

A PROJECT REPORT

On

BHAVATU

Submitted By

Mr. Vishal Navnath Aher

in partial fulfillment for the award of the degree of

BACHELOR OF SCIENCE

In

COMPUTER SCIENCE

under the guidance of

Prof. Sampurna Sarode

Department of Computer Science



VPM'S R Z SHAH COLLEGE OF ARTS,

SCIENCE AND COMMERCE

MITHAGAR ROAD, MULUND (E), MUMBAI -400 081

(NAAC Accredited "A" Grade)

(Sem V)

(2024-2025)



**VIDYA PRASARAK MANDAL'S
R Z SHAH COLLEGE OF ARTS, SCIENCE AND COMMERCE
MITHAGAR ROAD, MULUND (E), MUMBAI-400 081
(NAAC Accredited "A" Grade)
DEPARTMENT OF
INFORMATION TECHNOLOGY & COMPUTER SCIENCE**

CERTIFICATE

This is to certify that the project entitled, **BHAVATU** is bonfied work of **VISHAL NAVNATH AHER** bearing Seat.No: _____ submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE** in **Computer Science** on behalf of University of Mumbai during the academic year 2024-25.

Signature of Project Guide

Co-ordinator

Signature of H.O.D./

Signature of External Examiner

DECLARATION

I, Vishal Navnath Aher, hereby declare that the project entitled “BHAVATU ” Submitted in the partial fulfillment for the award of Bachelor of Science in Computer Science during the academic year **2024-2025** in my original work and the project has not formed the basis for the award of my degree, associateship, fellowship or any other similar titles.

Signature of Student:

Place:

Date:

Bhavatu

An AI-Driven Vastu and Astrology Marketplace



0%

Plagiarism

100%

Unique

Make it Unique

Start New Search



Check Grammar



AI Content Detector

To check plagiarism in photos click here

Reverse Image Search

INDEX

1. Introduction	pg: 1-8
1.1 Objective	
1.2 Description of the Current System	
1.3 Limitation of the Current System	
1.4 Features	
2. Requirement Specification	pg: 9-12
2.1 Software Requirements	
2.2 Hardware Requirements	
3. System Design	pg: 13-20
3.1 Class Diagram	
3.2 Use Case Diagram	
3.3 Data Flow Diagram	
3.4 Activity Diagram	
4. System Implementation	pg: 21-60
4.1 Frontend	
4.2 Backend	
4.3 Payment Gateway Integration	
4.4 AI Assistant Integration	
5. Results/Output	pg: 61-70
6. Future Enhancements & Conclusion	pg: 71-75
7. References	pg: 76

INTRODUCTION

1.1 Objective of the Project

Bhavatu is an integrated platform that combines ancient practices such as **Vastu** and **astrology** with modern-day technology, powered by **AI**. It provides users with tools to:

- Analyze **Vastu blueprints** using AI and generate recommendations for improvements in home and office layouts.
- Get **daily astrology updates**, which are tailored to the user's horoscope and planetary alignments.
- Purchase **Vastu-related products**, including gemstones and spiritual items, through a secure e-commerce system.
- Interact with an **AI assistant** (Reyna) for recommendations and queries about Vastu, astrology, and product purchases.

Key Objectives:

- **AI-driven Vastu Blueprint Analysis:** Empower users to upload their home or office blueprints, which are then analyzed using AI to provide suggestions for optimizing energy flow.
- **Astrology Feed:** Offer daily astrology updates, personalized horoscopes, and planetary transits.
- **E-commerce Platform:** Provide a user-friendly marketplace where users can purchase Vastu-related products like gemstones, pyramids, and yantras.
- **AI Chatbot (Reyna):** Serve as an assistant to answer queries and offer suggestions related to Vastu, astrology, and the e-commerce platform.

1.2 Description of the Current System

In the current system, users have to rely on:

- **Manual Vastu Consultation:** This process can be expensive, time-consuming, and difficult to access for users in remote areas.
- **Multiple Platforms:** Separate platforms are used for astrology, Vastu consultation, and e-commerce. These platforms do not provide a unified user experience, making it difficult for users to manage their needs in one place.
- **Lack of Personalized Assistance:** Most current platforms do not offer AI-driven insights, which means users rely on human consultants or generalized advice.

1.3 Limitations of the Current System

- **No Unified Platform:** Users must visit different websites for Vastu consultations, astrology updates, and purchasing items related to these fields.
- **Manual and Time-Consuming:** Traditional Vastu analysis requires human consultation, which can be slow and limited in availability.
- **No AI Assistance:** There are limited AI-driven platforms that provide real-time assistance for Vastu and astrology.
- **Lack of Integration:** Current systems do not integrate AI, astrology, and e-commerce on one platform, making it harder for users to receive tailored advice.

1.4 Features of Bhavatu

- **AI-Powered Vastu Blueprint Analysis:** Analyze uploaded blueprints of users' homes or offices and provide recommendations to optimize energy flow based on Vastu principles.
- **Daily Astrology Updates:** Personalized astrology updates, horoscopes, and transit reports are provided to users in real-time.
- **E-commerce Platform:** Bhavatu allows users to purchase Vastu-related items like gemstones, pyramids, and yantras. Secure payment is handled via **Razorpay**.
- **AI Assistant (Reyna):** Reyna helps users navigate the platform, answers Vastu and astrology-related questions, and makes product recommendations.
- **Mobile-First Design:** The platform is responsive and works smoothly on mobile devices, ensuring accessibility for all users.
- **Firebase Authentication:** Provides secure user authentication for the platform, using Google or email/password sign-ins.

Merits of Bhavatu

1. AI-Driven Insights:

- Provides accurate and personalized Vastu analysis based on user-uploaded blueprints, enhancing the effectiveness of traditional practices.

2. Comprehensive Platform:

- Combines Vastu consultation, daily astrology updates, and an e-commerce marketplace in one place, offering a seamless user experience.

3. User-Friendly Interface:

- Designed with an intuitive interface, making it accessible to users of all ages and technical backgrounds.

4. Daily Astrology Updates:

- Keeps users informed with personalized astrology updates, enhancing their connection to celestial influences.

5. Secure Payment Processing:

- Integration with Razorpay ensures secure transactions, building trust among users when purchasing Vastu-related items.

6. AI Assistant (Reyna):

- The chatbot provides immediate assistance, answering user queries and guiding them through the platform, thus improving user engagement.

7. Market Demand:

- Addresses the growing interest in Vastu and astrology, appealing to a diverse audience looking for holistic solutions for their living spaces.
- Available online, making it convenient for users to access services from anywhere without the need for in-person consultations.

Demerits of Bhavatu

1. Dependency on Internet Connectivity:

- Users without stable internet access may face difficulties in utilizing the platform effectively, especially for real-time updates and interactions.

2. Limited Target Audience:

- Primarily targets individuals interested in Vastu and astrology, which may restrict the appeal to broader user demographics.

3. Potential Information Overload:

- Users may feel overwhelmed with the amount of information provided, particularly in astrology updates and product choices.

4. AI Limitations:

- While AI provides insights, it may not fully replace the nuanced understanding of a qualified Vastu consultant, especially for complex cases.

5. Development and Maintenance Costs:

- Integrating multiple features like AI analysis, e-commerce, and real-time updates can lead to high initial development and ongoing maintenance costs.

6. User Trust and Data Privacy Concerns:

- Users may be hesitant to share personal data, especially related to Vastu and astrology, raising concerns about data security and privacy.

7. Learning Curve for New Users:

- While the platform aims to be user-friendly, some advanced features may require a learning curve, particularly for less tech-savvy users.

REQUIREMENT SPECIFICATION

2.1 Software Requirements

- **Frontend Technologies:**

- **React.js:** The main technology for the frontend of the website.
- **Vite:** Build tool used to optimize the performance of the application.
- **Axios:** To handle HTTP requests for API integration.
- **HTML, CSS, and JavaScript:** Core technologies for building the user interface.
- Visual studio code

- **Backend Technologies:**

- **Strapi:** Headless CMS used for managing products, categories, and orders in the e-commerce platform.
- **Firebase:** Used for user authentication and login mechanisms.

- **AI Libraries:**

- **OpenAI API:** Used for the chatbot, Reyna, which interacts with users and provides insights into Vastu and astrology.

- **Payment Integration:**

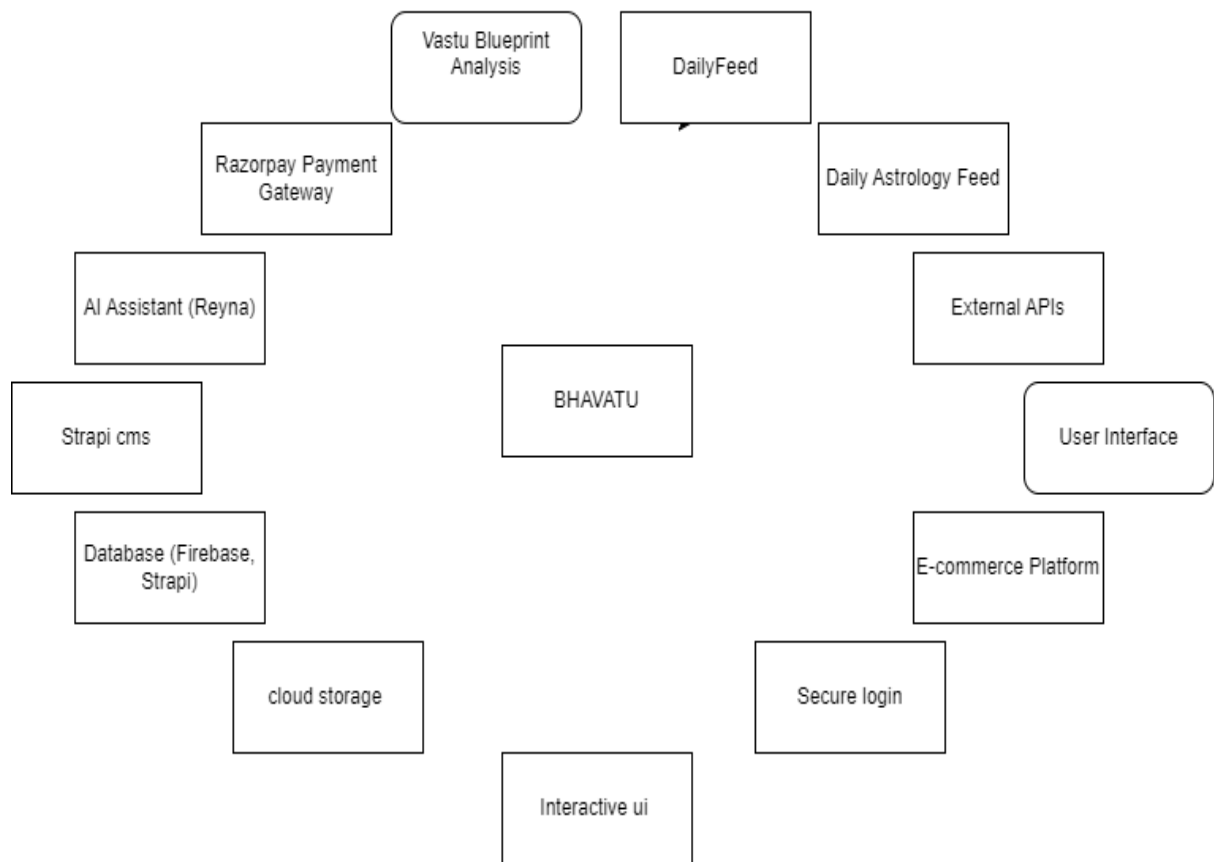
- **Razorpay:** Integrated for handling secure transactions during product purchases on the platform.

2.2 Hardware Requirements

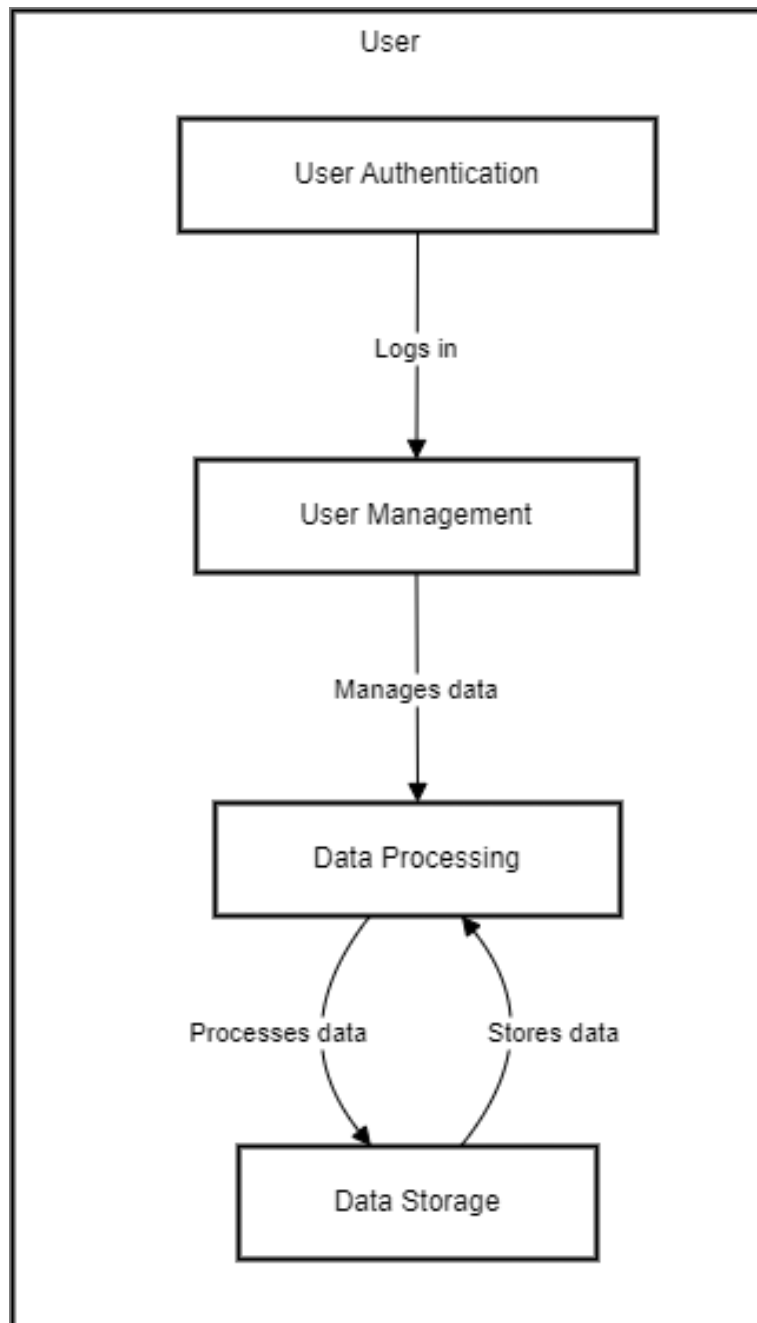
- **Processor:** Intel i5 or equivalent
- **RAM:** 8 GB minimum
- **Storage:** 512 GB SSD
- **Internet:** Stable connection for accessing Firebase, Strapi, and real-time astrology updates

SYSTEM DESIGN

Architecture Diagram



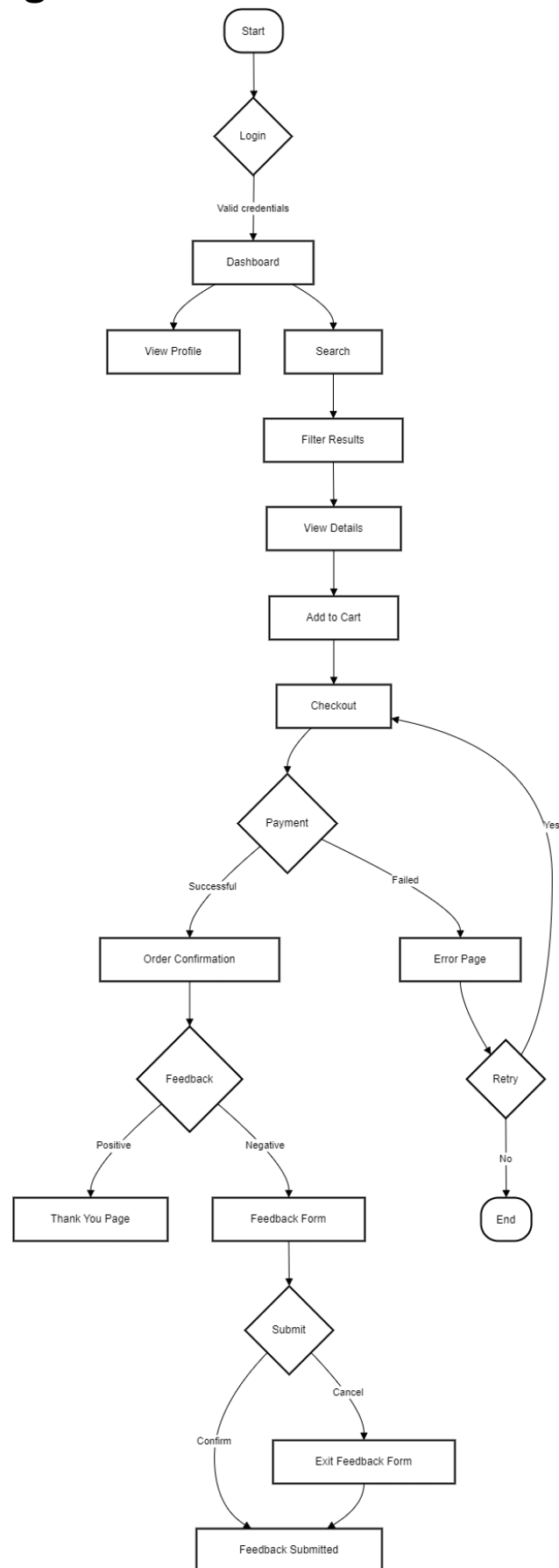
Data Flow Diagram for Bhavatu



Use Case Diagram



Activity Diagram



SYSTEM IMPLEMENTATION

4.1 Frontend Implementation

React Components

- **EcomHeader:** Displays navigation, search, and cart icons.
- **Products:** Fetches and displays a list of products.
- **Product:** Displays the details of a single product.
- **AstrologyFeed:** Shows daily astrology updates fetched from an external API.

State Management

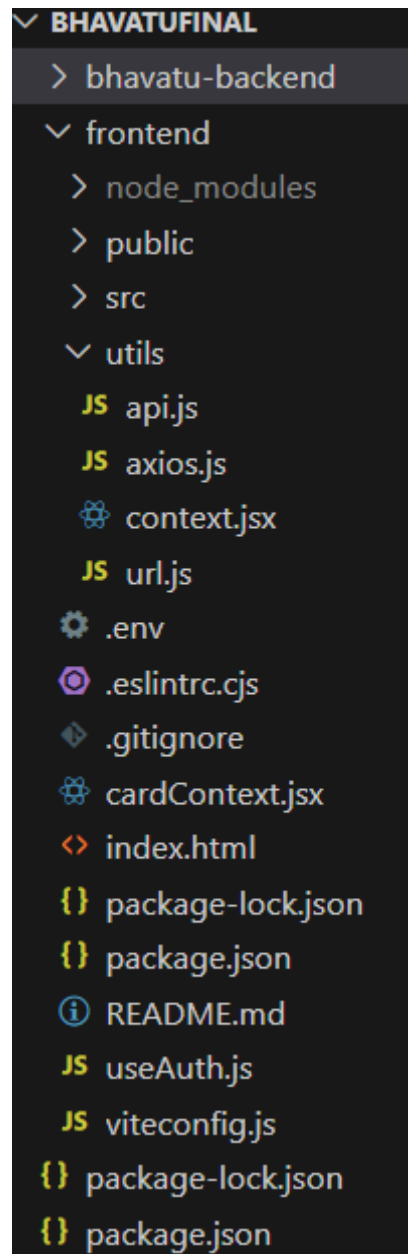
- **React Hooks:** Use `useState` and `useEffect` to manage the state for categories, products, cart, and daily feeds.
- **Context API:** To share state (such as products and cart) across multiple components.

CSS and Styling

- CSS is optimized for a dark theme, ensuring that fonts and images are clearly visible. Media queries are used to ensure that the website is fully responsive on all devices.

Source code :

File structure :



Home.jsx

```
import React from "react";

import "./Home.css";

import { Link } from "react-router-dom";

export default function Home() {

  return (

    <>

      <section className="promo-section">

        <div className="overlay"></div>

        <div className="content">

          <h1>

            Bhavatu leverages advanced AI algorithms

            <span className="highlight">

              to provide you with comprehensive statistical analysis

            </span>

            Trial <br />

            detailed reports about the Vastu of your home

          </h1>

          <p className="para">

            By assessing various parameters and aligning them with Vastu

            principles, our app helps you understand the current state of your

            living environment and offers actionable insights for improvement..

          </p>

          <div className="buttons">

            <Link to="/Vastu" className="btn">

              Get Started

            </Link>

            <Link to="/DailyFeed" className="play-btn">

              <div className="play-icon">▶</div>

            </Link>

          </div>

        </div>

      </section>

      <div className="wrapper">

        <div className="grid-cards">
```

```
<div className="card-container">

  <div className="gradient-wrapper green-gradient">

    <h3 className="text-title">VASTU ANALYSIS</h3>

    <p className="text-description">

      AN AI BASED VASTU ANALYSIS POWERD CHATBOT

    </p>

    <Link to="/vastu" className="read-more-btn">Read More</Link>

  </div>

</div>
```

```
<div className="card-container">

  <div className="gradient-wrapper white-gradient">

    <h3 className="text-title">DAILYFeed</h3>

    <p className="text-description">

      Daily horoscope and quotes to your zodiac sign in just a click

    </p>

    <Link to="/dailyfeed" className="read-more-btn">Read More</Link>

  </div>

</div>
```

```
<div className="card-container">

  <div className="gradient-wrapper yellow-gradient">

    <h3 className="text-title">Our Products</h3>

    <p className="text-description">

      Our Astrology products and jewel that promises wealth and health along with mindful peace and harmony

    </p>

    <Link to="/products" className="read-more-btn">Read More</Link>

  </div>

</div>
```

```
<div className="card-container">

  <div className="gradient-wrapper blue-gradient">

    <h3 className="text-title">KUNDLI</h3>
```

<p className="text-description">

Save your Kundli here and get a personal space

</p>

<Link to="/kundli" className="read-more-btn">Read More</Link>

</div>

</div>

<div className="card-container">

<div className="gradient-wrapper blue-gradient">

<h3 className="text-title">KUNDLI MatchMAking</h3>

<p className="text-description">

Let's try to find you a compatible partner with a great study of Kundli and history of knowledge

</p>

<Link to="/kundli-matchmaking" className="read-more-btn">Read More</Link>

</div>

</div>

<div className="card-container">

<div className="gradient-wrapper blue-gradient">

<h3 className="text-title">AI Assistant</h3>

<p className="text-description">

Reyna, an AI assistant that guides you with your personal questions and queries

</p>

<Link to="/Assistant" className="read-more-btn">Read More</Link>

</div>

</div>

<div className="card-container">

<div className="gradient-wrapper blue-gradient">

<h3 className="text-title">login</h3>

<p className="text-description">

Login for further future part of BHAvatu family

</p>

<Link to="/Login" className="read-more-btn">Read More</Link>

```

        </div>

    </div>

    </div>

    </div>

    </>

    );

}

```

App.jsx

```

import React, { useState } from "react";

import { Route, Routes } from "react-router-dom";

import "./App.css";

import Navbar from "./Navbar/Navbar";

import Home from "./Home/Home";

import Vastu from "./Vastu/VastuPage";

import Kundli from "../src/kundli/kundli"; // Fixed casing

import DailyFeed from "./DailyFeed/dailyfeed"; // Fixed casing

import Login from "./Login/login"; // Fixed casing

import Singleproduct from "./SingleProducts/singleproduct";

// import Checkout from "./Products/Checkout";

import Assistant from "./REyna/reyna"; // Fixed casing

import Cart from "./cart/Cart"; // Fixed casing

import ChatbotButton from "./REyna/reyna"; // Fixed import to match component name

import Footer from "./Footer"; // Fixed casing

import Prod from "./prod/prod"

import RelatedProducts from "./RelatedProducts/RelatedProducts";

import Search from "./search/search";

import Category from "./Category/category";

import IndiviCategory from "./individualcategory/indivcategory";

function App() {

    const [cart, setCart] = useState([]);

    return (

        <div className="App">

```

```
<ChatbotButton />
```

```
<Navbar />
```

```
<Routes>
```

```
  <Route path="/" element={<Home />} />
```

```
  <Route path="/vastu" element={<Vastu />} />
```

```
  <Route path="/kundli" element={<Kundli />} />
```

```
  { /* <Route path="/" element={<CategoriesPage />} />
```

```
  <Route path="/category/:id" element={<IndiviCategory />} /> */}
```

```
  <Route path="/login" element={<Login />} />
```

```
  <Route path="/dailyfeed" element={<DailyFeed />} />
```

```
  <Route path="/assistant" element={<Assistant />} />
```

```
  <Route path="/categories" element={<Category />} />
```

```
  <Route path="/category/:id" element={<IndiviCategory />} />
```

```
  <Route path="/prod" element={<Prod />} />
```

```
  <Route path="/product/:id" element={<Singleproduct/>} />
```

```
  <Route path="/cart" element={<Cart />} />
```

```
  <Route path="/chatbot" element={<ChatbotButton />} />
```

```
  <Route path="/RelatedProducts" element={<RelatedProducts />} />
```

```
  <Route path="/Search" element={<Search />} />
```

```
</Routes>
```

```
<Footer />
```

```
</div>
```

```
);
```

```
}
```

```
export default App;
```

main.jsx

```
import React from "react";
```

```
import ReactDOM from "react-dom/client";
```

```
import { BrowserRouter } from "react-router-dom";
```

```
import App from "./App";
```

```
import "./index.css";
```

```
import AppProvider from "../utils/context"; // Updated import
```

```
ReactDOM.createRoot(document.getElementById("root")).render(  
  
  <React.StrictMode>  
  
    <BrowserRouter>  
  
      <AppProvider>  
  
        <App />  
  
      </AppProvider>  
  
    </BrowserRouter>  
  
  </React.StrictMode>  
  
);
```

Api.js

```
import axios from 'axios';
```

```
// Common headers for Axios requests
```

```
const params = {  
  
  headers: {  
  
    Authorization: `Bearer ${import.meta.env.VITE_STRIPE_APP_KEY}`,  
  
  },  
  
};
```

```
// Fetch products by category ID
```

```
export const fetchProductsByCategoryId = async (id) => {  
  
  console.log("Fetching products for category ID:", id); // Log the category ID being fetched  
  
  try {  
  
    const response = await fetch(  
  
      `http://localhost:1337/api/products?filters[category][id][$eq]=${id}&populate=*`  
  
    );  
  
    if (!response.ok) {  
  
      throw new Error(`HTTP error! status: ${response.status}`);  
  
    }  
  
    const result = await response.json();  
  
    console.log('Fetched products for category ID:', id, result.data); // Log fetched products
```

```

    return result.data;
  } catch (error) {
    console.error('Error fetching products by category ID:', error);
    throw error;
  }
};

// Generic function to fetch data from API using Axios
export const fetchDataFromApi = async (url) => {
  const fullUrl = `${import.meta.env.VITE_DEV_URL}${url}`;
  console.log("Fetching data from:", fullUrl);

  try {
    const response = await axios.get(fullUrl, params);
    console.log("API response:", response.data);
    return response.data;
  } catch (error) {
    console.error('Error fetching data from API:', error.response?.data || error.message);
    throw error;
  }
};

// Create an Axios instance for payment requests
export const makePaymentRequest = axios.create({
  baseURL: import.meta.env.VITE_DEV_URL,
  headers: {
    Authorization: `Bearer ${import.meta.env.VITE_STRIPE_APP_KEY}`,
  },
});

// Function to create an order
export const createOrder = async (orderData) => {
  try {
    const response = await makePaymentRequest.post('/api/orders', orderData);
    console.log('Order created successfully:', response.data); // Log successful order creation
    return response.data; // Return the order data
  } catch (error) {

```



```

    console.error('Error creating order:', error.response?.data || error.message);

    throw error;
  }
};

// Additional function to fetch product details (if needed)
export const fetchProductById = async (id) => {
  console.log("Fetching product with ID:", id); // Log the product ID being fetched

  try {
    const response = await fetch(`http://localhost:1337/api/products/${id}?populate=*`);

    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }

    const result = await response.json();

    console.log('Fetched product:', result.data); // Log fetched product

    return result.data;
  } catch (error) {
    console.error('Error fetching product by ID:', error);

    throw error;
  }
};

```

Context.js

```

import React, { createContext, useState, useContext, useEffect, useMemo } from "react";
import { useLocation } from "react-router-dom";

export const Context = createContext();

const AppProvider = ({ children }) => {
  const [categories, setCategories] = useState([]);

  const [products, setProducts] = useState([]);

  const [showCart, setShowCart] = useState(false);

  const [cartItems, setCartItems] = useState([]);

  const location = useLocation();

```

```

// Scroll to the top when location changes

useEffect(() => {

    window.scrollTo(0, 0);

}, [location]);


// Calculate total cart item count

const cartCount = useMemo(() => {

    return cartItems.reduce((count, item) => count + (item.attributes?.quantity || 0), 0);

}, [cartItems]);


// Calculate cart subtotal

const cartSubTotal = useMemo(() => {

    return cartItems.reduce((total, item) => total + (parseFloat(item.attributes?.price) || 0) * (item.attributes?.quantity || 0), 0);

}, [cartItems]);


// Handle adding items to the cart

const handleAddToCart = (product, quantity) => {

    setCartItems(prevItems => {

        const index = prevItems.findIndex(item => item.id === product.id);

        if (index !== -1) {

            // Update quantity if item exists

            const updatedItems = [...prevItems];

            updatedItems[index].attributes.quantity += quantity; // Increment the existing quantity

            return updatedItems;

        } else {

            // Add new product to cart

            return [

                ...prevItems,

                {

                    ...product,

                    attributes: {

                        ...product.attributes,

                        quantity, // Set initial quantity for the new product

                    },

                },

            ];

        }

    });

```

```

    }
  });
};

// Handle removing an item from the cart
const handleRemoveFromCart = (product) => {
  setCartItems(prevItems => prevItems.filter(item => item.id !== product.id));
};

// Handle incrementing/decrementing product quantity in the cart
const handleCartProductQuantity = (type, product) => {
  setCartItems(prevItems => {
    const index = prevItems.findIndex(item => item.id === product.id);
    if (index === -1) return prevItems; // Return current state if item not found

    const updatedItems = [...prevItems];
    if (type === "inc") {
      updatedItems[index].attributes.quantity += 1; // Increment quantity
    } else if (type === "dec") {
      if (updatedItems[index].attributes.quantity > 1) {
        updatedItems[index].attributes.quantity -= 1; // Decrement quantity, prevent going below 1
      } else {
        // Remove item if quantity is 1 and decremented
        return updatedItems.filter(item => item.id !== product.id);
      }
    }
  });
  return updatedItems;
};

return (
  <Context.Provider
    value={{
      products,
      setProducts,
      categories,
      setCategories,

```

```

        cartItems,

        handleAddToCart,

        cartCount, // Exposing cartCount

        handleRemoveFromCart,

        showCart,

        setShowCart,

        handleCartProductQuantity,

        cartSubTotal,

    }}

    >

    {children}

</Context.Provider>

);

};

```

```

// Custom hook for easier access to context

export const useAppContext = () => {

    return useContext(Context);

};

```

```

export default AppProvider;

```

index.html

```

. <!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<link rel="icon" type="image/svg+xml" href="/vite.svg" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>BHAVAA</title>

</head>

<body>

<div id="root"></div>

<script src="https://cdn.botpress.cloud/webchat/v2.1/inject.js"></script>

<script src="https://mediafiles.botpress.cloud/2c7c3470-aa41-49c7-bcbf-a8863d66f224/webchat/v2.1/config.js"></script>

<script src="https://checkout.razorpay.com/v1/checkout.js"></script>

```

```
<script type="module" src="/src/main.jsx"></script>

</body>

</html>
```

Vastu.jsx

```
import React from 'react';

import './VastuPage.css'; // Ensure you have the CSS file

const Vastu = () => {

  return (

    <div className="vastu-container">

      <h1>Vastu Analysis</h1>

      <iframe

        style={{

          background: 'white',

          borderRadius: '8px',

          border: 'none',

          boxShadow: '5px 5px 24px 0px #0000001F',

          width: '100%',

          height: '600px',

          maxWidth: '800px'

        }}

        src="https://embed.writer.com/text-generation/MPIBsTAEAnqxxFuEYxd7oc-q00GBKrSOdToVrrl8i4M"

        title="Vastu Analysis"

      ></iframe>

    </div>

  );

};
```

```
export default Vastu;
```

dailyfeed.jsx

```
import React, { useEffect, useState } from 'react';

import { db } from '../firebase';

import { collection, getDocs } from 'firebase/firestore';

import './dailyfeed.css';
```

```

const DailyFeed = () => {

  const [feeds, setFeeds] = useState([]);

  useEffect(() => {

    const fetchFeeds = async () => {

      try {

        const feedCollection = collection(db, 'dailyfeed');

        const feedSnapshot = await getDocs(feedCollection);

        const feedList = feedSnapshot.docs.map(doc => ({ id: doc.id, ...doc.data() }));

        setFeeds(feedList);

      } catch (error) {

        console.error("Error fetching daily feed data: ", error);

      }

    };

    fetchFeeds();

  }, []);

  return (

    <div className="cccontainer">

      {feeds.map((feed, index) => (

        <div key={feed.id} className={`cccard ${index % 2 === 0 ? 'cccard-left' : 'cccard-right'}`}>

          <div className="cccard-image">

            <img src={feed.imageUrl} alt={feed.title} />

          </div>

          <div className="cccard-content">

            <h1 className="cccard-number">{`0${index + 1}`}</h1>

            <h2 className="cccard-name">{feed.name}</h2>

            <h2 className="cccard-title">{feed.title}</h2>

            <p className="cccard-description">{feed.description}</p>

          </div>

        </div>

      ))}

    </div>

  );

};

```

```
export default DailyFeed;
```

```
ecommercehomepage.jsx
```

```
import React, { useEffect, useContext } from "react";  
import EcomHeader from '../ecomHeader/ecomHeader';  
import Banner from '../ecombanner/banner';  
import Category from '../Category/category';  
import Products from '../Products/products';  
import { fetchDataFromApi } from "../../utils/api";  
import { Context } from "../../utils/context";  
import './prod.css';
```

```
const Prod = () => {  
  const { categories, setCategories, products, setProducts } = useContext(Context);
```

```
  useEffect(() => {  
    const getCategories = async () => {  
      try {  
        const res = await fetchDataFromApi("/api/categories?populate=*");  
        if (res.data && Array.isArray(res.data)) {  
          setCategories(res.data);  
        } else {  
          setCategories([]);  
        }  
      } catch (error) {  
        console.error("Failed to fetch categories:", error);  
      }  
    };  
  });
```

```
  const getProducts = async () => {  
    try {  
      const res = await fetchDataFromApi("/api/products?populate=*");  
      if (res.data && Array.isArray(res.data)) {  
        setProducts(res.data);  
      } else {  
        setProducts([]);  
      }  
    }  
  }  
}
```

```

    } catch (error) {

      console.error("Failed to fetch products:", error);

    }

  };

  getCategories();

  getProducts();

}, [setCategories, setProducts]);

return (

  <div>

    <EcomHeader />

    <Banner />

    <div className='main-content'>

      <div className='layout'>

        <Category categories={categories} />

        <Products products={products} headingText="Popular Products" />

      </div>

    </div>

  </div>

);

};

export default Prod;

```

products.jsx

```

import React from 'react';

import './product.css'; // Add styles specific to the product

import { useNavigate } from "react-router-dom";

const Product = ({ product }) => {

  const { id, title, price, img } = product; // Ensure 'id' is included

  const navigate = useNavigate();

  return (

    <div className="product-card" onClick={() => navigate(`/product/${id}`)}>

      {img && img.length > 0 && img[0].url ? (

```



```

      <img src={` ${import.meta.env.VITE_DEV_URL}${img[0].url}`} alt={title} />
    ) : (
      <p>No image available</p>
    )}
    <div className="prod-details">
      <h3 className="name">{title}</h3>
      <span className="price">&#8377;{price}</span>
    </div>
  </div>
);
};

```

```
export default Product;
```

singleproducts.jsx

```

import React, { useContext, useState } from 'react';
import './singleproduct.css';
import { FaFacebookF, FaTwitter, FaInstagram, FaLinkedin, FaPinterest, FaCartPlus } from 'react-icons/fa';
import { BsCart } from 'react-icons/bs'; // Importing the cart icon
import RelatedProducts from "../RelatedProducts/RelatedProducts";
import useFetch from "../hooks/useFetch";
import { useParams } from 'react-router-dom';
import { Context } from '../utils/context';
import Cart from '../cart/Cart'; // Import the Cart component

```

```

const SingleProduct = () => {
  const [quantity, setQuantity] = useState(1);
  const [showCart, setShowCart] = useState(false); // State to control cart visibility
  const { id } = useParams();
  const { data, loading, error } = useFetch(`/api/products?populate=*&[filters][id]=${id}`);
  const { handleAddToCart } = useContext(Context);

  // Increment and decrement quantity
  const increment = () => {

```

```

    setQuantity(prevState => prevState + 1);
};

const decrement = () => {
    setQuantity(prevState => (prevState > 1 ? prevState - 1 : 1)); // Prevent quantity from going below 1
};

// Handle loading and error states
if (loading) {
    return <p className="loading">Loading...</p>;
}

if (error) {
    return <p className="error">Error fetching product data.</p>;
}

if (!data || !data.data || data.data.length === 0) {
    return <p className="error">No product found.</p>;
}

const productData = data.data[0];
const { title, price, desc, img, categories } = productData;

// Construct image URL
const imageUrl = img?.[0]?.url ? `${import.meta.env.VITE_DEV_URL}${img[0].url}` :
'/path/to/default/image.jpg'; // Default image path

// Safely access categories and join their names if they exist
const categoryNames = categories && categories.length > 0
    ? categories.map(category => category.title).join(', ')
    : 'N/A';

// Get the category ID for related products

```

```
const categoryId = categories?.[0]?.id; // Assuming you want the first category
```

```
return (
```

```
<div className="singleproduct-maincontent">
```

```
<div className="layout">
```

```
<div className="singleproduct-page">
```

```
  {/* Left section: Product Image */}
```

```
<div className="left">
```

```
<img src={imageUrl} alt={title} />
```

```
</div>
```

```
  {/* Right section: Product Details */}
```

```
<div className="right">
```

```
<h1 className="name">{title}</h1>
```

```
<h2 className="price">₹{price}</h2>
```

```
<p className="desc">{desc}</p>
```

```
<div className="cart-buttons">
```

```
<div className="quantity-buttons">
```

```
<button className="quantity-btn" onClick={decrement} disabled={quantity === 1}></button>
```

```
<span className="quantity-display">{quantity}</span>
```

```
<button className="quantity-btn" onClick={increment}>+</button>
```

```
</div>
```

```
<button className="add-to-cart-button" onClick={() => handleAddToCart(productData, quantity)}>
```

```
<FaCartPlus size={20} /> ADD TO CART
```

```
</button>
```

```
</div>
```

```
<span className="divider" />
```

```
  {/* Product Category and Social Share */}
```

```
<div className="info-item">
```

```
<span className="text-bold">
```

```

        Category: <span>{categoryNames}</span>
    </span>
    <span className="text-bold">
        SHARE:
        <span className="social-icons">
            <FaFacebookF size={16} />
            <FaTwitter size={16} />
            <FaInstagram size={16} />
            <FaLinkedin size={16} />
            <FaPinterest size={16} />
        </span>
    </span>
</div>
</div>
</div>
</div>

{/* Cart Icon Button */}
<button className="cart-icon" onClick={() => setShowCart(prev => !prev)} aria-label="Toggle Cart">
    <BsCart size={30} className={showCart ? 'active' : ''} /> {/* Highlighted if cart is shown */}
</button>

{/* Cart Component */}
{showCart && <Cart setShowCart={setShowCart} />} {/* Ensure you have this component */}

{/* Related Products Section */}
{categoryId && <RelatedProducts productId={id} categoryId={categoryId} />}
</div>
);
};

export default SingleProduct;

```

reyna.jsx

import React from 'react'

const reyna = () => {

return (

<div>

<script src="https://cdn.botpress.cloud/webchat/v2.1/inject.js"></script>

<script src="https://mediafiles.botpress.cloud/2c7c3470-aa41-49c7-bcbf-a8863d66f224/webchat/v2.1/config.js"></script>

</div>

)

}

export default reyna

login.jsx

import { GoogleAuthProvider, signInWithPopup } from 'firebase/auth'

import React, { useState } from 'react';

import './login.css'

import { auth } from '../firebase'

export default function SigIn() {

function googleLogin() {

const provider=new GoogleAuthProvider

```

signInWithPopup(auth,provider).then(async(result)=>{
    console.log(result)

})
}
return (

// <div>

//   

// </div>

<div className="login-container">
<div className="login-box">
  <h2>Login</h2>
  <form>
    <div className="input-group">
      <input type="email" placeholder="Enter your email" required />
    </div>
    <div className="input-group">
      <input type="password" placeholder="Enter your password" required />
    </div>
    <div className="remember-forgot">
      <label>
        <input type="checkbox" />

```

```

        Remember me
      </label>
      <a href="#">Forgot password?</a>
    </div>
    <button className="ebutton" type="submit">Log In</button>
  </form>
  <p>Don't have an account? <a href="#" onClick={googleLogin}>Register</a></p>
</div>
</div>
)
}

Indivcategory.jsx
import React from 'react'; // Add this line
import { useParams } from "react-router-dom";
import useFetch from "../src/hooks/useFetch";
import Products from "../Products/products";
import "../indivcategory.css";

const IndiviCategory = () => {
  const { id } = useParams();
  const { data, loading, error } = useFetch(
    `/api/products?populate=*&[filters][categories][id]=${id}` // Fetch products based on
category ID
  );

  return (
    <div className="category-main-content">
      <div className="layout">
        {loading && <div>Loading...</div>}

```

```
{error && <div>Error: {error.message}</div>}
```

```
{/* Display category title if data is available */}
```

```
{data?.data?.[0]?.attributes?.categories?.data?.length > 0 && (
```

```
  <div className="category-title">
```

```
    {data.data[0].attributes.categories.data[0].attributes.title}
```

```
  </div>
```

```
))
```

```
{/* Render products if data is available */}
```

```
  {data && <Products innerPage={true} products={data.data} />} {/* Ensure to pass  
the correct products array */}
```

```
  </div>
```

```
</div>
```

```
);
```

```
};
```

```
export default IndiviCategory;
```

```
useFetch.js
```

```
import { useEffect, useState } from "react";
```

```
import { fetchDataFromApi } from "../../utils/api";
```

```
const useFetch = (endpoint) => {
```

```
  const [data, setData] = useState(null);
```

```
  const [loading, setLoading] = useState(true);
```

```
  const [error, setError] = useState(null);
```

```
  useEffect(() => {
```



```

const makeApiCall = async () => {
  setLoading(true);
  try {
    const res = await fetchDataFromApi(endpoint);
    if (res) {
      setData(res);
    } else {
      throw new Error("No data received");
    }
  } catch (err) {
    setError(err);
  } finally {
    setLoading(false);
  }
};

makeApiCall();
}, [endpoint]);

return { data, loading, error };
};

export default useFetch;

eCommerceHeader.jsx
import React, { useState, useEffect } from 'react'; // Import useState and useEffect
import { TbSearch } from 'react-icons/tb'; // Import icons
import { CgShoppingCart } from 'react-icons/cg'; // Import icons
import { AiOutlineHeart } from 'react-icons/ai'; // Import icons

```

```
import { useNavigate } from 'react-router-dom';  
import './ecomHeader.css'; // Ensure this path is correct  
import Search from '../search/search';  
import Cart from '../cart/Cart';  
import { useAppContext } from '../../utils/context'; // Adjust the import path according to  
your structure
```

```
const EcomHeader = () => {  
  const navigate = useNavigate(); // Correctly initialize navigate  
  const [scrolled, setScrolled] = useState(false);  
  const [ShowCart, setShowCart] = useState(false);  
  const [ShowSearch, setShowSearch] = useState(false);  
  const { cartCount } = useAppContext(); // Get cartCount from context
```

```
  const handleScroll = () => {  
    const offset = window.scrollY;  
    if (offset > 200) {  
      setScrolled(true);  
    } else {  
      setScrolled(false);  
    }  
  };  
};
```

```
useEffect(() => {  
  window.addEventListener("scroll", handleScroll);  
  
  return () => {  
    window.removeEventListener("scroll", handleScroll);  
  };  
};
```

```
}, []); // Empty dependency array ensures this runs once on mount
```

```
return (
```

```
  <>
```

```
    <header className={`main-header${scrolled ? ' sticky-header' : ''}>
```

```
      <div className="header-content">
```

```
        <ul>
```

```
          <li onClick={() => navigate("/")}>Home</li>
```

```
          <li>Cateegory</li>
```

```
        </ul>
```

```
      <div className="center"></div>
```

```
      <div className="right">
```

```
        <TbSearch onClick={() => setShowSearch(true)} />
```

```
        <AiOutlineHeart />
```

```
        <span className="cart-icon" onClick={() => setShowCart(true)}>
```

```
          <CgShoppingCart />
```

```
          {cartCount > 0 && <span className="cart-count">{cartCount}</span>} {/*
```

```
Display cart count */}
```

```
        </span>
```

```
      </div>
```

```
    </div>
```

```
  </header>
```

```
  {ShowCart && <Cart setShowCart={setShowCart} />}
```

```
  {ShowSearch && <Search setShowSearch={setShowSearch} />}
```

```
  </>
```

```
);
```

```
};
```

```
export default EcomHeader;
```

```
cart.jsx import React, { useContext, useState } from "react";  
import "./Cart.css";  
import { MdClose } from "react-icons/md";  
import { BsCart } from "react-icons/bs";  
import CartItem from './cartitem/cartitem';  
import { Context } from '../utils/context';  
import axios from 'axios';
```

```
const Cart = ({ setShowCart }) => {  
  const { cartItems } = useContext(Context);  
  const [error, setError] = useState(null);  
  
  // Function to handle Razorpay payment  
  const handlePayment = async () => {  
    if (!cartItems || cartItems.length === 0) {  
      alert("Your cart is empty.");  
      return;  
    }  
  }
```

```
  const subtotal = calculateSubtotal();
```

```
  const options = {  
    key: 'rzp_live_TqdzegDnrClaRh', // Razorpay Key ID  
    amount: subtotal * 100, // Subtotal in paise (1 INR = 100 paise)  
    currency: 'INR',  
    name: 'Bhavatu Store',  
    description: 'Purchase Description',  
    image: 'https://your-logo-url',
```

```
handler: async function (response) {  
  // Payment success callback  
  await saveOrder(response.razorpay_payment_id);  
},  
prefill: {  
  name: "Customer Name",  
  email: "customer@example.com",  
  contact: "9999999999",  
},  
theme: {  
  color: "#3399cc",  
},  
};
```

```
const rzp = new window.Razorpay(options);  
rzp.open();
```

```
rzp.on('payment.failed', function (response) {  
  alert(`Payment failed! Error: ${response.error.description}`);  
});  
};
```

```
// Calculate cart subtotal
```

```
const calculateSubtotal = () => {  
  return cartItems.reduce((total, item) => {  
    const price = parseFloat(item.price) || 0;  
    const quantity = parseInt(item.attributes?.quantity) || 1;  
    return total + price * quantity;  
  }, 0);
```

```
};
```

```
const subtotal = calculateSubtotal();
```

```
// Function to save order details to Strapi
```

```
const saveOrder = async (paymentId) => {
```

```
  const orderData = {
```

```
    PaymentId: paymentId,
```

```
    CustomerName: "Customer Name", // Replace with dynamic data if needed
```

```
    CustomerEmail: "customer@example.com", // Replace with dynamic data if needed
```

```
    CustomerContact: "9999999999", // Replace with dynamic data if needed
```

```
    Subtotal: subtotal,
```

```
    Items: cartItems.map(item => ({
```

```
      productId: item.id,
```

```
      quantity: item.attributes?.quantity || 1,
```

```
      price: item.price,
```

```
    })),
```

```
    Date: new Date(),
```

```
};
```

```
try {
```

```
  const response = await axios.post('http://localhost:1337/orders', orderData, {
```

```
    headers: {
```

```
      'Authorization': `Bearer YOUR_API_TOKEN`, // Ensure this is correct
```

```
    },
```

```
  });
```

```
  console.log('Order saved response:', response.data); // Log response for debugging
```

```
  alert("Order saved successfully!"); // Notify user of success
```

```

        setShowCart(false); // Close the cart after successful order
    } catch (err) {
        console.error("Error saving order:", err.response ? err.response.data : err);
        setError("Failed to save order. Please try again."); // Set error message
    }
};

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

            {cartItems.length === 0 ? (
                <div className="emptycart">
                    <BsCart size={50} style={{ color: '#8e2de2' }} />
                    <span>No products in the cart.</span>
                    <p>Please add some products to your cart.</p>
                    <button className='return-cta' onClick={() => setShowCart(false)}>Return to
shop</button>
                </div>
            ) : (
                <>

```

```

    {cartItems.map(item => (
      <CartItem key={item.id} item={item} />
    ))}
    <div className="cart-footer">
      <div className="subtotal">
        <span className="text">Subtotal:</span>
        <span className="text-total"> &#8377;{subtotal.toFixed(2)}</span>
      </div>
      <div className="button">
        <button className='checkout-cta'
onClick={handlePayment}>Checkout</button>
      </div>
      {error && <div className="error-message">{error}</div>}
    </div>
  </>
)}
</div>
</div>
);
};

```

export default Cart;

cart.css

```

.cart-panel {
  position: fixed;
  top: 0;
  right: 0;
  width: 400px; /* Fixed width for the cart panel */
}

```



```
height: 100%; /* Full height */
z-index: 100; /* Ensures it appears above other content */
display: flex;
justify-content: flex-end; /* Align content to the right */
overflow: hidden; /* Hide overflow content */
background-color: rgba(255, 255, 255, 0.95); /* Light background with slight transparency */
box-shadow: -5px 0px 15px rgba(0, 0, 0, 0.2); /* Shadow for depth */
}
```

```
.cart-content {
width: 400px; /* Fixed width for the content */
height: 100%; /* Full height */
background-color: #f4f6f9; /* Light background color */
position: relative;
display: flex;
flex-direction: column; /* Stack items vertically */
animation: slideCartWindow 0.4s ease forwards; /* Slide-in animation */
z-index: 1; /* Ensure it sits above the panel */
}
```

```
@keyframes slideCartWindow {
0% {
transform: translateX(100%); /* Start off-screen */
}
100% {
transform: translateX(0); /* Slide in */
}
}
```

```
.cart-header {  
  display: flex;  
  justify-content: space-between; /* Space out header items */  
  padding: 20px; /* Padding for header */  
  background-color: #8e2de2; /* Header background color */  
  color: white; /* Header text color */  
  font-weight: bold; /* Bold text */  
  font-size: 20px; /* Header font size */  
  align-items: center; /* Center items vertically */  
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); /* Shadow for separation */  
}
```

```
.close-button {  
  font-size: 20px; /* Close button size */  
  cursor: pointer; /* Pointer on hover */  
  color: white; /* Close button color */  
  transition: opacity 0.3s ease; /* Smooth opacity transition */  
}
```

```
.close-button:hover {  
  opacity: 0.7; /* Slightly transparent on hover */  
}
```

```
.cart-body {  
  padding: 20px; /* Padding for body */  
  flex-grow: 1; /* Allow body to grow and fill space */  
  overflow-y: auto; /* Enable vertical scrolling if needed */  
}
```

```
.cart-footer {  
  border-top: 1px solid rgba(0, 0, 0, 0.1); /* Light border at the top */  
  padding: 20px 15px; /* Padding for footer */  
  margin-top: auto; /* Push footer to the bottom */  
}
```

```
.subtotal {  
  display: flex; /* Flex layout for subtotal */  
  justify-content: space-between; /* Space between items */  
  align-items: center; /* Center items vertically */  
}
```

```
.subtotal .text {  
  font-size: 20px; /* Subtotal text size */  
  font-weight: 800; /* Bold text */  
  text-transform: uppercase; /* Uppercase text */  
  color: #333; /* Dark color for contrast */  
}
```

```
.subtotal .text-total {  
  font-size: 20px; /* Total text size */  
  font-weight: 400; /* Regular weight */  
  color: #8e2de2; /* Accent color for total */  
}
```

```
.button {  
  display: flex;  
  justify-content: center; /* Center the button */  
}
```

```
margin-top: 20px; /* Space above the button */  
}
```

```
.checkout-cta {  
  background-color: #8e2de2; /* Button background color */  
  color: white; /* Button text color */  
  padding: 15px 30px; /* Standard button padding */  
  font-size: 20px; /* Button text size */  
  border-radius: 5px; /* Rounded corners */  
  cursor: pointer; /* Pointer on hover */  
  border: none; /* Remove border */  
  transition: background-color 0.3s ease; /* Smooth background transition */  
}
```

```
.checkout-cta:hover {  
  background-color: #6516aa; /* Darker shade on hover */  
}
```

```
.opac-layer {  
  display: none; /* Layer visibility */  
}
```

```
.emptycart {  
  display: flex;  
  flex-direction: column; /* Stack items vertically */  
  justify-content: center; /* Center vertically */  
  align-items: center; /* Center horizontally */  
  height: calc(100vh - 150px); /* Full viewport height minus header/footer */  
  text-align: center; /* Center text */  
}
```

```
padding: 20px; /* Padding for spacing */
background-color: #f4f6f9; /* Light background color */
border-radius: 8px; /* Rounded corners */
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1); /* Subtle shadow for depth */
}
```

```
.emptycart span {
  font-size: 20px; /* Adjusted font size for the message */
  color: #333; /* Dark color for text */
  margin: 15px 0; /* Space above and below the message */
}
```

```
.emptycart .return-cta {
  background-color: #8e2de2; /* Main color for the button */
  color: white; /* Button text color */
  padding: 10px 20px; /* Button padding */
  font-size: 18px; /* Button font size */
  border: none; /* No border */
  border-radius: 5px; /* Rounded corners */
  cursor: pointer; /* Pointer cursor on hover */
  transition: background-color 0.3s ease; /* Transition for hover effect */
}
```

```
.emptycart .return-cta:hover {
  background-color: #6516aa; /* Darker shade on hover */
}
```

```
Cartitem.jsx import React, { useContext } from "react";
import { MdClose } from "react-icons/md";
```

```

import { Context } from "../../utils/context";
import "./cartitem.css";

const CartItem = ({ item }) => {
  const { handleRemoveFromCart, handleCartProductQuantity } = useContext(Context);

  // Access price and quantity directly from the item, ensuring safety
  const price = parseFloat(item.price) || 0; // Ensure price is a float
  const quantity = parseInt(item.attributes?.quantity) || 1; // Default to 1 if quantity is
  undefined

  return (
    <div className="cart-product">
      <div className="img-container">
        <img
          src={`${import.meta.env.VITE_DEV_URL}${item.img[0].url}`}
          alt={item.title}
        />
      </div>
      <div className="product-details">
        <span className="name">{item.title}</span>
        <MdClose
          className="close-btn"
          onClick={(e) => {
            e.stopPropagation();
            handleRemoveFromCart(item);
          }}
        />
        <div className="quantity-buttons">

```

```

    <span
      onClick={() => handleCartProductQuantity("dec", item)}
      className="quantity-btn"
    >
      -
    </span>
    <span>{quantity}</span>
    <span
      onClick={() => handleCartProductQuantity("inc", item)}
      className="quantity-btn"
    >
      +
    </span>
  </div>
  <div className="text">
    <span>{quantity}</span>
    <span>x</span>
    <span className="highlight">
      <span>&#8377;</span>
      {(price * quantity).toFixed(2)} { /* Calculate item total */ }
    </span>
  </div>
</div>
</div>
);
};

export default CartItem;

```

cartitem.css

```
.cart-products {  
    display: flex;  
    flex-direction: column;  
    gap: 15px; /* Space between each cart item */  
}
```

```
.cart-product {  
    display: flex;  
    align-items: center; /* Center items vertically */  
    padding: 10px; /* Reduced padding for compactness */  
    background-color: rgba(0, 0, 0, 0.02); /* Light background for each product */  
    border-radius: 8px; /* Rounded corners */  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); /* Subtle shadow for depth */  
    transition: background-color 0.3s; /* Smooth transition on hover */  
}
```

```
.cart-product:hover {  
    background-color: rgba(0, 0, 0, 0.05); /* Darker shade on hover for visual feedback */  
}
```

```
.img-container {  
    background-color: rgba(0, 0, 0, 0.1);  
    width: 50px; /* Adjusted width for smaller images */  
    height: 50px; /* Adjusted height for smaller images */  
    flex-shrink: 0;  
    border-radius: 5px; /* Rounded corners for the image container */  
    overflow: hidden; /* Ensures image doesn't overflow */  
}
```



```
.img-container img {  
  width: 100%;  
  height: 100%;  
  object-fit: cover; /* Maintains aspect ratio */  
}
```

```
.product-details {  
  flex-grow: 1; /* Allows text to take up remaining space */  
  padding-left: 10px; /* Space between image and text */  
  display: flex;  
  flex-direction: column; /* Stack details vertically */  
}
```

```
.name {  
  font-size: 16px; /* Adjusted font size for product name */  
  font-weight: bold; /* Emphasized font */  
  color: #333; /* Dark color for product name */  
}
```

```
.close-btn {  
  cursor: pointer; /* Pointer on hover */  
  margin-left: auto; /* Push to the right */  
  color: #8e2de2; /* Accent color for close button */  
  transition: color 0.3s; /* Smooth color change */  
}
```

```
.close-btn:hover {  
  color: #6516aa; /* Darker shade on hover */  
}
```

```
}
```

```
.quantity-buttons {  
    width: fit-content;  
    display: flex;  
    border: 2px solid rgba(0, 0, 0, 0.2);  
    margin-right: 10px;  
    height: 40px; /* Slightly reduced height for compactness */  
    border-radius: 5px; /* Rounded corners */  
    align-items: center; /* Center buttons vertically */  
}
```

```
.quantity-btn {  
    font-size: 18px;  
    width: 40px; /* Fixed width for buttons */  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    cursor: pointer;  
    color: black;  
    transition: background-color 0.3s; /* Smooth background change */  
}
```

```
.quantity-btn:hover {  
    background-color: rgba(0, 0, 0, 0.1); /* Light background on hover */  
}
```

```
.quantity-btn:first-child {  
    border-right: 1px solid rgba(0, 0, 0, 0.2); /* Right border for the decrement button */  
}
```

```
}
```

```
.quantity-btn:last-child {  
  border-left: 1px solid rgba(0, 0, 0, 0.2); /* Left border for the increment button */  
}
```

```
.text {  
  font-size: 14px; /* Adjusted font size for pricing info */  
  color: #333; /* Dark color for text */  
  margin-top: 5px; /* Space above price text */  
  display: flex; /* Use flex for better alignment */  
  align-items: center; /* Center vertically */  
}
```

```
.highlight {  
  font-weight: bold; /* Emphasized total price */  
  margin-left: 5px; /* Space between quantity and price */  
  color: #e74c3c; /* Highlighted price color */  
}
```

Navbar.jsx

```
import React from "react";  
import "./Navbar.css";  
import { Link } from "react-router-dom";
```

```
const Navbar = () => {  
  return (  

```

```
<nav class="navbar">
```

```

<div class="logo-container">
  {/*  */}
  <span class="brand-text">BHAVATU</span>
</div>

<div class="tabs-container">
  <Link class="tab" to="/">Home</Link>
  <Link class="tab" to="/vastu">Vastu</Link>

  <Link class="tab" to="/prod">Products</Link>
  {/* <Link class="tab" to="/Cart">Cart</Link> */}
  <Link class="tab" to="/kundli">Kundli</Link>
  <Link class="tab" to="/DailyFeed">DailyFeed</Link>
  <Link class="tab" to="/Login">Login</Link>

</div>
</nav>

);
};

```

```
export default Navbar;
```

admin.ts

```

export default ({ env }) => ({
  auth: {
    secret: env('ADMIN_JWT_SECRET'),
  },
  apiToken: {
    salt: env('API_TOKEN_SALT'),
  },

```

```
transfer: {  
  token: {  
    salt: env('TRANSFER_TOKEN_SALT'),  
  },  
},  
flags: {  
  nps: env.bool('FLAG_NPS', true),  
  promoteEE: env.bool('FLAG_PROMOTE_EE', true),  
},  
});
```

```
export default {  
  rest: {  
    defaultLimit: 25,  
    maxLimit: 100,  
    withCount: true,  
  },  
};
```

Database.ts

```
import path from 'path';
```

```
export default ({ env }) => {  
  const client = env('DATABASE_CLIENT', 'sqlite');  
  
  const connections = {  
    mysql: {  
      connection: {  
        host: env('DATABASE_HOST', 'localhost'),
```

```
port: env.int('DATABASE_PORT', 3306),
database: env('DATABASE_NAME', 'strapi'),
user: env('DATABASE_USERNAME', 'strapi'),
password: env('DATABASE_PASSWORD', 'strapi'),
ssl: env.bool('DATABASE_SSL', false) && {
  key: env('DATABASE_SSL_KEY', undefined),
  cert: env('DATABASE_SSL_CERT', undefined),
  ca: env('DATABASE_SSL_CA', undefined),
  capath: env('DATABASE_SSL_CAPATH', undefined),
  cipher: env('DATABASE_SSL_CIPHER', undefined),
  rejectUnauthorized: env.bool('DATABASE_SSL_REJECT_UNAUTHORIZED', true),
},
},
pool: { min: env.int('DATABASE_POOL_MIN', 2), max: env.int('DATABASE_POOL_MAX', 10)
},
},
postgres: {
  connection: {
    connectionString: env('DATABASE_URL'),
    host: env('DATABASE_HOST', 'localhost'),
    port: env.int('DATABASE_PORT', 5432),
    database: env('DATABASE_NAME', 'strapi'),
    user: env('DATABASE_USERNAME', 'strapi'),
    password: env('DATABASE_PASSWORD', 'strapi'),
    ssl: env.bool('DATABASE_SSL', false) && {
      key: env('DATABASE_SSL_KEY', undefined),
      cert: env('DATABASE_SSL_CERT', undefined),
      ca: env('DATABASE_SSL_CA', undefined),
      capath: env('DATABASE_SSL_CAPATH', undefined),
```

```

    cipher: env('DATABASE_SSL_CIPHER', undefined),
    rejectUnauthorized: env.bool('DATABASE_SSL_REJECT_UNAUTHORIZED', true),
  },
  schema: env('DATABASE_SCHEMA', 'public'),
},
pool: { min: env.int('DATABASE_POOL_MIN', 2), max: env.int('DATABASE_POOL_MAX', 10)
},
},
sqlite: {
  connection: {
    filename: path.join(__dirname, '..', '..', env('DATABASE_FILENAME', '.tmp/data.db')),
  },
  useNullAsDefault: true,
},
};

```

```

return {
  connection: {
    client,
    ...connections[client],
    acquireConnectionTimeout: env.int('DATABASE_CONNECTION_TIMEOUT', 60000),
  },
};
};

```

Middleware.ts

```

export default [
  'strapi::logger',
  'strapi::errors',

```

```
'strapi::security',  
'strapi::cors',  
'strapi::poweredBy',  
'strapi::query',  
'strapi::body',  
'strapi::session',  
'strapi::favicon',  
'strapi::public',  
];
```

Index.html

```
<!doctype html>  
<html lang="en">
```

```
<!--
```

This file was automatically generated by Strapi.
Any modifications made will be discarded.

```
-->
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<meta
```

```
  name="viewport"
```

```
  content="width=device-width, initial-scale=1, viewport-fit=cover"
```

```
/>
```

```
<meta name="robots" content="noindex" />
```

```
<meta name="referrer" content="same-origin" />
```

```
<title>Strapi Admin</title>
```

```
<style>
```

```
  html,
```

```
  body,
```



```
#strapi {
  height: 100%;
}

body {
  margin: 0;
  -webkit-font-smoothing: antialiased;
}

</style>
</head>
<body>

<div id="strapi"></div>

<noscript

><div class="strapi--root">

  <div class="strapi--no-js">

    <style type="text/css">

      .strapi--root {

        position: absolute;

        top: 0;

        right: 0;

        left: 0;

        bottom: 0;

        background: #fff;

      }

      .strapi--no-js {

        position: absolute;

        top: 50%;

        left: 50%;

        transform: translate(-50%, -50%);
```

```

    text-align: center;

    font-family: helvetica, arial, sans-serif;

  }
</style>

<h1>JavaScript disabled</h1>

<p>

  Please

  <a href="https://www.enable-javascript.com/">enable JavaScript</a>

  in your browser and reload the page to proceed.

</p>

</div>

</div></noscript>

>

<script type="module" src="/.strapi/client/app.js"></script>

</body>

</html>

```

App.js /**

* This file was automatically generated by Strapi.

* Any modifications made will be discarded.

*/

import strapiCloud from "@strapi/plugin-cloud/strapi-admin";

import usersPermissions from "@strapi/plugin-users-permissions/strapi-admin";

import { renderAdmin } from "@strapi/strapi/admin";

renderAdmin(document.getElementById("strapi"), {

plugins: {

"strapi-cloud": strapiCloud,

"users-permissions": usersPermissions,

```
},  
});
```

```
Order.ts import { factories } from '@strapi/strapi';
```

```
// Create a core controller for orders
```

```
export default factories.createCoreController('api::order.order', ({ strapi }) => ({
```

```
  // Override the default `find` function if needed
```

```
  async find(ctx) {
```

```
    try {
```

```
      // You can add custom logic here if necessary
```

```
      const { query } = ctx;
```

```
      // Call the default find service
```

```
      const orders = await strapi.entityService.findMany('api::order.order', {
```

```
        filters: query.filters,
```

```
        populate: query.populate || '*', // Adjust based on relationships or populate settings
```

```
      });
```

```
      return orders;
```

```
    } catch (error) {
```

```
      ctx.throw(500, 'An error occurred while fetching orders.');
```

```
    }
```

```
  },
```

```
  // Add other methods like findOne, create, update, delete as needed
```

```
});
```

```
Firebase.jsx // src/firebase.js
```

```
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";
import { getAuth, GoogleAuthProvider, signInWithPopup } from "firebase/auth";

const firebaseConfig = {
  apiKey: "AlzaSyA_mZ2HpOYUQxWMicmvSNup5TRlaJnzvfs",
  authDomain: "bhavatucom.firebaseio.com",
  projectId: "bhavatucom",
  storageBucket: "bhavatucom.appspot.com",
  messagingSenderId: "747578764267",
  appId: "1:747578764267:web:7940801253582080bdc7ae",
  measurementId: "G-VKQXE9YC5X"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

// Initialize Firestore
export const db = getFirestore(app);

// Initialize Firebase Authentication
export const auth = getAuth(app);

// Export Google Sign-In function
export const signInWithGoogle = async () => {
  const provider = new GoogleAuthProvider();
  try {
    const result = await signInWithPopup(auth, provider);
```

```

    console.log('User signed in: ', result.user);
    // Here you can handle the signed-in user
  } catch (error) {
    console.error('Error signing in with Google: ', error);
  }
};

```

Product.jsx

```

import React from 'react';
import './product.css'; // Add styles specific to the product
import { useNavigate } from "react-router-dom";

const Product = ({ product }) => {
  const { id, title, price, img } = product; // Ensure 'id' is included
  const navigate = useNavigate();

  return (
    <div className="product-card" onClick={() => navigate(`/product/${id}`)}>
      {img && img.length > 0 && img[0].url ? (
        <img src={`${import.meta.env.VITE_DEV_URL}${img[0].url}`} alt={title} />
      ) : (
        <p>No image available</p>
      )}
      <div className="prod-details">
        <h3 className="name">{title}</h3>
        <span className="price">&#8377;{price}</span>
      </div>
    </div>
  );
};

```

```
};
```

```
export default Product;
```

```
search.jsx
```

```
import { useState } from 'react';
```

```
import React from 'react';
```

```
import { MdClose } from 'react-icons/md';
```

```
import './search.css';
```

```
import { useNavigate } from 'react-router-dom';
```

```
import useFetch from "../hooks/useFetch";
```

```
const Search = ({ setShowSearch }) => {
```

```
  const [query, setQuery] = useState("");
```

```
  const navigate = useNavigate();
```

```
  // Use custom hook to fetch products based on the query
```

```
  const { data, loading, error } =
```

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

```
  const onChange = (e) => {
```

```
    setQuery(e.target.value);
```

```
  };
```

```
  return (
```

```
    <div className='search-model'>
```

```
      <div className="form-field">
```

```
        <input
```

```
          type="text"
```

```

    autoFocus
    placeholder='Search for product'
    value={query}
    onChange={onChange} // Updated to handle input change
  />
  <MdClose className='close-btn' onClick={() => setShowSearch(false)} />
</div>
<div className="search-result-content">
  <div className="search-result">
    {loading && <p>Loading...</p>} {/* Show loading state */}
    {error && <p>Error fetching products.</p>} {/* Show error state */}
    {data && data.data && data.data.length > 0 ? (
      data.data.map(item => (
        <div key={item.id} className="search-result-item" onClick={() =>
navigate(`/product/${item.id}`)}>
          <div className="img-container">
            {item.img && item.img[0]?.url ? ( // Ensure image exists and access the first image
              <img src={` ${import.meta.env.VITE_DEV_URL}${item.img[0].url}`} alt={item.title
|| "Product Image"} />
            ) : (
              <p>No image available</p> // Fallback if no image
            )}
          </div>
          <div className="product-details">
            <span className="name">{item.title}</span>
            <span className="desc">{item.description || "Product description goes
here"}</span> {/* Added description */}
            <span className="price">&#8377;{item.price}</span> {/* Display price if
available */}
          </div>

```

```

        </div>
    ))
    ) : (
        query.length > 0 && <p>No products found</p> // Display message when no results
        match the query
    )}
</div>
</div>
</div>
);
};

```

export default Search;

footer.jsx

import React from 'react'

import './footer.css'

function Footer() {

return (

<div className='all'>

<footer className="footer">

<div className="waves">

<div className="wave" id="wave1"></div>

<div className="wave" id="wave2"></div>

<div className="wave" id="wave3"></div>

<div className="wave" id="wave4"></div>

</div>


```
<ul className="social-icon">
  <li className="social-icon__item"><a className="social-icon__link" href="#">
    <ion-icon name="logo-facebook"></ion-icon>
  </a></li>
  <li className="social-icon__item"><a className="social-icon__link" href="#">
    <ion-icon name="logo-twitter"></ion-icon>
  </a></li>
  <li className="social-icon__item"><a className="social-icon__link" href="#">
    <ion-icon name="logo-linkedin"></ion-icon>
  </a></li>
  <li className="social-icon__item"><a className="social-icon__link" href="#">
    <ion-icon name="logo-instagram"></ion-icon>
  </a></li>
</ul>

<ul className="menu">
  <li className="menu__item"><a className="menu__link" href="#">Home</a></li>
  <li className="menu__item"><a className="menu__link" href="#">About</a></li>
  <li className="menu__item"><a className="menu__link" href="#">Services</a></li>
  <li className="menu__item"><a className="menu__link" href="#">Team</a></li>
  <li className="menu__item"><a className="menu__link" href="#">Contact</a></li>

</ul>

<p>&copy;2021 Nadine Coelho | All Rights Reserved</p>

</footer>

<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>

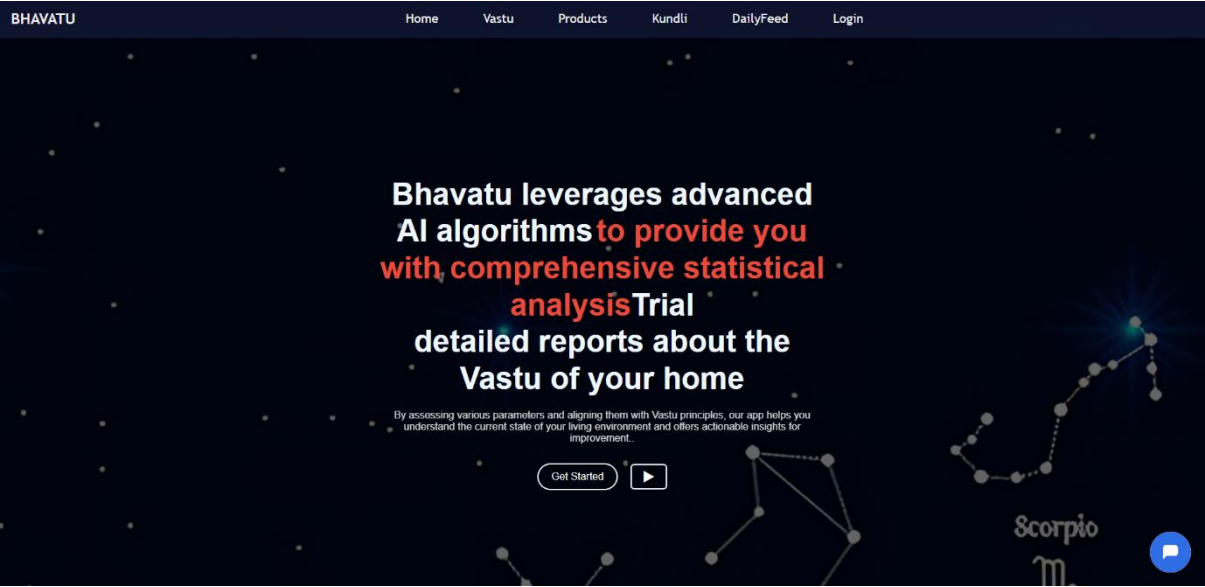
<script noModule
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>

</div>
```

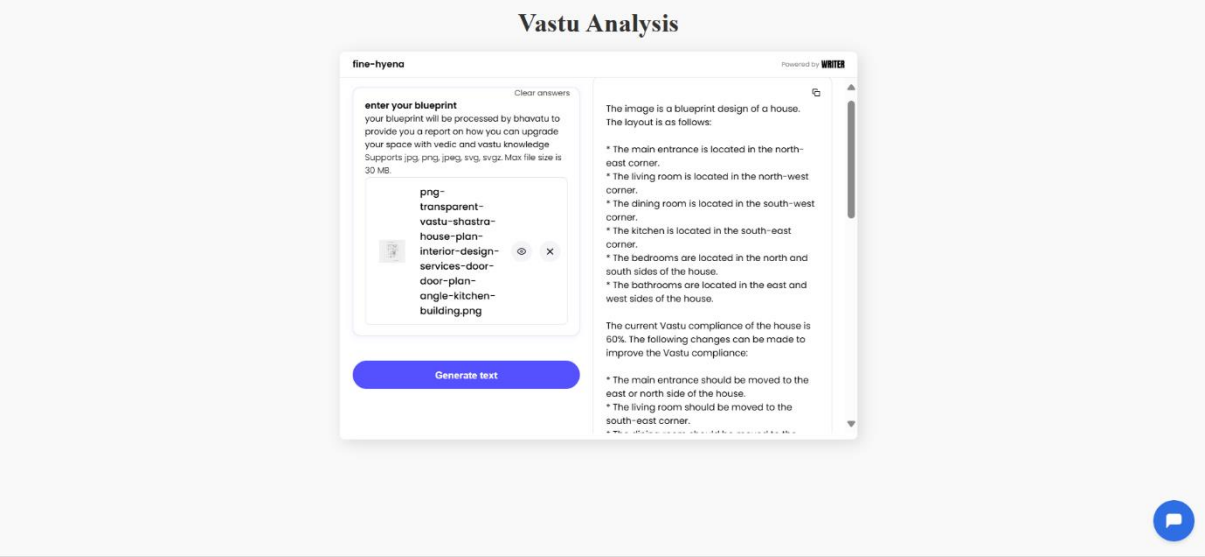
```
)  
}  
export default Footer;
```

RESULTS/OUTPUT

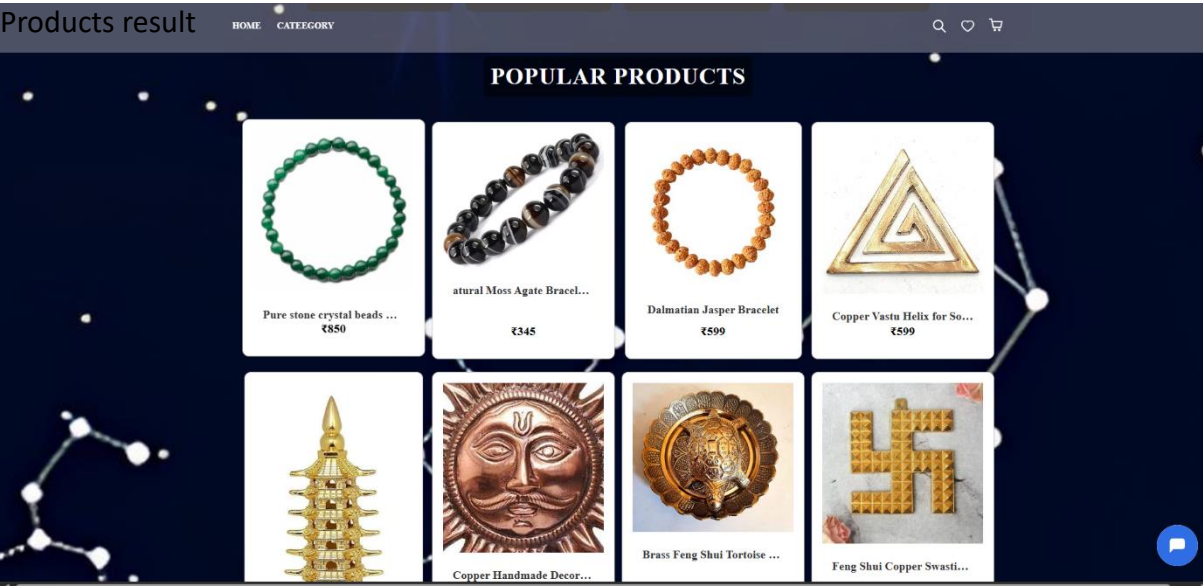
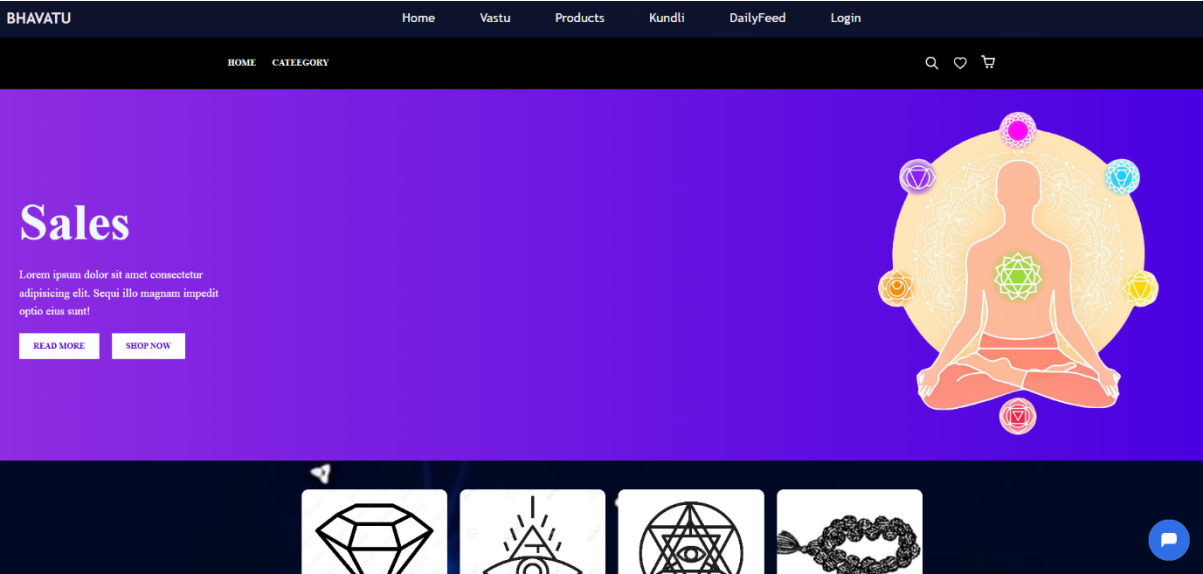
Home page result



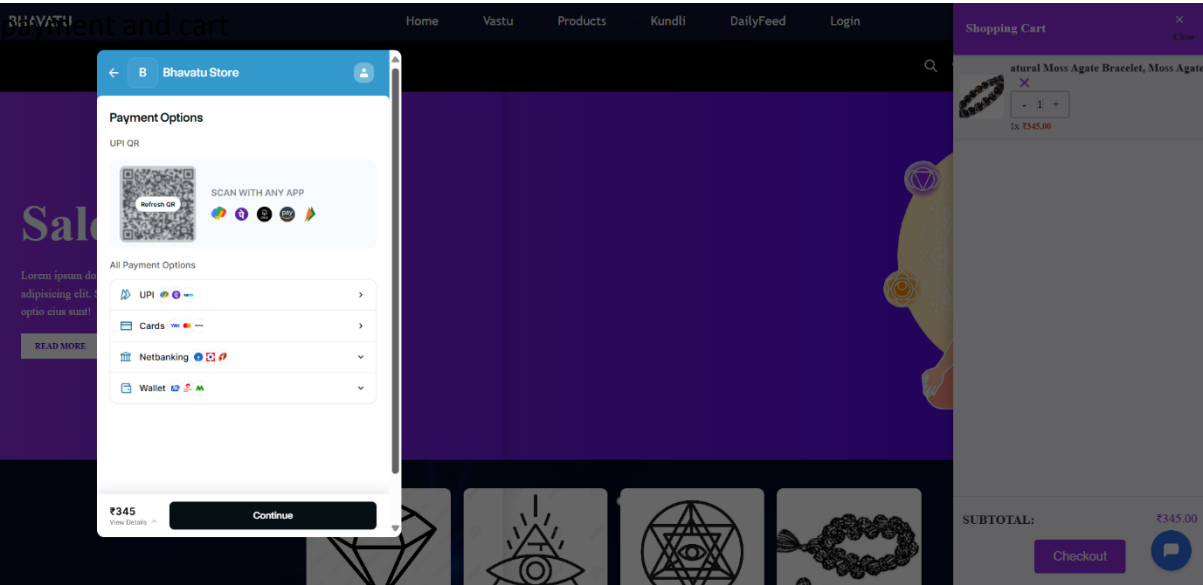
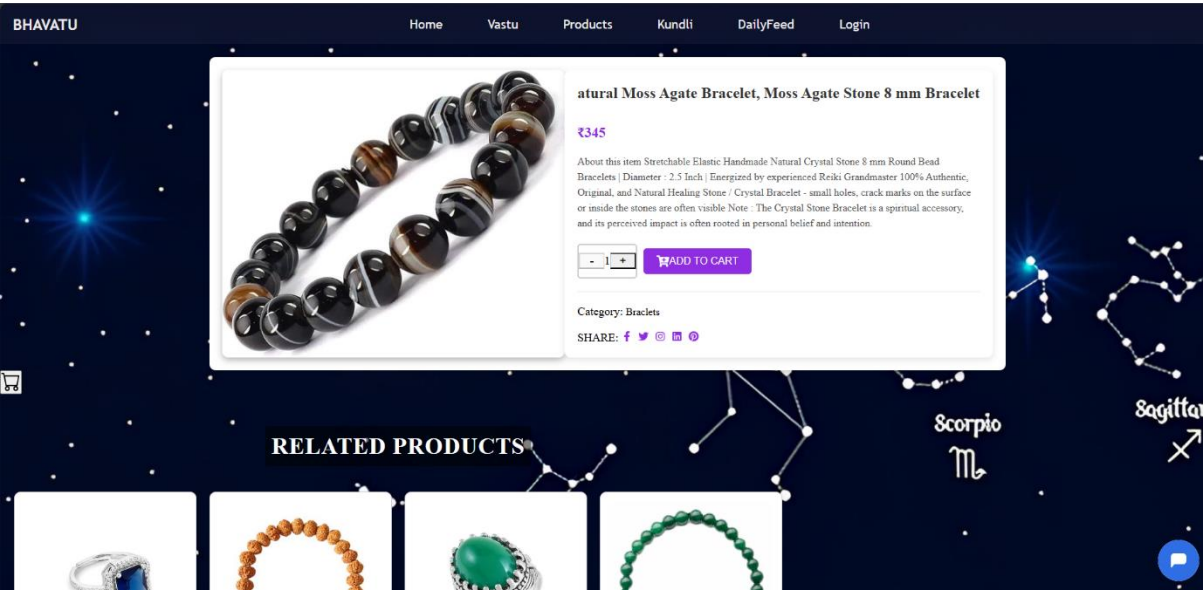
vastupage result



Ecommerce result

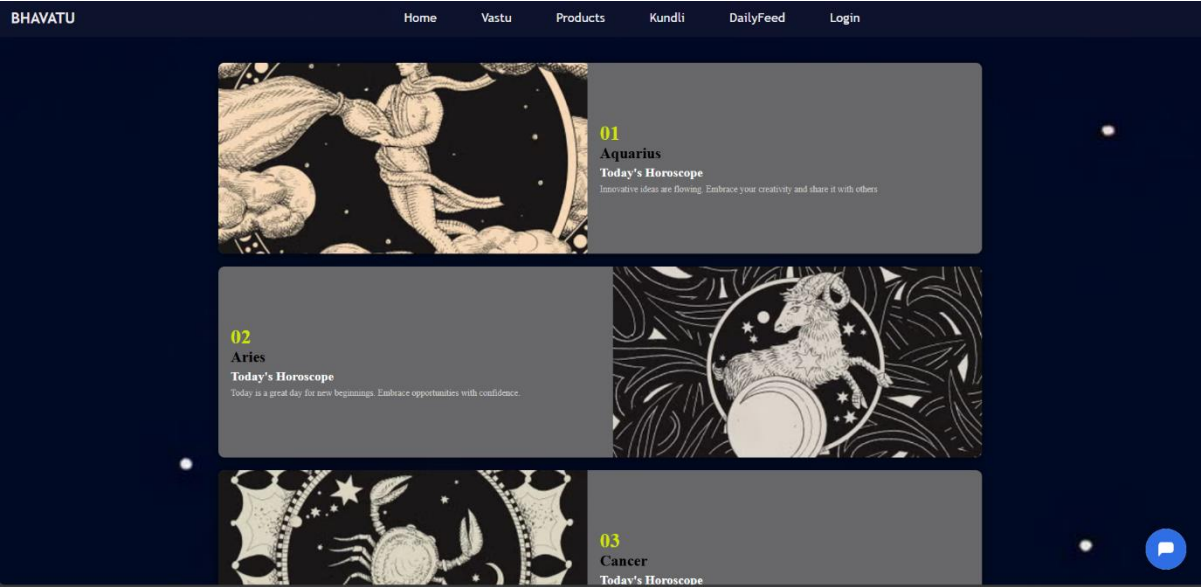


Singleproduct

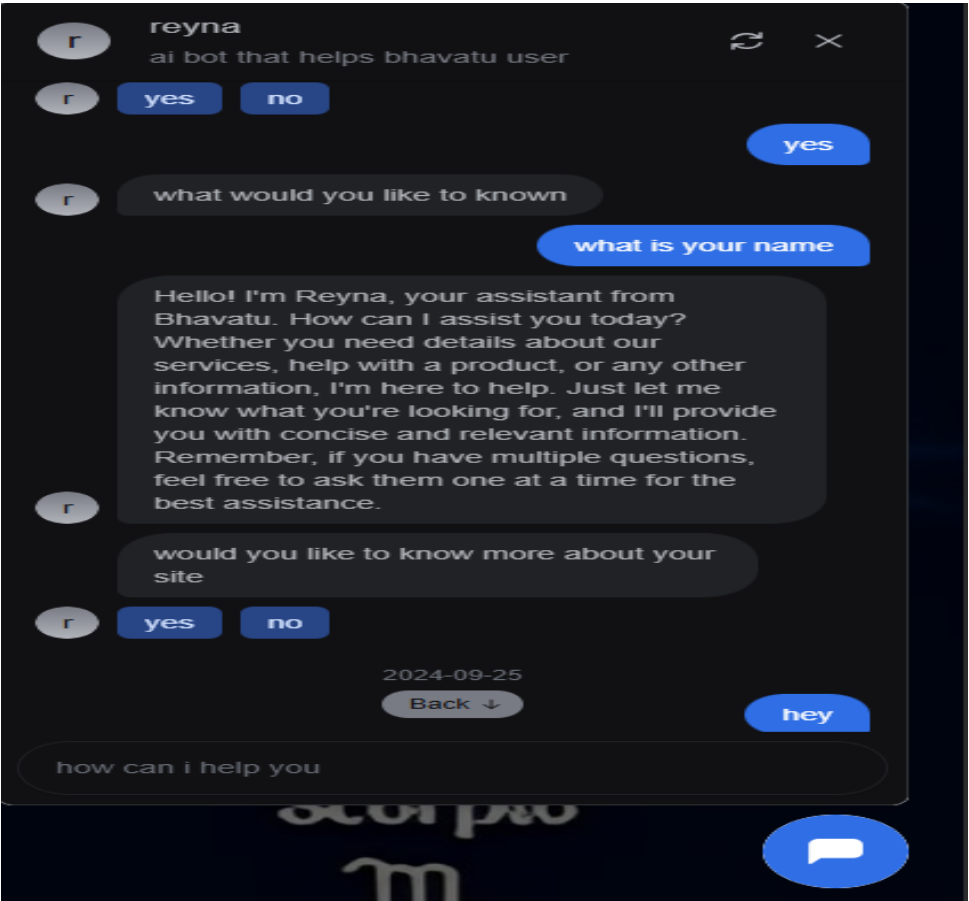


new

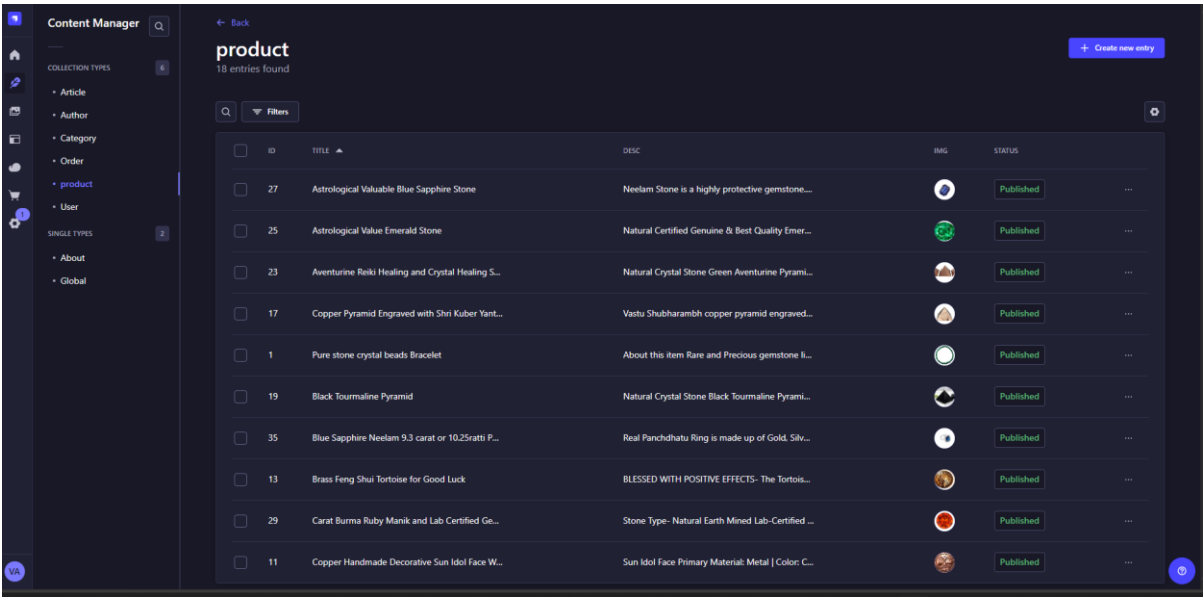
dailyfeed result



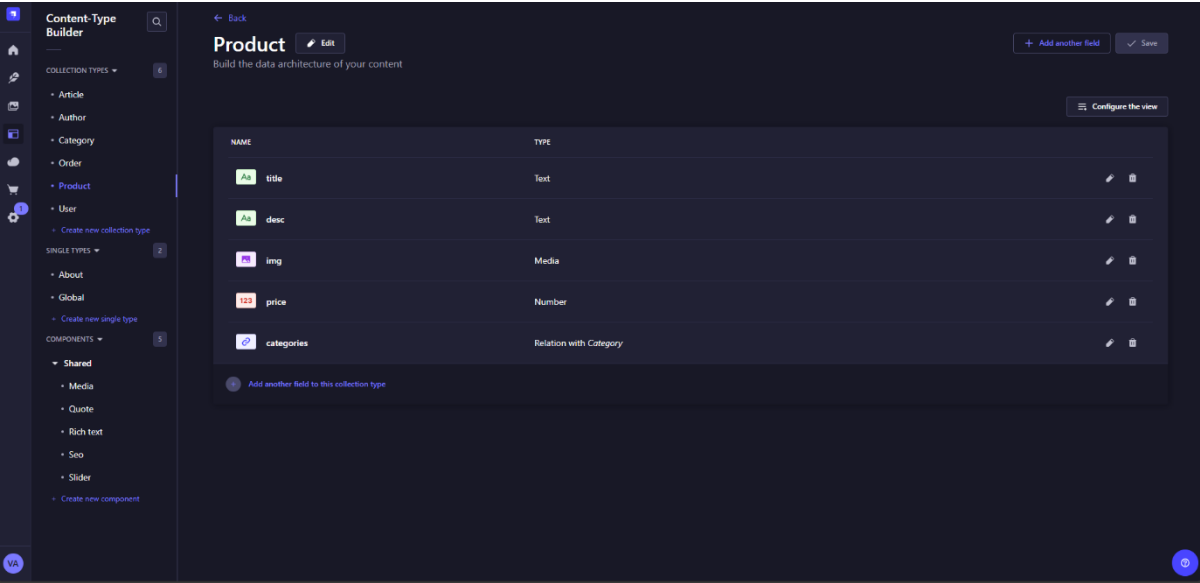
chatbot



Strapy cms

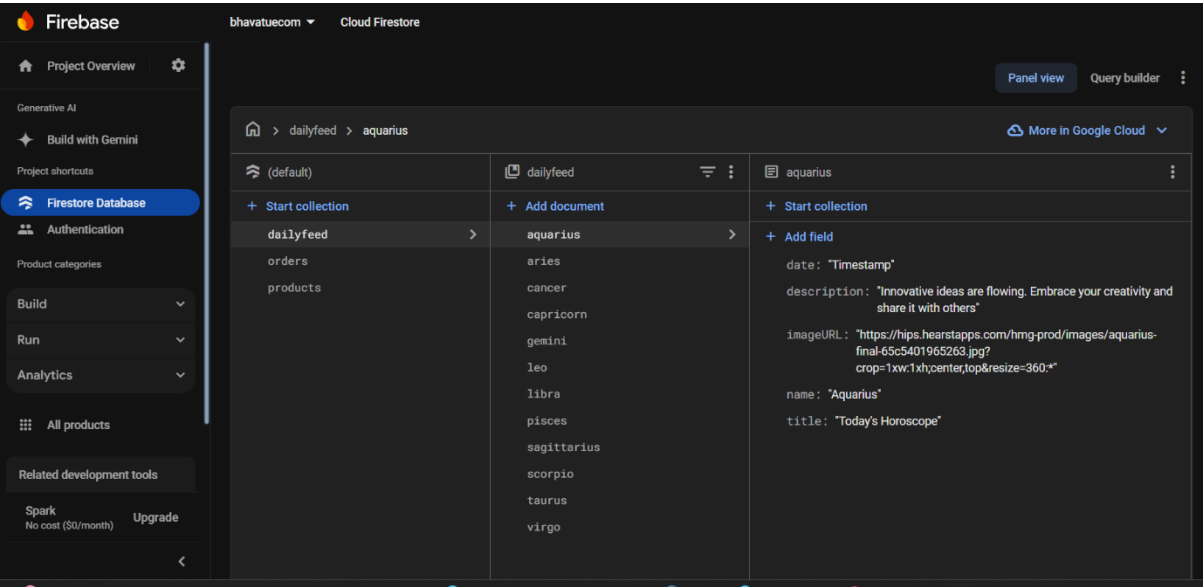


Add products

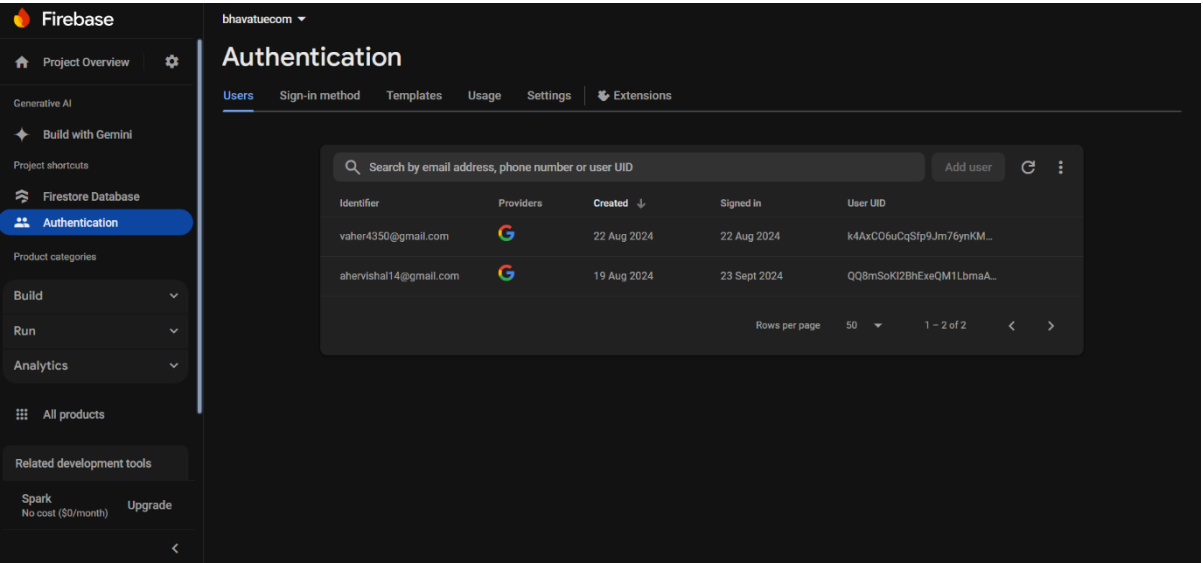


new

firebasedailyfeed



firebaseauth



new

FUTURE ENHANCEMENTS & CONCLUSION

6.1 Conclusion

The Bhavatu platform effectively integrates AI with traditional practices like Vastu and astrology, offering users a seamless experience. The addition of e-commerce features, secure payments, and AI-driven assistance makes it a comprehensive tool for anyone interested in these domains.

6.2 Future Enhancements

- **Expand Reyna's Capabilities:** Future iterations can improve Reyna's understanding of more complex Vastu and astrology queries.
- **Personalized Recommendations:** Adding a recommendation engine that tailors products based on user preferences and history.
- **Enhanced Vastu Analysis:** Using more sophisticated AI models to provide more detailed blueprint analysis.

REFERENCES

- **React.js Documentation:** <https://reactjs.org/>
- **Vite Documentation:** <https://vitejs.dev/>
- **Strapi CMS Documentation:** <https://strapi.io/>
- **OpenAI API Documentation:** <https://beta.openai.com/docs/>
- **Razorpay API Documentation:** <https://razorpay.com/docs/>