const Banner = () => {
  return (
    <div className="hero-banner">
      <div className="content">

        <div className="text-content">
          <h1>Crystal Sound</h1>
          <p>Welcome to the world of bass-boasted audio
experience with our top-rated Bluetooth headphones.</p>
          <div className="ctas">
            <div className="banner-cta">Explore</div>
            <div className="banner-cta v2">Shop Now</div>
          </div>
        </div>

        <img className="banner-img" src={BannerImg} alt=""/>
      </div>
    </div>
  );
};

export default Banner;
```

Banner SCSS

```
@import "../../../css-config/mixins.scss";

.hero-banner {
  padding: 40px 0;
  position: relative;
  background-image: linear-gradient(to right, #964140, #421818);
  @include md {
    height: calc(100vh - 80px);
  }
  .content{
    height: 100%;
    display: flex;
    justify-content: flex-end;
    align-items: center;
    flex-direction: column-reverse;
    max-width: calc(100% - 20px);
    margin: 0 auto;
    position: relative;
    @include md {
      flex-direction: row;
      max-width: 1200px;
    }
  }
  .banner-img {
    position: relative;
    z-index: 9;
    width: 200px;
    margin-bottom: 20px;
    @include md {
      width: 600px;
      margin-right: 60px;
      margin-top: 50px;
      margin-bottom: 0;
    }
  }
}
```

```

    }
    @include xxl {
        width: 600px;
        margin-right: -100px;
        margin-top: 0;
    }
}

.banner-img {
    animation: moving 1s ease-in-out infinite alternate;
}

@keyframes moving {
    0% {
        transform: translateY(-10px);
    }
    100% {
        transform: translateY(10px);
    }
}

.text-content{
    color: white;
    font-weight: 200;
    text-align: center;
    display: flex;
    flex-direction: column;
    align-items: center;
    @include md{
        position: absolute;
        left: 50px;
        top: 50%;
        transform: translateY(-50%);
    }
}

```

```

}
@include xxl{
    left: 0;
}
h1 {
    font-size: 80px;
    font-weight: 700;
    line-height: 1;
    color: white;
    margin-bottom: 20px;
    @include md {
        font-size: 180px;
    }
}
p{
    max-width: 300px;
    font-size: 14px;
    line-height: 20px;
    margin-bottom: 40px;
    @include md{
        max-width: 500px;
        font-size: 18px;
        line-height: 24px;
        margin-bottom: 40px;
    }
}
.ctas{
    display: flex;
    justify-content: center;
    gap: 20px;
    .banner-cta {
        text-transform: uppercase;
    }
}

```

```
        font-size: 16px;
        font-weight: 500;
        width: fit-content;
        border: 2px solid white;
        border-radius: 80px;
        padding: 10px 20px;
        cursor: pointer;
        &.v2{
            background-color: white;
            color: black;
        }
        &:hover {
            opacity: 0.6;
        }
    }
}
}
```

Home / Category

```
import { useNavigate } from "react-router-dom";
import "./Category.scss";

const Category = ({ categories }) => {
  const navigate = useNavigate();
  return (
    <div className="shop-by-category">
      <div className="categories">
        {categories?.data?.map((item) => (
          <div
            key={item.id}
            className="category"
            onClick={() =>
              navigate(`/category/${item.id}`)}
          >
            <img src={
              process.env.REACT_APP_DEV_URL +
              item.attributes.img.data.attributes.url}
              alt=""/>
          </div>
        ))}
      </div>
    </div>
  );
};

export default Category;
```


Home/Category SCSS

```
@import "../../../css-config/mixins.scss";

.shop-by-category{
  margin: 25px 0;
  @include md {
    margin: 50px 0;
  }
  .categories{
    display: flex;
    flex-flow: wrap;
    gap: 10px;
    .category{
      background-color: black;
      width: calc(50% - 5px);
      cursor: pointer;
      overflow: hidden;
      @include md {
        width: calc(25% - 10px);
      }
      img {
        width: 100%;
        display: block;
        transition: all ease 0.3s;
      }
      &:hover {
        img{
          transform: scale(1.2);
        }
      }
    }
  }
}
```

Products

```
import "./Products.scss";
import Product from "./Product/Product";

const Products = ({ products, innerPage, headingText }) => {
  return (
    <div className="products-container">
      {!innerPage && <div className="sec-
heading">{headingText}</div>}
      <div className={`products ${innerPage ? "innerPage" :
""}`}>
        {products?.data?.map((item) => (
          <Product
            key={item.id}
            id={item.id}
            data={item.attributes}
          />
        ))}
      </div>
    </div>
  );
};

export default Products;
```

Products SCSS

```
@import "../css-config/mixins.scss";

.products-container{
  margin: 50px 0;
  @include md {
    margin: 75px 0;}
  .sec-heading{
    margin-bottom: 20px;
    font-size: 18px;
    font-weight: 500;
    text-transform: uppercase;
    @include md {
      margin-bottom: 35px;
      font-size: 24px;}
    &:after{
      content: "";
      display: block;
      margin-top: 5px;
      width: 50px;
      height: 3px;
      background-color: #8e413f;
      @include md{
        margin-top: 10px;}}}}
  .products{
    display: flex;
    flex-flow: wrap;
    gap: 10px;
    @include md {
      gap: 20px;
    }
  }
}
```

Product

```
import React from "react";
import { useNavigate } from "react-router-dom";
import "../Product.scss";
const Product = ({ data, id }) => {
  const navigate = useNavigate();
  return (
    <div
      className="product-card"
      onClick={() => navigate("/product/" + id)}
    >
      <div className="thumbnail">
        <img
          src={
            process.env.REACT_APP_DEV_URL +
            data.image.data[0].attributes.url
          } alt=""
        />
      </div>
      <div className="prod-details">
        <span className="name">{data.title}</span>
        <span className="price">₹{data.price}</span>
      </div>
    </div>
  );
};

export default Product;
```

Product SCSS

```
@import "../../../css-config/mixins.scss";

.product-card{
  width: calc(50% - 5px);
  margin-bottom: 20px;
  @include md{
    width: calc(25% - 15px);
  }
  .thumbnail{
    width: 100%;
    height: 180px;
    background-color: rgba(0, 0, 0, 0.05);
    margin-bottom: 10px;
    padding: 25px;
    display: flex;
    align-items: center;
    @include md {
      height: 350px;
    }
    img{
      transition: all ease 0.3s;
      display: block;
      width: 100%;
    }
  }
  .prod-details{
    .name{
      font-size: 14px;
      display: block;
      text-overflow: ellipsis;
      white-space: nowrap;
      overflow: hidden;
      @include md {
```

```
        font-size: 16px;
        margin-bottom: 10px;
    }}
    .price{
        font-size: 18px;
        color: #8e413f;
        @include md {
            font-size: 24px;
        }}
    &:hover{
        .thumbnail{
            img{
                transform: scale(1.2);}}}}}
```

Single Product

```
import { useContext, useState } from "react";
import { Context } from "../../utils/context";
import { useParams } from "react-router-dom";
import useFetch from "../../hooks/useFetch";
import RelatedProducts from "../RelatedProducts/RelatedProducts";
import {
  FaFacebookF,
  FaTwitter,
  FaInstagram,
  FaLinkedinIn,
  FaPinterest,
  FaCartPlus,
} from "react-icons/fa";
import "../SingleProduct.scss";
const SingleProduct = () => {
  const [quantity, setQuantity] = useState(1);
  const { id } = useParams();
  const { handleAddToCart } = useContext(Context);
  const { data } =
useFetch(`/api/products?populate=*[filters][id]=${id}`);
  const decrement = () => {
    setQuantity((prevState) => {
      if (prevState === 1) return 1;
      return prevState - 1; });});
  const increment = () => {
    setQuantity((prevState) => prevState + 1);
  };
  if (!data) return;
  const product = data?.data?.[0]?.attributes;

  return (
```

```

<div className="single-product-main-content">
  <div className="layout">
    <div className="single-product-page">
      <div className="left">
        <img
          src={
            process.env.REACT_APP_DEV_URL +
            product.image.data[0].attributes.url
          } alt=""
        />
      </div>
      <div className="right">
        <span className="name">{product.title}</span>
        <span
className="price">&#8377; {product.price}</span>
        <span
className="desc">{product.description}</span>

        <div className="cart-buttons">
          <div className="quantity-buttons">
            <span onClick={decrement}>-</span>
            <span>{quantity}</span>
            <span onClick={increment}>+</span>
          </div>
          <button
            className="add-to-cart-button"
            onClick={() => {
              handleAddToCart(data?.data?.[0],
quantity);
              setQuantity(1);
            }}
          >
            <FaCartPlus size={20} />

```



```

        ADD TO CART
      </button>
    </div>

    <span className="divider" />
    <div className="info-item">
      <span className="text-bold">
        Category:{" "}
        <span>
          {
product.categories.data[0].attributes
                                .title
                                }
          </span>
        </span>
      <span className="text-bold">
        Share:
        <span className="social-icons">
          <FaFacebookF size={16} />
          <FaTwitter size={16} />
          <FaInstagram size={16} />
          <FaLinkedinIn size={16} />
          <FaPinterest size={16} />
        </span></span>
      </div></div></div>
    <RelatedProducts
      productId={id}
      categoryId={product.categories.data[0].id}/>
    </div></div>);};
export default SingleProduct;

```

Single Product SCSS

```
@import "../css-config/mixins.scss";

.single-product-main-content{
  margin: 20px 0;
  @include md {
    margin: 75px 0;
  }
  .layout{
    max-width: calc(100% - 20px);
    margin: 0 auto;
    @include md {
      max-width: 1200px;
    }
    .single-product-page{
      display: flex;
      flex-direction: column;
      @include md {
        flex-direction: row;
      }
      .left{
        width: 100%;
        background-color: rgba(0, 0, 0, 0.05);
        flex-shrink: 0;
        @include md{
          width: 400px;
          height: 400px;
        }
        img{
          width: 100%;
          display: block;
        }
      }
    }
  }
}
```

```

.right {
  display: flex;
  flex-direction: column;
  padding-top: 20px;
  @include md {
    padding: 0 35px;
  }
  .name{
    font-size: 20px;
    line-height: 28px;
    margin-bottom: 20px;
    @include md {
      font-size: 24px;
      line-height: 32px
    }
  }
  .price{
    font-size: 24px;
    line-height: 32px;
    margin-bottom: 20px;
    color: #8e413f;

  }
  .desc {
    font-size: 14px;
    line-height: 20px;
    margin-bottom: 20px;
    color: #6b6b6b;
    @include md {
      font-size: 16px;
      line-height: 24px;
      margin-bottom: 20px;
    }
  }
}

```

```

    }
}
.cart-buttons{
  display: flex;
  @include md {
    margin-top: 30px;
  }
.quantity-buttons{
  width: fit-content;
  display: flex;
  border: 2px solid rgba(0, 0, 0, 0.2);
  margin-right: 10px;
  height: 50px;
  span{
    font-size: 18px;
    width: 40px;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    color: #6b6b6b;

    &:nth-child(1){
      border-right: 2px solid rgba(0, 0, 0,
0.2);
    }
    &:nth-child(2){
      width: 60px;
    }
    &:nth-child(3){
      border-left: 2px solid rgba(0, 0, 0,
0.2);
    }
  }
}

```

```

    }
}
.add-to-cart-button{
    outline: 0;
    border: 0;
    height: 50px;
    width: 180px;
    display: flex;
    align-items: center;
    justify-content: center;
    cursor: pointer;
    font-size: 16px;
    color: white;
    background-color: #8e413f;
    border-bottom: 3px solid #662e2c;
    flex-grow: 1;
    @include md {
        flex-grow: unset;
    }
    svg {
        margin-right: 10px;
    }
}
}
.divider{
    margin: 20px 0;
    height: 1px;
    width: 100%;
    background-color: rgba(0, 0, 0, 0.1);
}
.info-item{
    .text-bold{

```

```
        font-size: 18px;
        font-weight: 500;
        display: block;
        &:nth-child(1){
            margin-bottom: 20px
        }
        span{
            font-size: 16px;
            font-weight: 400;
            cursor: pointer;
            color: #6b6b6b;

        }
        svg{
            margin: 0 5px;
        }
    }}
}}
}
```

Related Product

```
import React from "react";
import useFetch from "../../hooks/useFetch";
import Products from "../../Products/Products";
const RelatedProducts = ({ categoryId, productId }) => {
  const { data } = useFetch(
    `/api/products?populate=*&filters[id][$ne]=${productId}&filters[categories][id]=${categoryId}&pagination[start]=0&pagination[limit]=4`
  );

  return (
    <div className="related-products">
      <Products headingText="Related Products" products={data} />
    </div>
  );
};
export default RelatedProducts;
```

App

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Header from "../components/Header/Header";
import Footer from "../components/Footer/Footer";
import Home from "../components/Home/Home";
import Category from "../components/Category/Category";
import SingleProduct from "../components/SingleProduct/SingleProduct";
import Newsletter from "../components/Footer/Newsletter/Newsletter";
import AppContext from "../utils/context";

function App() {
  return (
    <BrowserRouter>
    <AppContext>

    <Header />
    <Routes>
      <Route path="/" element={<Home/>}/>
      <Route path="/category/:id" element={<Category/>}/>
      <Route path="/product/:id" element={<SingleProduct/>}/>
    </Routes>
    <Newsletter />
    <Footer />
  </AppContext>
</BrowserRouter>
);
}

export default App;
```


index

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.scss";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

index SCSS

```
@import
url("https://fonts.googleapis.com/css2?family=Cabin:wght@400;500;600;700&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Cabin", sans-serif;
}

body {
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  font-size: 16px;
  line-height: 22px;
  font-weight: 500;
}
```

Order

```
("use strict");
const stripe = require("stripe")(process.env.STRIPE_KEY);
/**
 * order controller
 */
const { createCoreController } = require("@strapi/strapi").factories;
module.exports = createCoreController("api::order.order", ({ strapi })
=> ({
  async create(ctx) {
    const { products } = ctx.request.body;
    try {
      const lineItems = await Promise.all(
        products.map(async (product) => {
          const item = await strapi
            .service("api::product.product")
            .findOne(product.id);

          return {
            price_data: {
              currency: "inr",
              product_data: {
                name: item.title,
              },
              unit_amount: Math.round(item.price * 100),
            },
            quantity: product.attributes.quantity,
          };
        })
      );
    }

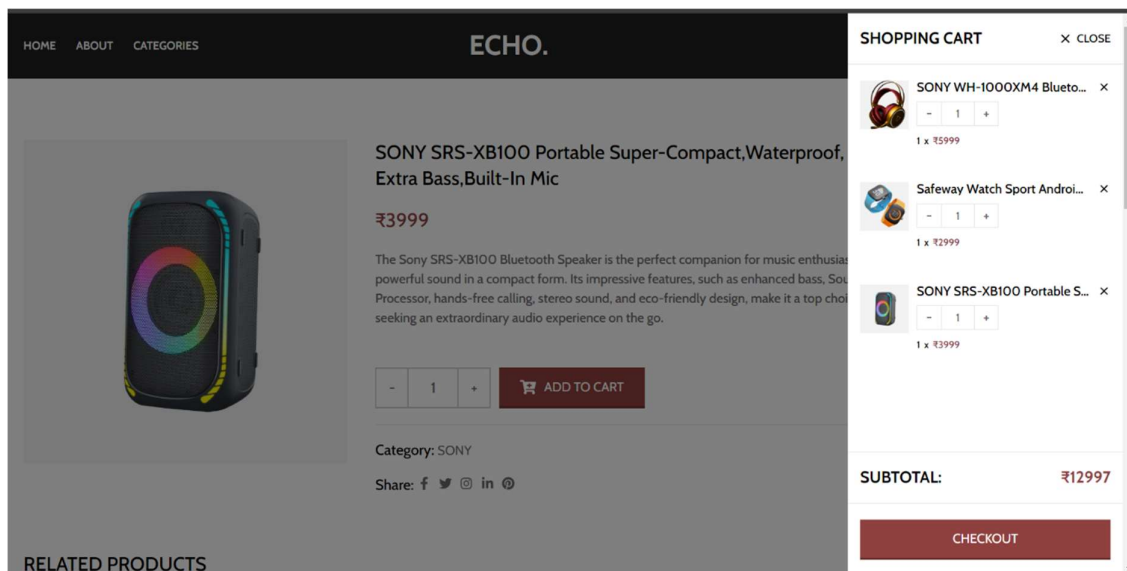
    const session = await stripe.checkout.sessions.create({
```

```

        shipping_address_collection: { allowed_countries: ["IN"] },
        payment_method_types: ["card"],
        mode: "payment",
        success_url: process.env.CLIENT_URL + "/success",
        cancel_url: process.env.CLIENT_URL + "?success=false",
        line_items: lineItems,
    });
    await strapi
        .service("api::order.order")
        .create({ data: { products, stripeId: session.id } });
    return { stripeSession: session };
} catch (error) {
    ctx.response.status = 500;
    return { error };
},
}
)
);

```

Screenshot



About

Lorem ipsum dolor sit amet consectetur adipiscing elit. Harum quisquam vel possimus quaerat, facere repellat commodi iusto unde eum cupiditate.

Contact

Maharana Pratap Nagar, Bhopal, Madhya Pradesh (462011)
Phone : +91 9876543210
Email : web@echo.com

Categories

Headphones
Smart Watches
Bluetooth Speakers
Wireless Earbuds
Earphones
Home Theatre

Pages

Home
About
Privacy Policy
Returns
Terms & Conditions
Contact Us

ECHO 2024 E-Commerce Website for Music Gadgets.



NEWSLETTER

SIGN UP FOR LATEST UPDATES AND OFFERS

Email Address

Subscribe

Will be used in accordance with our Privacy Policy

f

t

@

in

HOME ABOUT CATEGORIES

ECHO.



Crystal Sound

Welcome to the world of bass-boosted audio experience with our top-rated Bluetooth headphones.

EXPLORE

SHOP NOW



B



- boAt Rockerz 450 DC edition | Wireless Headphone with 40mm Dynamic Driver

There is no better audio to your playlist & songs than Rockerz 450 DC edition. It has battery & your energy. It...
- Rockerz 550 Over Ear Bluetooth Headphone

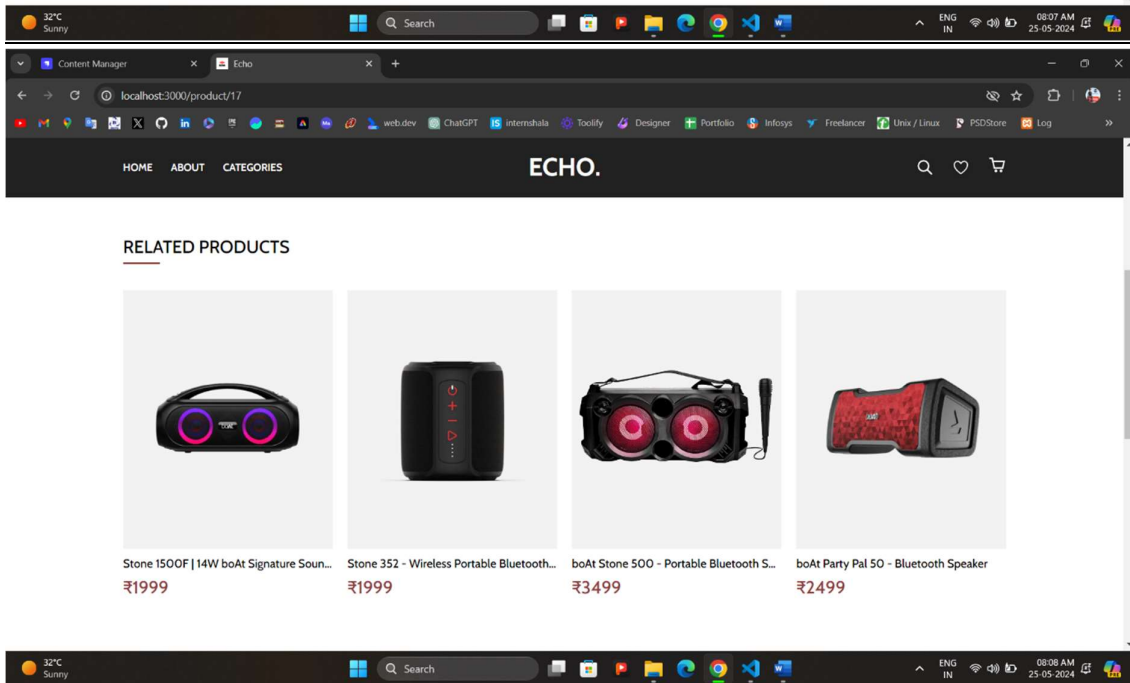
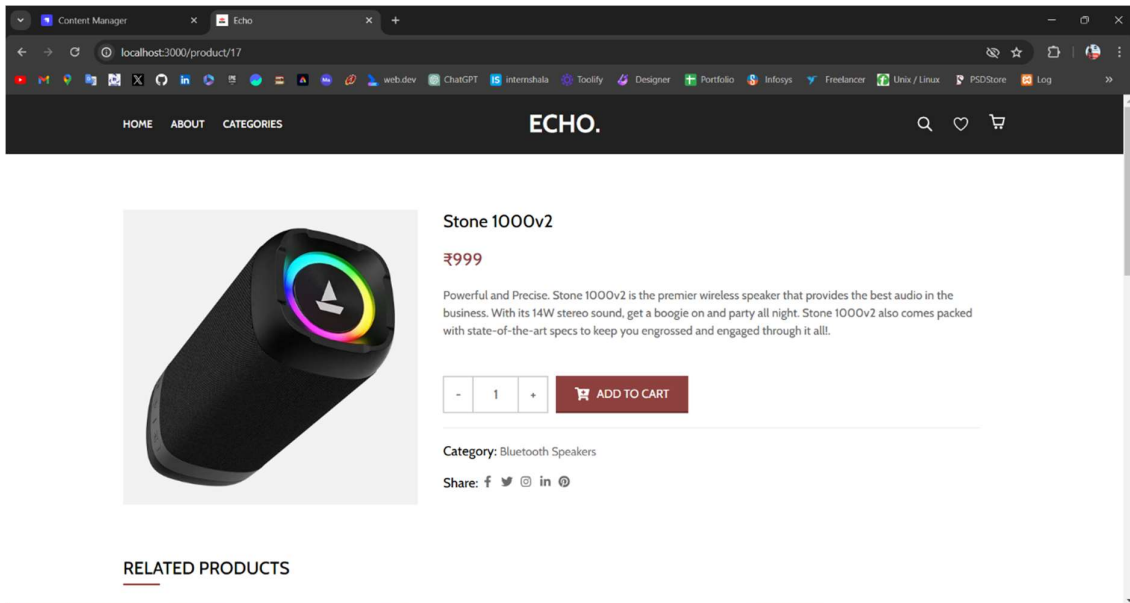
boAt Rockerz 550 is an over-ear wireless headset that has been ergonomically designed to meet the needs of music lo...
- boAt Rockerz 518 Wireless Headphone

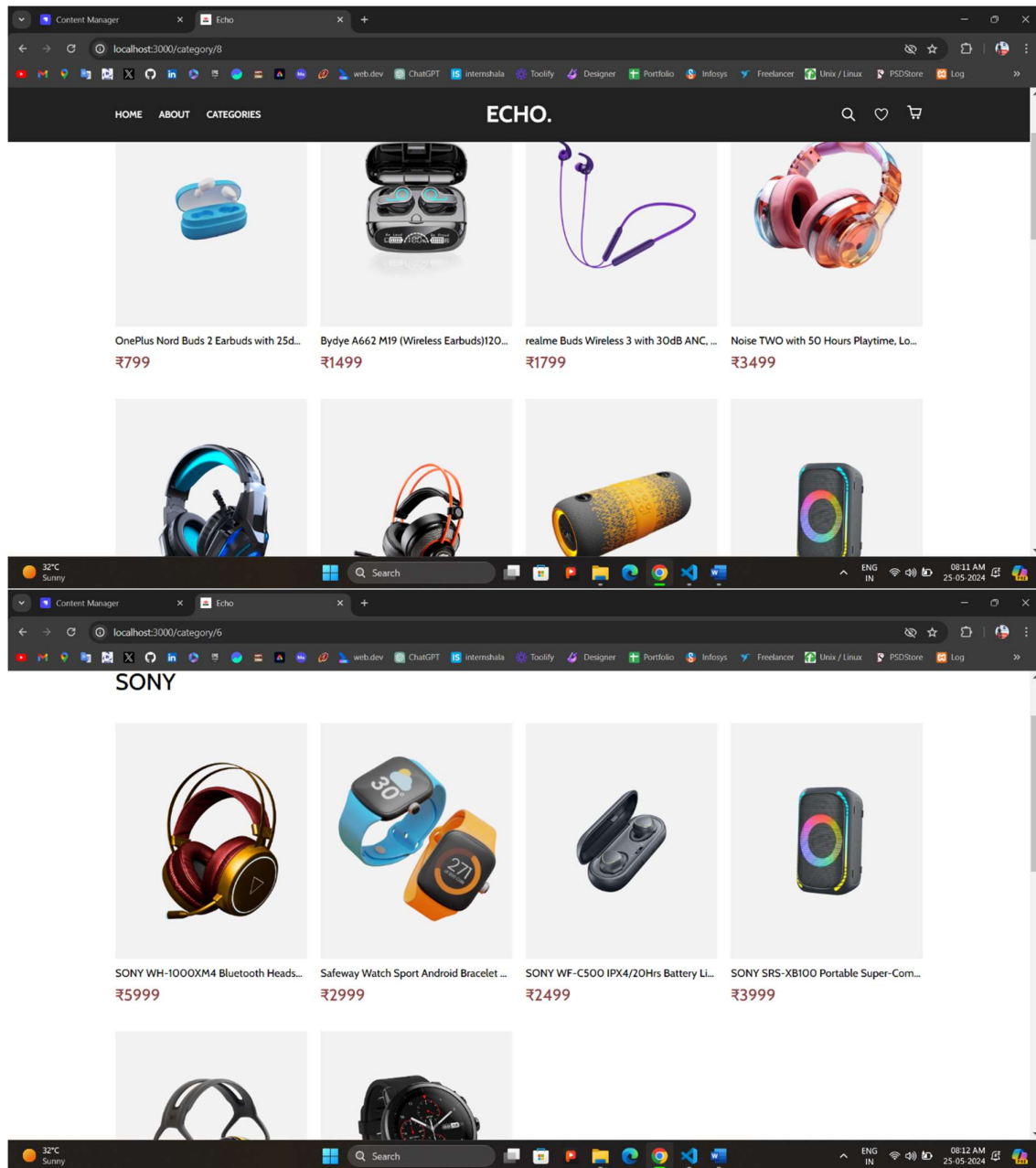
Now just stop searching about where heading from here and there. Pair with any Bluetooth device, connect and play. Do...
- Trebel Rockerz 550 Wireless Bluetooth Headphones

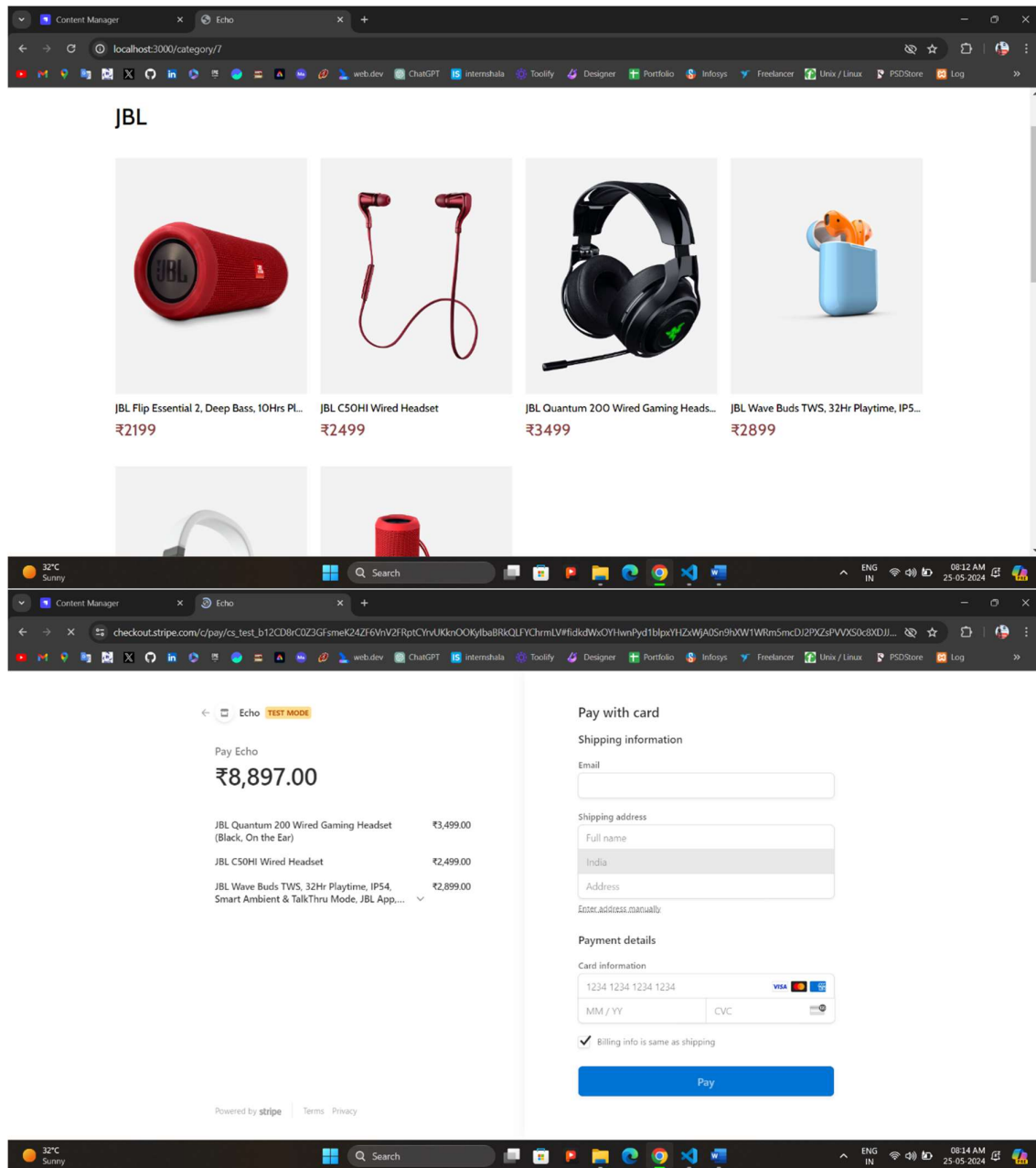
Let's, the time hath come to reach the soul of tone. Make your hustle shine like a diamond with Trebel Rockerz 550...
- Rockerz 450 Superman DC Edition - Wireless Bluetooth Headphone

There is no better audio to your playlist than Rockerz 450 DC edition. It has battery & your energy. It...
- boAt Wave Neo | 1.69" HD Display, Upto 7 Days Battery Life, IP68 Sweat

Introducing a iPhonelike to live smart and stylish—the all-new Watch Wave Neo. Check everything that you need to do...
- boAt Iris | Round Dial Smart Watch with 1.39" AMOLED Display, Multiple Watch Faces







Conclusion

Here are some general points that might be included in the conclusion of an e-commerce website:

Achievement of Goals: Summarize whether the website achieved its intended goals, such as increasing sales, expanding customer base, or improving brand visibility.

User Experience: Reflect on the overall user experience of the website, including ease of navigation, clarity of product information, and efficiency of the checkout process.

Technology and Performance: Evaluate the effectiveness of the website's technology stack, including its stability, speed, and responsiveness across different devices and browsers.

Marketing and Promotion: Discuss the success of marketing strategies implemented to drive traffic to the website, including SEO, social media campaigns, email marketing, and paid advertising.

Customer Feedback: Incorporate feedback from customers, including reviews, ratings, and comments, to assess satisfaction levels and identify areas for improvement.

Competitive Analysis: Analyse the website's performance in comparison to competitors, considering factors such as pricing, product selection, customer service, and overall market share.

Future Plans: Outline future strategies and initiatives to further enhance the website's performance and competitiveness, such as expanding product offerings, improving website functionality, or entering new markets.

Financial Performance: Provide an overview of the website's financial performance, including revenue generated, profit margins, and return on investment.

Gratitude and Acknowledgments: Express gratitude to all stakeholders involved in the development and success of the website, including team members, partners, suppliers, and customers.

Future Scope of the Project

Focus on Sustainability:

Eco-friendly packaging: Use recycled and biodegradable materials for packaging.

Sustainable materials in headphones: Offer headphones made from recycled plastics or other sustainable materials.

Partner with eco-conscious brands: Partner with brands that share your commitment to sustainability.

Subscription and Rental Models:

Headphone subscription boxes: Offer curated subscription boxes with different headphones each month, allowing users to try before they buy.

Headphone rental Service: Provide a rental service for high- end headphone for specific occasions or travel.

Customization:

Customization and Personalization: Future headphones may offer greater customization options, allowing users to tailor sound profiles, fit, and design elements to their individual preferences. This could involve the use of companion apps and software updates to optimize the listening experience based on user feedback and usage patterns.