Project Report on

InfyTech – A Complete Tech Store

at

Webcreta Technologies

External Guide :

Mr. Anurag Vaishnav

Internal Guide :

Prof. Dhiren Prajapati

Prepared By :

Mr. Kaushal Vibhakar
(19012011091)

B.Tech Semester VIII
(Computer Engineering)
May 2023

Submitted to,
Department of Computer Engineering
U.V. Patel College of Engineering
Ganpat University, Kherva - 384 012

U.V. PATEL COLLEGE OF ENGINEERING

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering
CELEBRATING 25 YEARS OF EXCELLENCE 1997 - 2022
25 Years excellence in innovative technical education in shaping engineers

13/05/23

<u>CERTIFICATE</u>

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **<u>Mr. Kaushal Vibhakar</u>** student of **B.Tech. Semester VIII (Computer Engineering)** has completed his full semester on site project work titled "**<u>InfyTech</u>**" satisfactorily in partial fulfillment of the requirement of Bachelor of Technology degree of Computer Engineering of Ganpat University, Kherva, Mehsana in the year 2022-2023.

Prof. Dhiren Prajapati,	**Dr. Paresh M. Solanki,**
College Project Guide.	**HOD, Computer Engineering.**

WEBCRETA

Internship Completion Letter

Date - 03/05/2023

This is to certify that **Mr. Kaushal Vibhakar**, student of Ganpat University, has successfully completed his 4-Months training as **"React.js Developer"** commencing from January 02 2023 to May 02 2023, under the guidance of Mr. Anurag Vaishnav at Webcreta Technologies Pvt Ltd.

His Internship activates includes

- Web Development
- API Development
- Review of Project Requirements
- An Ecommerce Website Project

During the training period with us, he was exposed to various processes and was found diligent and inquisitive.

We wish him every success in his life and career

Jignesh Chauhan
CEO

WEBCRETA TECHNOLOGIES GANDHINAGAR

Webcreta Technologies

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contributed to this world’s development and influenced our thinking, behavior and acts during the course of study.

I would like to express my deep gratitude to my company project guide **Mr. Anurag Vaishnav** as well as my college project guide **Prof. Dhiren Prajapati**, Department of Computer Engineering, UVPCE and for their guidance with unsurpassed knowledge and immense encouragement.

I am grateful to **Dr. Paresh M. Solanki**, Head of the Department, of Computer Engineering, for providing me with the required facilities for the completion of the project work.

I am very much thankful to the **Principal and Management, of UVPCE, Ganpat University** for their encouragement and cooperation to carry out this work.

I express my thanks to all the teaching faculty of the Department of Computer Engineering, whose suggestions during reviews helped me in the accomplishment of our project.

I would like to thank my parents, friends, and classmates for their encouragement throughout our project period.

- **Kaushal Vibhakar [19012011091]**

ABSTRACT

Business in any field has a lot of competition. They always look out for a proven way to increase their business revenue. Restaurants, Retails Shops and many businesses want to increase their sales and since the world is going online in every aspect of life, if this kind of business doesn’t have an e-commerce website, they are leaving money on the table. The world has moved online - a fact that all businesses have to accept and put up a responsive website to address. Amazon and Flipkart are prime examples of websites with all the key elements making up a good e-commerce site. Almost all websites were initially put together with HTML, CSS and JavaScript. But as time progressed and different frameworks came into the picture, the website got a makeover. Through this project we’ll discover a way to build a functional e-commerce website, relying on React and its libraries.

My project is “InfyTech – A Complete Tech Store” which is a React Ecommerce Website. This website will be useful for everyone to find and buy all types of computer products on the Internet. It is useful in a way that it makes it an easier way to buy products online. This project is an interactive e-commerce solution providing users with an opportunity to buy their desired products easily. In this project, users can log in/signup and buy the products from the website and can find various types of deals and offer on the available products. The UI is created in such a way that any user can find it easy to access this website.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Project Introduction :

Online shopping or e-commerce is to buy products or services directly from the seller through the Internet. It is in trend as more people than before are using e-commerce to shop for a variety of items and daily needs. With this project, we will convert the retail business into a digital business. With the help of an e-commerce website, the shop is available for customers anytime. Customers don’t need to go outside to purchase or inquire about the product. Every possible product is available on the website.

Nowadays people have multiple options to choose their platform to buy and sell the products. Consumer attitudes towards online shopping are usually determined by two factors: one is trust and another is benefits. Therefore, trust and benefits seem to be the critical presumption of consumer behavior towards e-commerce platforms. Moreover, information quality, website design, transaction capability, payment, security/privacy, fast delivery, and customer service are strongly predictive of online shopping satisfaction. The e-commerce store which utilizes value-added mechanisms in the search engine and provides customers with a good experience may increase customers' shopping enjoyment.

1.2 Objective :

The central concept of the application is to allow customers to shop virtually using the Internet and allow customers to buy the items and articles they desire from the store. On our website, every type of product is available like electronics items. The website is designed into two modules: one in which customers can search and buy products and the other module is for admin where he can perform CRUD operations. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the products.

The following are the steps used for designing the website :

- The first task, once we get the development environment ready, will be to set up the React Router.

- Once we have everything in place, we can start with creating the website header which will serve as the navigation bar, as in most modern websites. Next up is the home page building. In this project, we’ll be keeping it simple by showing all our product categories on the homepage.

- Then we’ll be setting up the React Context API. The Context API is a component structure provided by the React framework, which enables us to share specific states across all levels of the application. In our project, we’ll need to manage two states: basket (to manage the shopping cart) and user (for managing the details of the currently logged user).

- For setting up the payments functionality, we’ll be using APIs provided by Stripe. Handling our database and authentication needs to be supported and we’ll be using Firebase for the same.

- We will use separate databases for storing login and product information.

- Once we have everything set up, we can work on the Login page of our application.

- Successful implementation of the above requirements will lead to the completion of the core implementation of our e-commerce solution. Next up, deploy!

1.3 Basic Workflow :

Register > Login > Home Page > Products > Cart > Checkout > Logout

1.4 Benefits :

This application will have no ads and no subscription requirements and so it’s completely free to use.

1.5 Project Duration :

16 Weeks.

2. Feasibility Study

2.1 Study of the Current System :

Many sites are available on the internet for online shopping. There are some more popular than others. Amazon and Flipkart are prime examples. They provide various functionalities like viewing all products, adding or removing product/s in the cart, managing personal profiles, ordering product/s, cancelling the order, making online payments, giving an option of offline payment, managing various delivery addresses, tracking the product, returning or replace the product, customer care supports, product feedbacks, and many other features. They use courier services to deliver the products. Multiple vendors are available on those sites. So we also call them multi-vendor sites. Also, they give facilities for the vendors to register on the site to sell their products.

2.2 Requirements for a New System :

- All the existing systems are non-responsive and full of ads.
- Most of the current systems are bloated with unnecessary features which confuse new customers and lead to a bad user experience.
- We might not be able to visit the vendor's physical location to resolve our issue/s.
- When we apply for the replacement of products, it will take a long time. It takes around 7 to 10 days.
- Only customers who have premium accounts can fast-order deliveries.

Whenever a new system (hardware or software) is to be introduced, there is a need to study the new system in every aspect or manner before working on it. It gives the idea of whether the project is adequate or not. This is where the Feasibility study comes in. The three key considerations that are involved in the feasibility analysis are as under :

2.3 Technical Feasibility :

It refers to technical know-how.

Existing technology supports the system completely. Hence our system is technically feasible.

2.4 Operational Feasibility :

It means that it's possible to practically implement the project. While installing this software, the hardware and software requirements should be specified.

Our system will be easy to install and use. Hence our system is operationally feasible.

2.5 Economical Feasibility :

It refers to the financial support required. Meaning, finance incurred during the development of the project.

Economical Feasibility is mainly concerned with the cost incurred in the implementation of the software. Since this project is developed using React with JavaScript and Firebase Server which is most commonly available and even the cost involved in the installation process is not high. Hence this project has good economical feasibility.

2.6 Hardware and Software Requirements :

Hardware Requirements :

- Processor : Intel Pentium 4 or More
- RAM : 1 GB or More
- Internet Connectivity is a must for this purpose.
- Basic Peripherals

Software Requirements :

- HTML, CSS and JavaScript
- React.js and Strapi
- SQLite.
- Firebase Database Servers
- Microsoft Visual Studio 2008

3. System Requirements Study

3.1 Functional Requirements :

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed andthe output expected. They are the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples : Register & Login.
: Verification Email is sent when the user registers for the first time.
: OTP is sent when the user registers with a mobile number.
: Login via Google Account.

3.2 Non-functional Requirements :

These are the quality constraints that the system must satisfy according to the project contract.The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements.

They deal with issues like : Portability
: Security
: Maintainability
: Reliability
: Scalability
: Performance
: Reusability

Examples : Each user must use different Emails.
: Password should contain Alphabets as well as Numbers.

4. Project Planning

4.1 Gantt Chart :

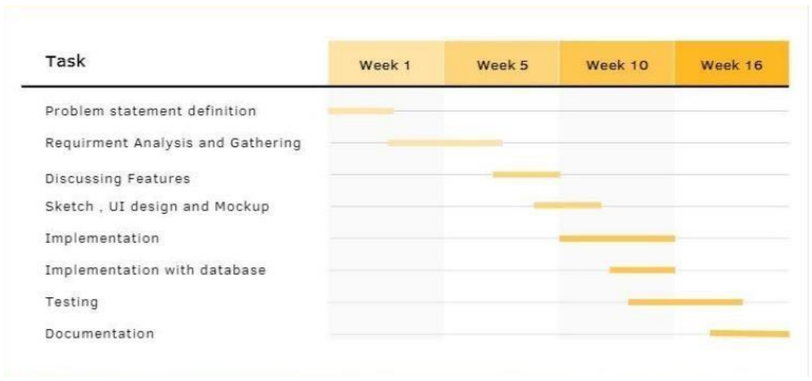


Figure 4.1 – Gantt Chart

4.2 Flowchart :

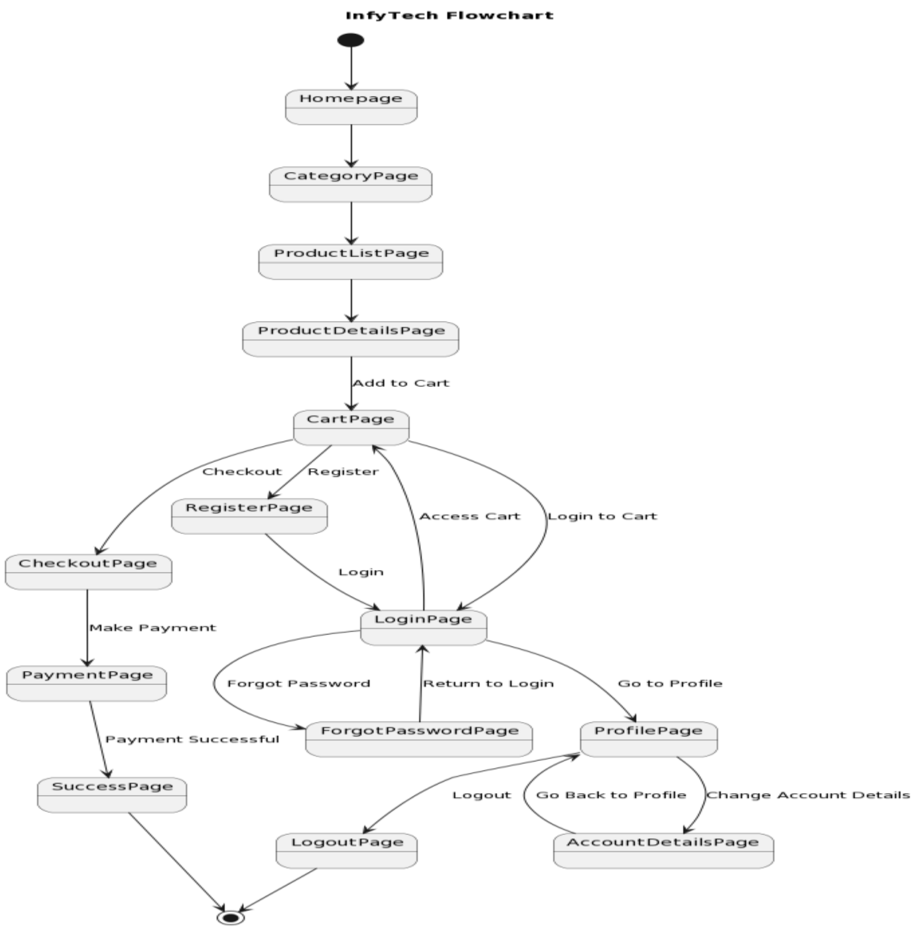


Figure 4.2 – Flowchart

5. System Design

5.1 Use Case Diagram :

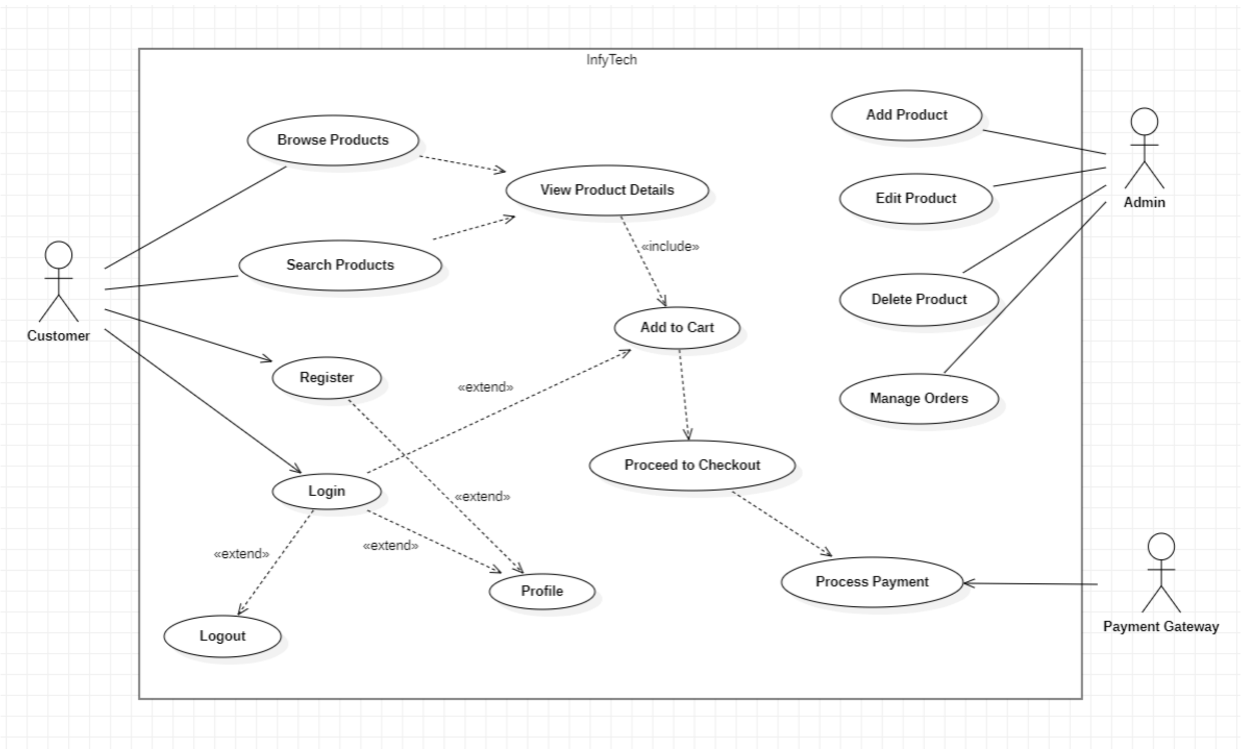


Figure 5.1 – Use Case Diagram

5.2 Class Diagram :

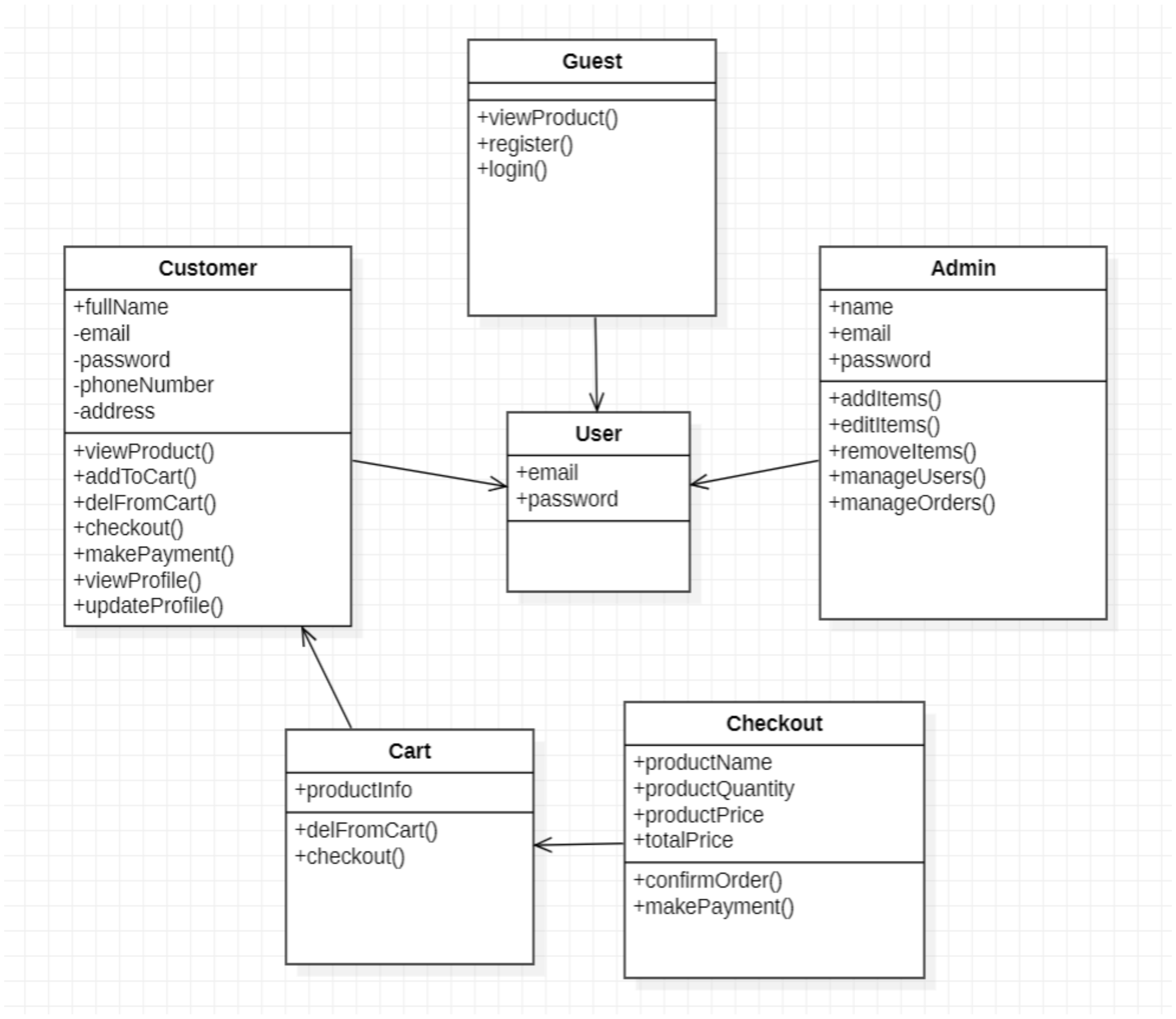


Figure 5.2 – Class Diagram

5.3 Sequence Diagram :

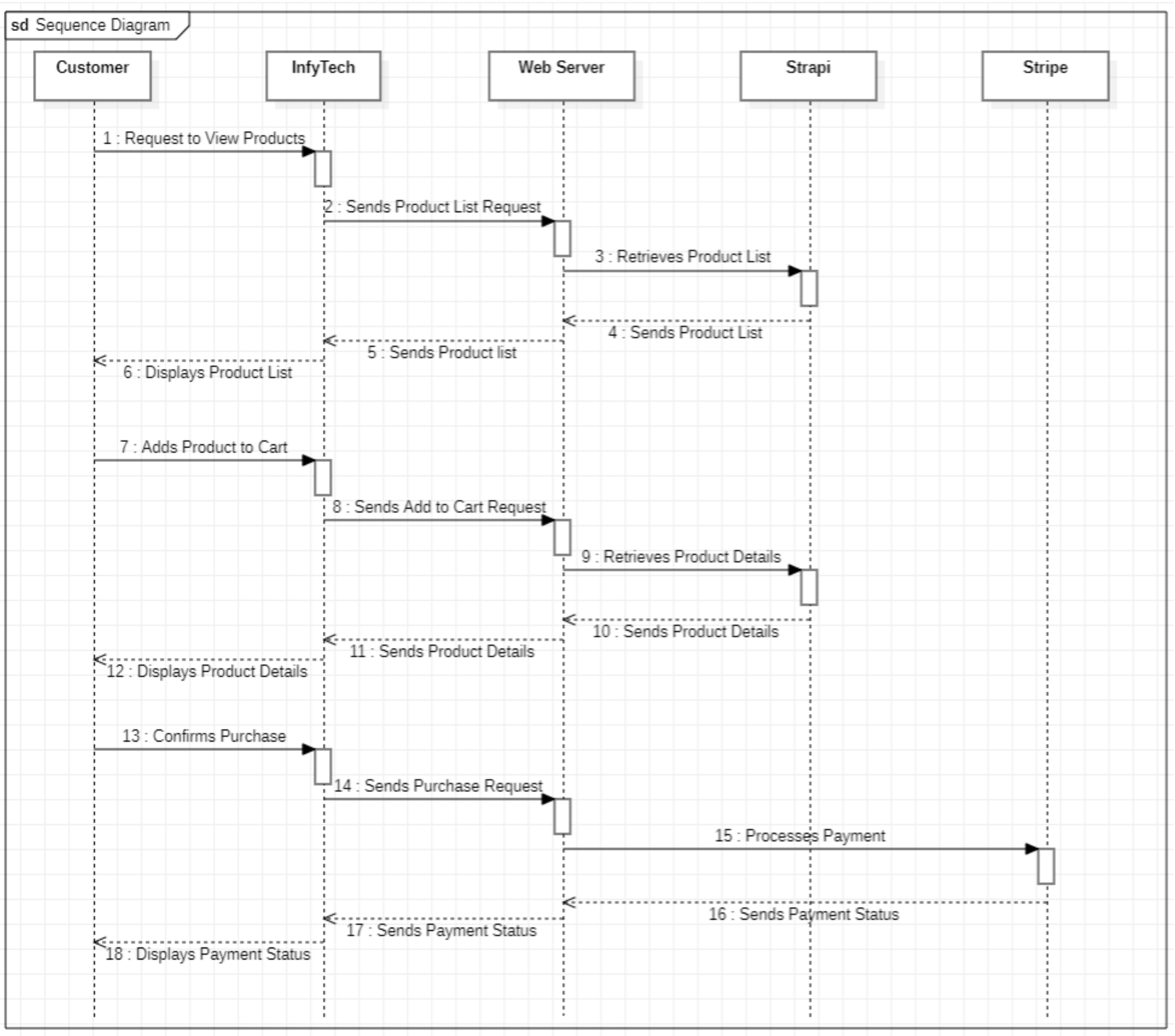

sd Sequence Diagram
Customer
InfyTech
Web Server
Strapi
Stripe
1 : Request to View Products
2 : Sends Product List Request
3 : Retrieves Product List
4 : Sends Product List
5 : Sends Product list
6 : Displays Product List
7 : Adds Product to Cart
8 : Sends Add to Cart Request
9 : Retrieves Product Details
10 : Sends Product Details
11 : Sends Product Details
12 : Displays Product Details
13 : Confirms Purchase
14 : Sends Purchase Request
15 : Processes Payment
16 : Sends Payment Status
17 : Sends Payment Status
18 : Displays Payment Status

Figure 5.3 – Sequence Diagram

5.4 Activity Diagram :

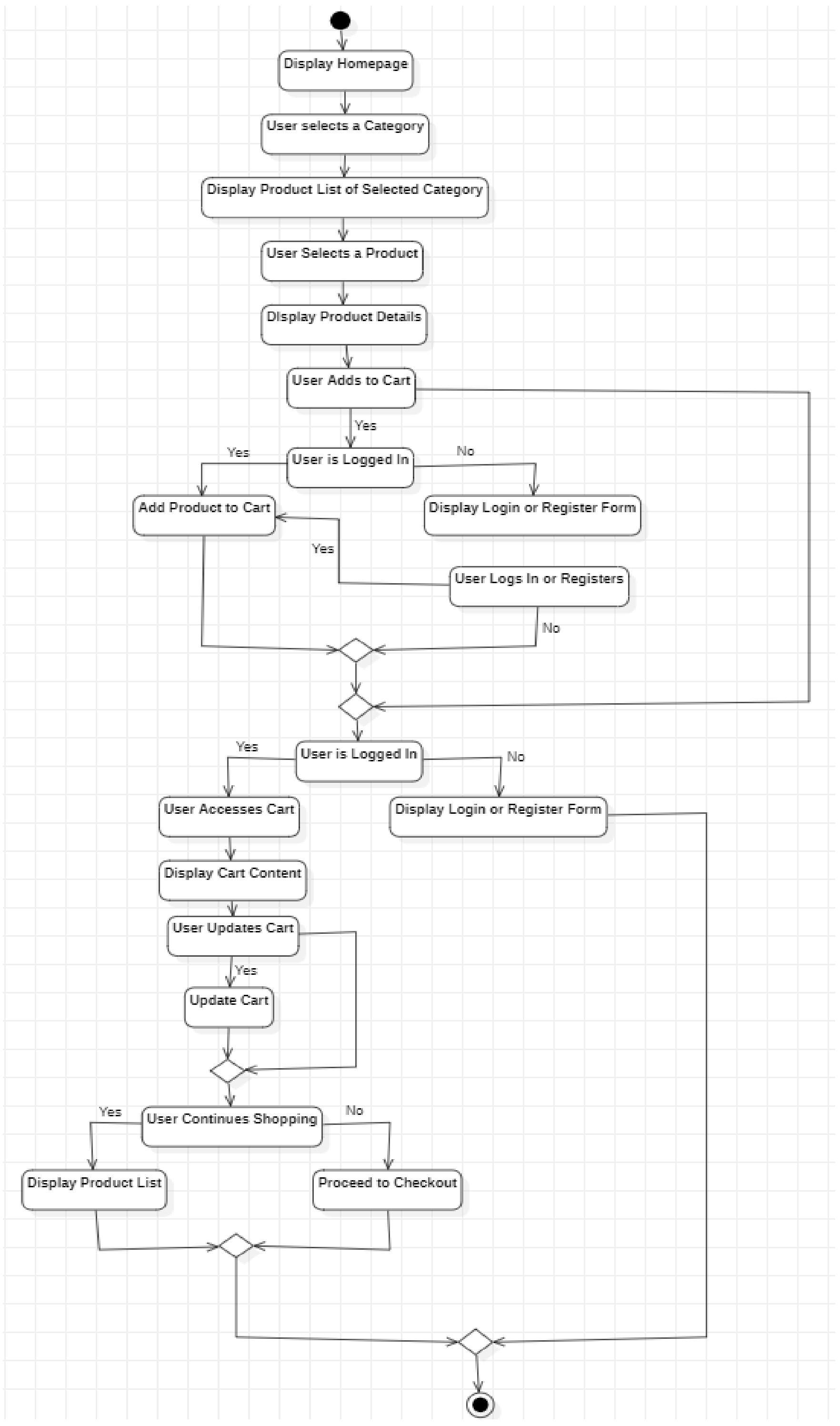


Figure 5.4 – Activity Diagram

5.5 Data Flow Diagrams :

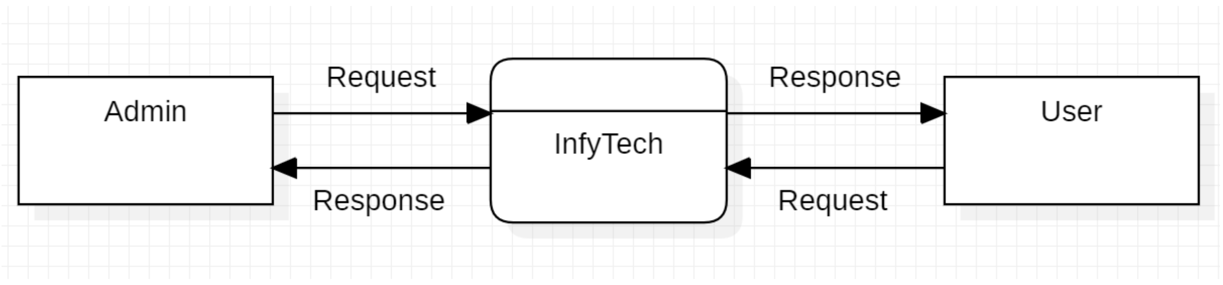


Figure 5.5 – Zero Level DFD

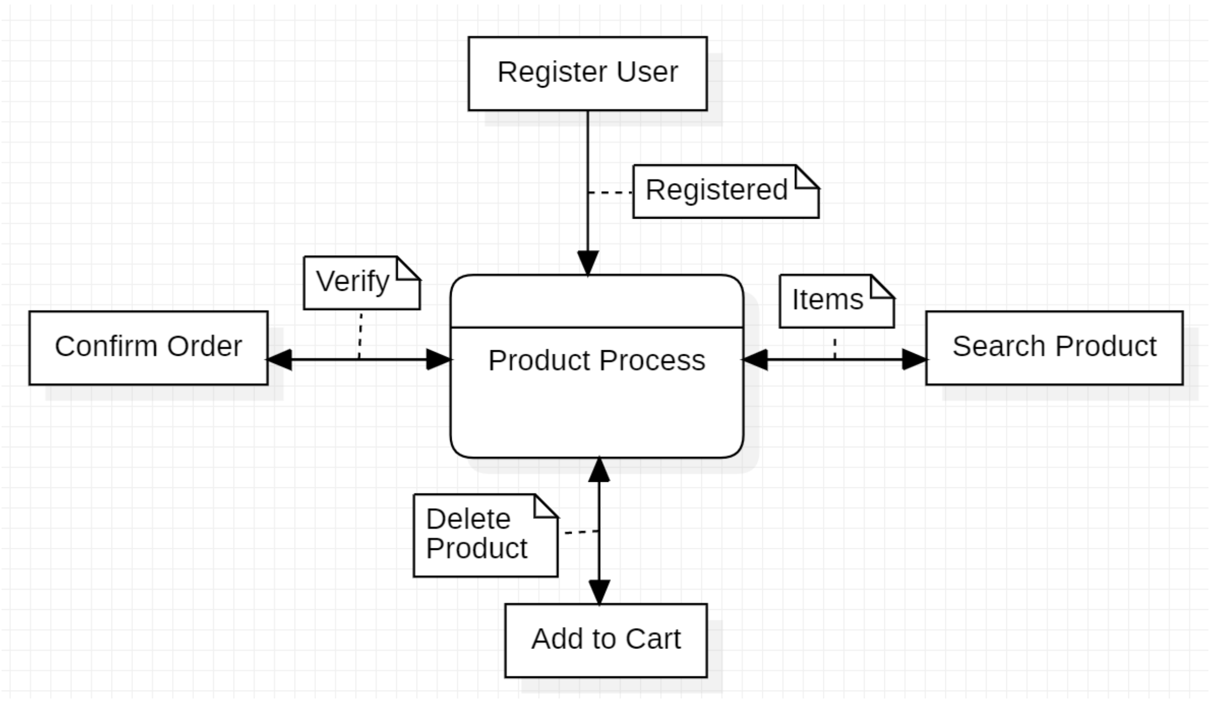


Figure 5.6 – First Level DFD

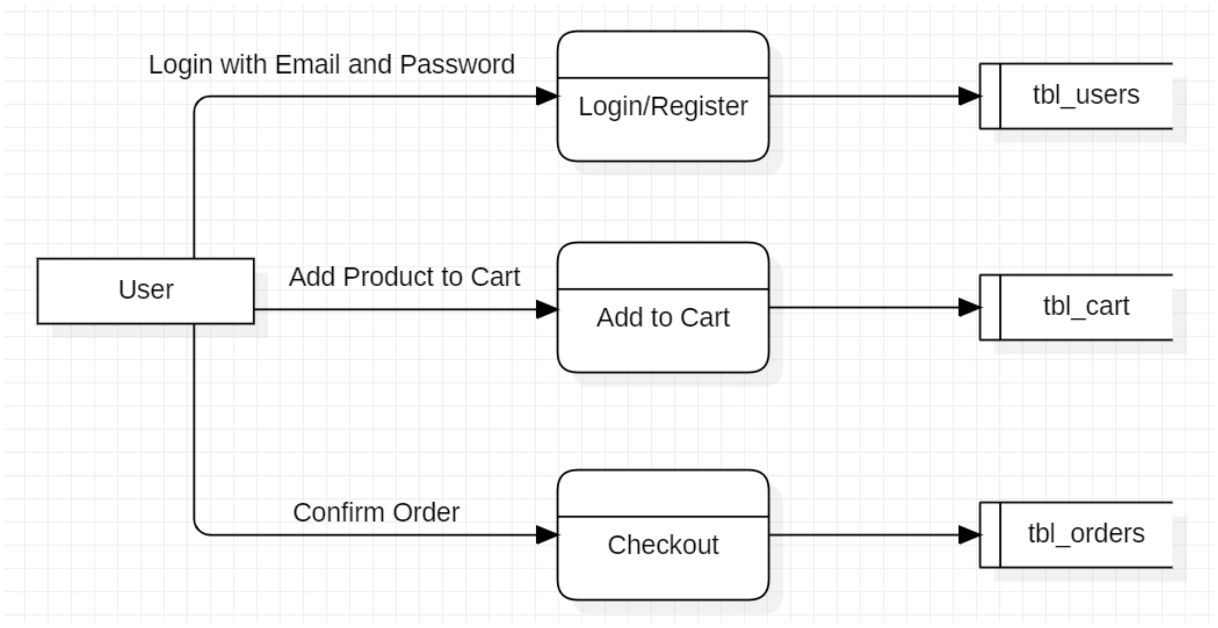


Figure 5.7 – Second Level DFD

6. Data Dictionary

6.1 Category Data Table :

Field Name	Data Type	Constraint	Description
Name	Varchar	Not Null	Category’s Title
filterType	Varchar	Not Null	Type of Filters in the Category

Table 6.1 – Category Data Table

6.2 Product Data Table :

Field Name	Data Type	Constraint	Description
Name	Varchar	Not Null	Product’s Name
Company	Varchar	Not Null	Product’s Company Name
Price	Number	Not Null	Product’s Price Value
Desc	Varchar	Not Null	Description of the Product
Img1	Media	Not Null	First Image of the Product
Img2	Media	None	Second Image of the Product

Table 6.2 – Product Data Table

6.3 Customer Data Table :

Field Name	Data Type	Constraint	Description
Fname	Varchar	Not Null	Customer’s First Name
Lname	Varchar	Not Null	Customer’s Last Name
Email	Varchar	Not Null	Email for Login Procedure
Pass	Varchar	Not Null > 6	Password for Login Procedure
Phone	Number	Not Null	Phone Number for Shipping and Billing Details
AddL1	Varchar	Not Null	Address Line 1 for Shipping and Billing Details
AddL2	Varchar	None	Address Line 2 for Shipping and Billing Details
City	Varchar	Not Null	City Name for Shipping and Billing Details
State	Varchar	Not Null	State Name for Shipping and Billing Details
Pincode	Number	Not Null <= 6	Pincode Number for Shipping and Billing Details

Table 6.3 – Customer Data Table

6.4 Order Data Table :

Field Name	Data Type	Constraint	Description
Stripe Id	Varchar	Not Null	Stripe Id of the Placed Order
Products	JSON	Not Null	Details of the Ordered Products

Table 6.4 – Order Data Table

7. Prototype

UI Design in Figma :

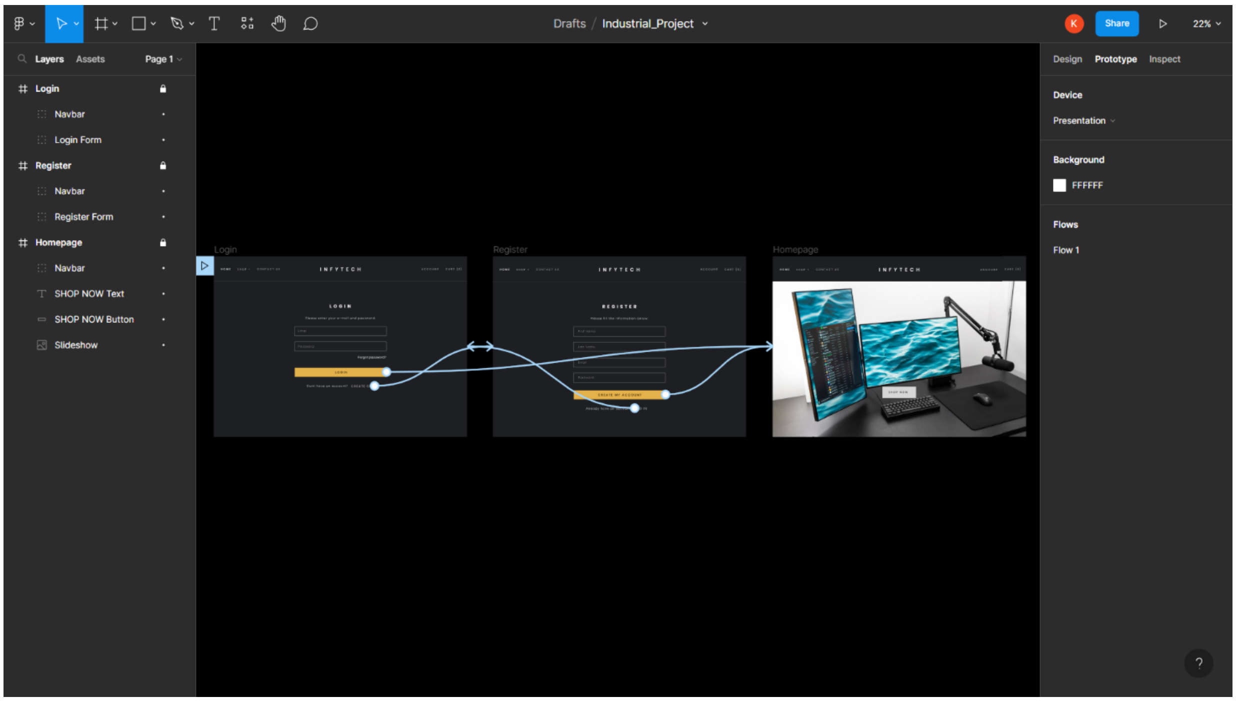
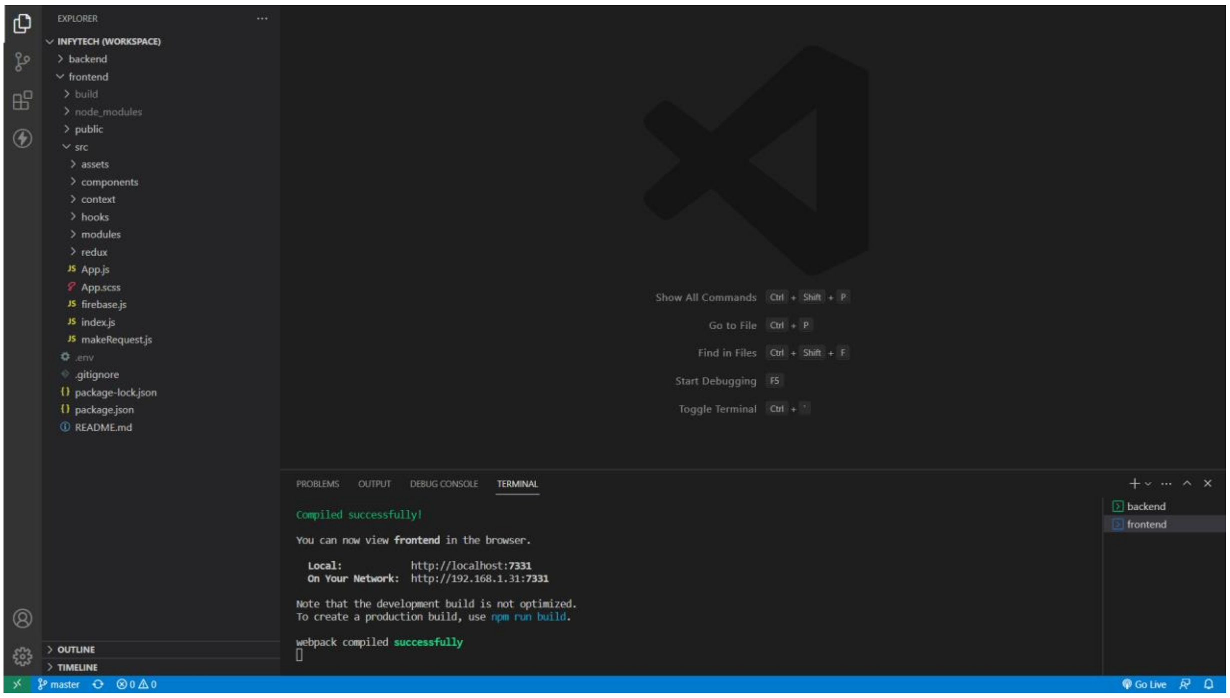


Figure 7.0 – UI Design in Figma

8. Implementation

Folder Structure :



```
Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:7331
  On Your Network:  http://192.168.1.31:7331

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Figure 8.0 – Folder Structure

8.1 Coding :

index.js :

```
import React from "react";
import ReactDOM from "react-dom/client";
import { ToastContainer } from "react-toastify";
import { Provider } from "react-redux";
import { persistor, store } from "./redux/store";
import { PersistGate } from "redux-persist/integration/react";
import App from "./App";
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
   <Provider store={store}>
      <PersistGate loading={"loading"} persistor={persistor}>
        <App />
        <ToastContainer />
      </PersistGate>
   </Provider>
);
```

App.js :

```
import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-
dom";
import ScrollToTop from "./components/ScrollToTop";
import { AuthContextProvider } from "./context/AuthContext";
import ProtectedRoute from "./components/ProtectedRoute";
import AnotherProtectedRoute from "./components/AnotherProtectedRoute";
import Home from "./modules/Home/Home";
import AllProducts from "./modules/Products/AllProducts";
import Products from "./modules/Products/Products";
import ProductDetails from "./modules/ProductDetails/ProductDetails";
import Register from "./modules/Account/Register";
import Login from "./modules/Account/Login";
import ResetPassword from "./modules/Account/ResetPassword";
import Profile from "./modules/Account/Profile";
import About from "./modules/About/About";
import Contact from "./modules/Contact/Contact";
import TermsNConditions from "./modules/Policies/TermsNConditions";
import PrivacyPolicy from "./modules/Policies/PrivacyPolicy";
import ReturnPolicy from "./modules/Policies/ReturnPolicy";
import Error from "./modules/Error/Error";
import "./App.scss";
const App = () => {
  return (
    <Router>
      <ScrollToTop />
      <AuthContextProvider>
```

```
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/products/all" element={<AllProducts />} />
          <Route path="/products/:id" element={<Products />} />
          <Route path="/product/:id" element={<ProductDetails />} />
          <Route
            path="/register"
            element={
              <AnotherProtectedRoute>
                <Register />
              </AnotherProtectedRoute>
            }
          />
          <Route
            path="/login"
            element={
              <AnotherProtectedRoute>
                <Login />
              </AnotherProtectedRoute>
            }
          />
          <Route
            path="/reset-password"
            element={
              <AnotherProtectedRoute>
                <ResetPassword />
              </AnotherProtectedRoute>
            }
          />
          <Route
            path="/profile"
            element={
              <ProtectedRoute>
                <Profile />
              </ProtectedRoute>
            }
          />
          <Route path="/about-us" element={<About />} />
          <Route path="/contact-us" element={<Contact />} />
          <Route path="/terms-and-conditions"
element={<TermsNConditions />} />
          <Route path="/privacy-policy" element={<PrivacyPolicy />} />
          <Route path="/return-policy" element={<ReturnPolicy />} />
          <Route path="*" element={<Error />} />
        </Routes>
      </AuthContextProvider>
    </Router>
  );
};
export default App;
```

ProtectedRoute.js :

```
import React from "react";
import { Navigate } from "react-router-dom";
import { UserAuth } from "../context/AuthContext";
const ProtectedRoute = ({ children }) => {
  const { user } = UserAuth();
  if (!user) {
    return <Navigate to="/login" />;
  }
  return children;
};
export default ProtectedRoute;
```

Firebase.js :

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
import { getFirestore } from "firebase/firestore";
const firebaseConfig = {
  apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
  authDomain: "infytech1337.firebaseapp.com",
  projectId: "infytech1337",
  storageBucket: "infytech1337.appspot.com",
  messagingSenderId: "307116764783",
  appId: "1:307116764783:web:e868592e1d99da4161e79a",
};
const app = initializeApp(firebaseConfig);
export const auth = getAuth(app);
export const db = getFirestore(app);
export default app;
```

AuthContext.js :

```
import { createContext, useContext, useEffect, useState } from "react";
import {
  createUserWithEmailAndPassword,
  signInWithEmailAndPassword,
  signOut,
  sendPasswordResetEmail,
  onAuthStateChanged,
} from "firebase/auth";
import { auth } from "../firebase";
const UserContext = createContext();
export const AuthContextProvider = ({ children }) => {
  const [user, setUser] = useState({});
  const createUser = (email, password) => {
    return createUserWithEmailAndPassword(auth, email, password);
  };
  const signIn = (email, password) => {
    return signInWithEmailAndPassword(auth, email, password);
  };
  const resetPassword = (email) => {
    return sendPasswordResetEmail(auth, email);
  };
  const logout = () => {
    return signOut(auth);
  };
  useEffect(() => {
    const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
      setUser(currentUser);
    });
    return () => unsubscribe();
  }, []);
  return (
    <UserContext.Provider
      value={{ user, createUser, signIn, resetPassword, logout }}
    >
      {children}
    </UserContext.Provider>
  );
};
export const UserAuth = () => {
  return useContext(UserContext);
};
```

useFetchUser.js :

```
import { useEffect, useState } from "react";
import { doc, getDoc } from "firebase/firestore";
import { db } from "../firebase";
const useFetchUser = (uid) => {
  const [userData, setUserData] = useState(null);
  useEffect(() => {
    const fetchUserData = async () => {
      try {
        if (uid) {
          const docRef = doc(db, "users", uid);
          const docSnap = await getDoc(docRef);
          setUserData(docSnap.data());
        }
      } catch (error) {
        console.log(error.message);
      }
    };
    fetchUserData();
  }, [uid]);
  return userData;
};
export default useFetchUser;
```

MakeRequest.js :

```
import axios from "axios";
export const makeRequest = axios.create({
  baseURL: process.env.REACT_APP_API_URL,
  headers: {
    Authorization: "bearer " + process.env.REACT_APP_API_TOKEN,
  },
});
```

useFetchData.js :

```
import { useEffect, useState } from "react";
import { makeRequest } from "../makeRequest";
const useFetchData = (url) => {
  const [data, setData] = useState([]);
  const [filterData, setFilterData] = useState([]);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(false);
  useEffect(() => {
    const fetchData = async () => {
      try {
        setLoading(true);
        const res = await makeRequest.get(url);
        setData(res.data.data);
        setFilterData(res.data.data);
      } catch (err) {
        setError(true);
      }
      setLoading(false);
    };
    fetchData();
  }, [url]);
  return { data, filterData, loading, error };
};
export default useFetchData;
```

FilteredProductList.js :

```
import React from "react";
import useFetchData from "../../hooks/useFetchData";
import ProductCard from "../ProductCard/ProductCard";
import { ClipLoader } from "react-spinners";
import "./ProductList.scss";
const FilteredProductList = ({
  categoryId,
  filters,
  maxPrice,
  sort,
  searchQuery,
}) => {
  const { data, loading, error } = useFetchData(
    `/products?populate=*&[filters][categories][id]=${categoryId}${filt
ers.map(
      (item) => `&[filters][sub_categories][id]=${item}&`
    )}&[filters][price][$lte]=${maxPrice}&sort=price:${sort}&[filters][
$or][0][name][$containsi]=${searchQuery}&[filters][$or][1][company][$co
ntainsi]=${searchQuery}`
  );
  return (
         {/* Filtered List Component's JSX Code. */}
  );
};
export default FilteredProductList;
```

ProductDetails.js :

```
import React, { useState } from "react";
import { NavLink, useNavigate, useParams } from "react-router-dom";
import { useDispatch } from "react-redux";
import useFetchData from "../../hooks/useFetchData";
import { addToCart } from "../../redux/cartReducer";
import Navbar from "../../components/Navbar/Navbar";
import RelatedProductList from
"../../components/ProductList/RelatedProductList";
import Footer from "../../components/Footer/Footer";
import { UserAuth } from "../../context/AuthContext";
import { toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import "./ProductDetails.scss";
import Cart from "../../components/Cart/Cart";
const ProductDetails = () => {
  const productId = useParams().id;
  const [selectedImg, setSelectedImg] = useState("img1");
  const [quantity, setQuantity] = useState(1);
  const { data, loading, error } = useFetchData(
    `/products/${productId}?populate=*`
  );
  const dispatch = useDispatch();
  const rel = data?.attributes?.categories?.data[0]?.id;
  const navigate = useNavigate();
  const [openCart, setOpenCart] = useState(false);
  const closeCart = () => setOpenCart(false);
  const notifyA = () =>
    toast.success("ADDED TO THE CART!", {
      className: "toastify",
      position: "bottom-right",
      autoClose: 3000,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      progress: undefined,
      theme: "dark",
      onClick: () => {
        setOpenCart(true);
      },
    });
  const notifyB = () =>
    toast.warning("PLEASE LOGIN TO USE CART!", {
      className: "toastify",
      position: "bottom-right",
      autoClose: 3000,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      progress: undefined,
      theme: "dark",
      onClick: () => {
        navigate("/login");
      },
    });
  const { user } = UserAuth();
  return (
         {/* Product Details Page's JSX Code. */}
  );
};
export default ProductDetails;
```

CartReducer.js :

```
import { createSlice } from "@reduxjs/toolkit";
const initialState = {
  products: [],
};
export const cartSlice = createSlice({
  name: "cart",
  initialState,
  reducers: {
    addToCart: (state, action) => {
      const item = state.products.find((item) => item.id ===
action.payload.id);
      if (item) {
        item.quantity += action.payload.quantity;
      } else {
        state.products.push(action.payload);
      }
    },
    removeFromCart: (state, action) => {
      state.products = state.products.filter(
        (item) => item.id !== action.payload
      );
    },
    resetCart: (state) => {
      state.products = [];
    },
  },
});
export const { addToCart, removeFromCart, resetCart } =
cartSlice.actions;
export default cartSlice.reducer;
```

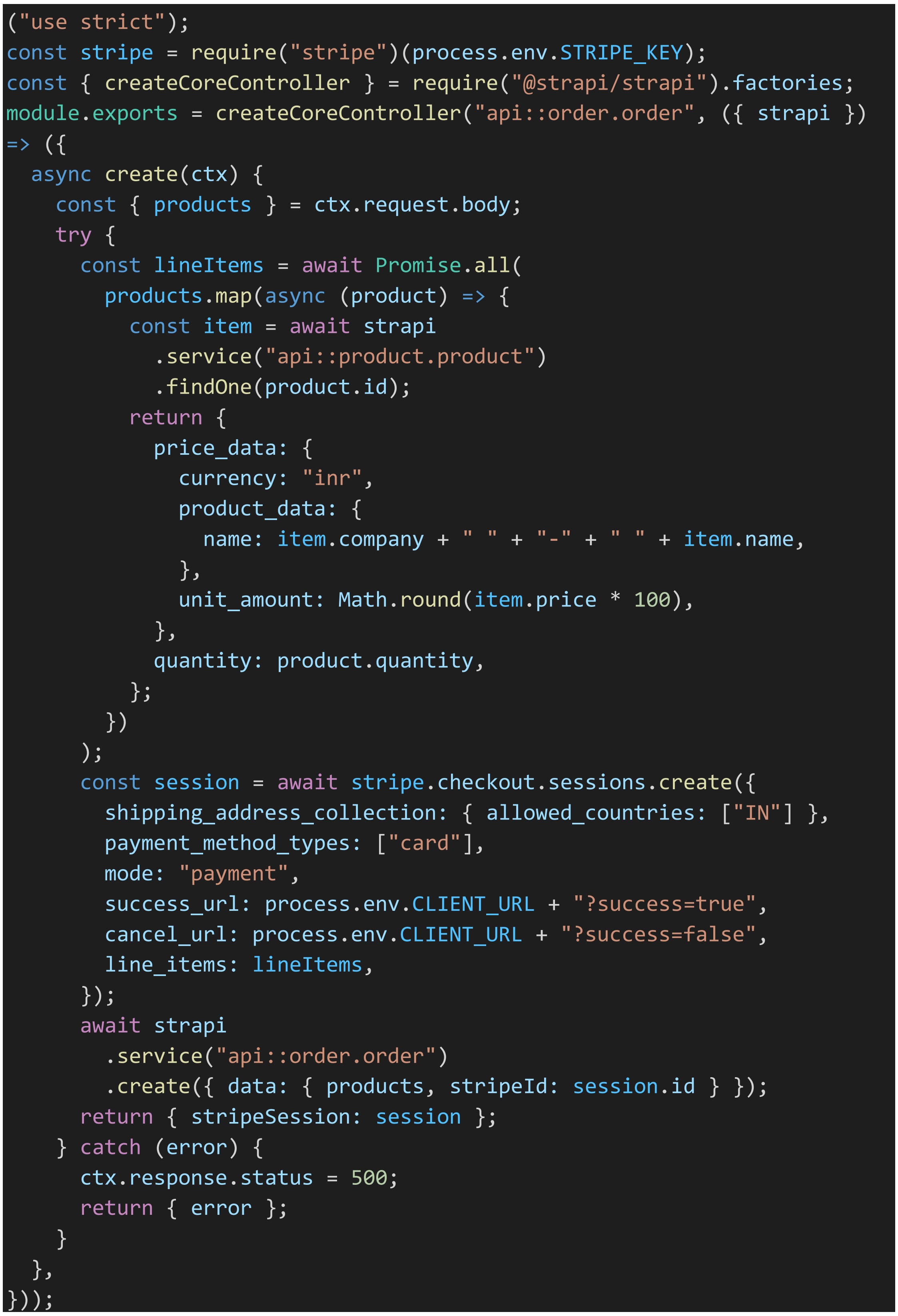
Store.js :

```
import { configureStore } from "@reduxjs/toolkit";
import cartReducer from "./cartReducer";
import {
  persistStore,
  persistReducer,
  FLUSH,
  REHYDRATE,
  PAUSE,
  PERSIST,
  PURGE,
  REGISTER,
} from "redux-persist";
import storage from "redux-persist/lib/storage";
const persistConfig = {
  key: "root",
  version: 1,
  storage,
};
const persistedReducer = persistReducer(persistConfig, cartReducer);
export const store = configureStore({
  reducer: {
    cart: persistedReducer,
  },
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({
      serializableCheck: {
        ignoredActions: [FLUSH, REHYDRATE, PAUSE, PERSIST, PURGE,
REGISTER],
      },
    }),
});
export let persistor = persistStore(store);
```

Cart.js :

```
import React, { useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { removeFromCart } from "../../redux/cartReducer";
import { loadStripe } from "@stripe/stripe-js";
import { makeRequest } from "../../makeRequest";
import "./Cart.scss";
const Cart = ({ closeCart }) => {
  useEffect(() => {
    document.body.style.overflowY = "hidden";
    return () => {
      document.body.style.overflowY = "scroll";
    };
  }, []);
  const products = useSelector((state) => state.cart.products);
  const dispatch = useDispatch();
  const totalPrice = () => {
    let total = 0;
    products.forEach((item) => {
      total += item.quantity * item.price;
    });
    return total.toFixed(2);
  };
  const stripePromise = loadStripe(
    "pk_test_51MoLLYSFvhYDcU3mBTG1lC4WewLICocYpuD0sk68BDMmkZEwbuq4zUYSo
g2mW2rqVu37dAOrnWW0NtvMUCNt6tDx00ySYApFd7"
  );
  const handlePayment = async () => {
    try {
      const stripe = await stripePromise;
      const res = await makeRequest.post("/orders", {
        products,
      });
      await stripe.redirectToCheckout({
        sessionId: res.data.stripeSession.id,
      });
    } catch (err) {
      console.log(err.message);
    }
  };
  return (
         {/* Cart Component’s JSX Code. */}
  );
};
export default Cart;
```

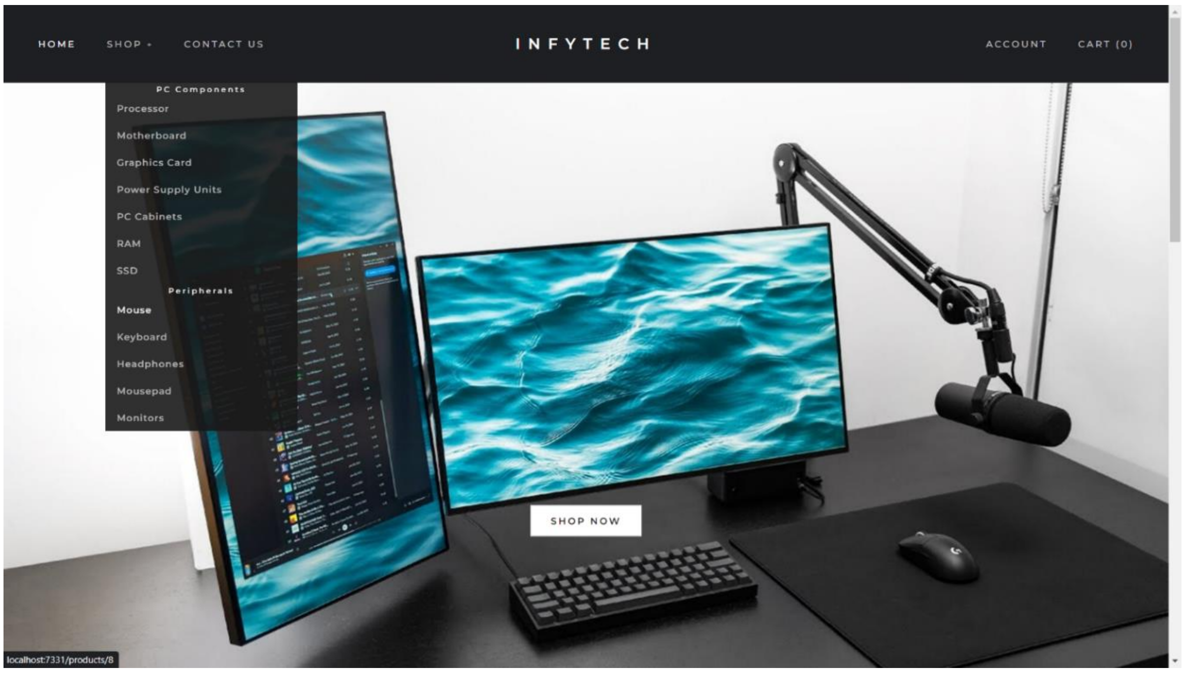
Order.js :



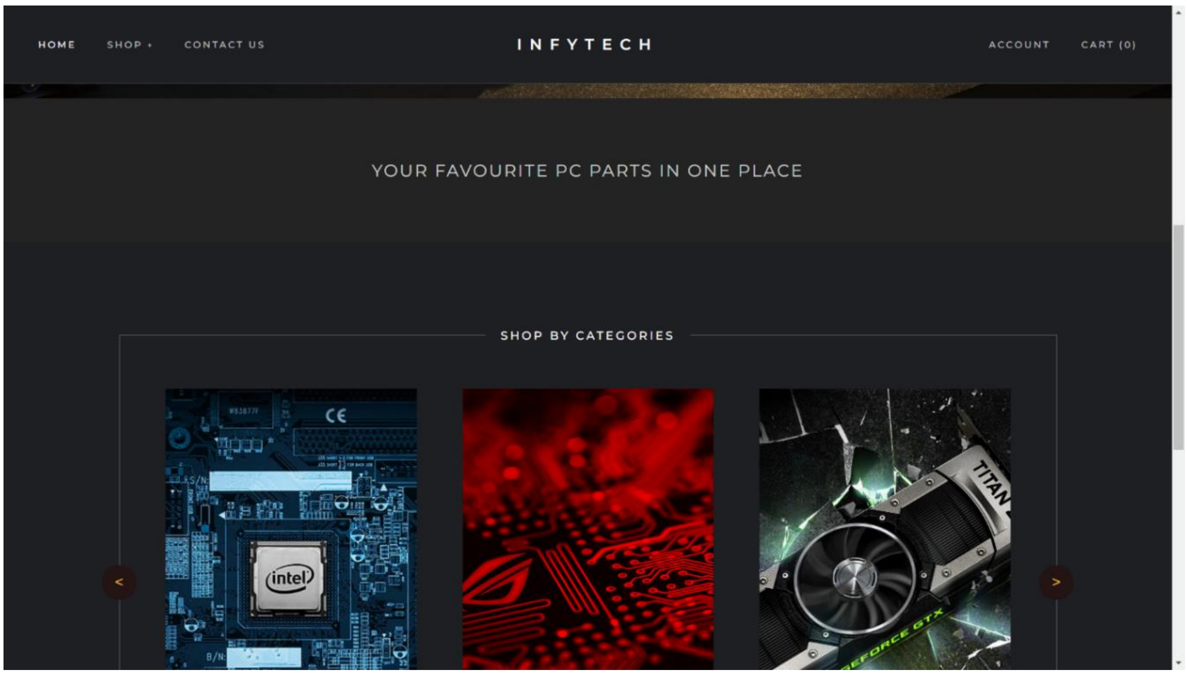
```
("use strict");
const stripe = require("stripe")(process.env.STRIPE_KEY);
const { createCoreController } = require("@strapi/strapi").factories;
module.exports = createCoreController("api::order.order", ({ strapi })
=> ({
  async create(ctx) {
    const { products } = ctx.request.body;
    try {
      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.company + " " + "-" + " " + item.name,
              },
              unit_amount: Math.round(item.price * 100),
            },
            quantity: product.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=true",
        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 };
    }
  },
}));
```

8.2 Screenshots :

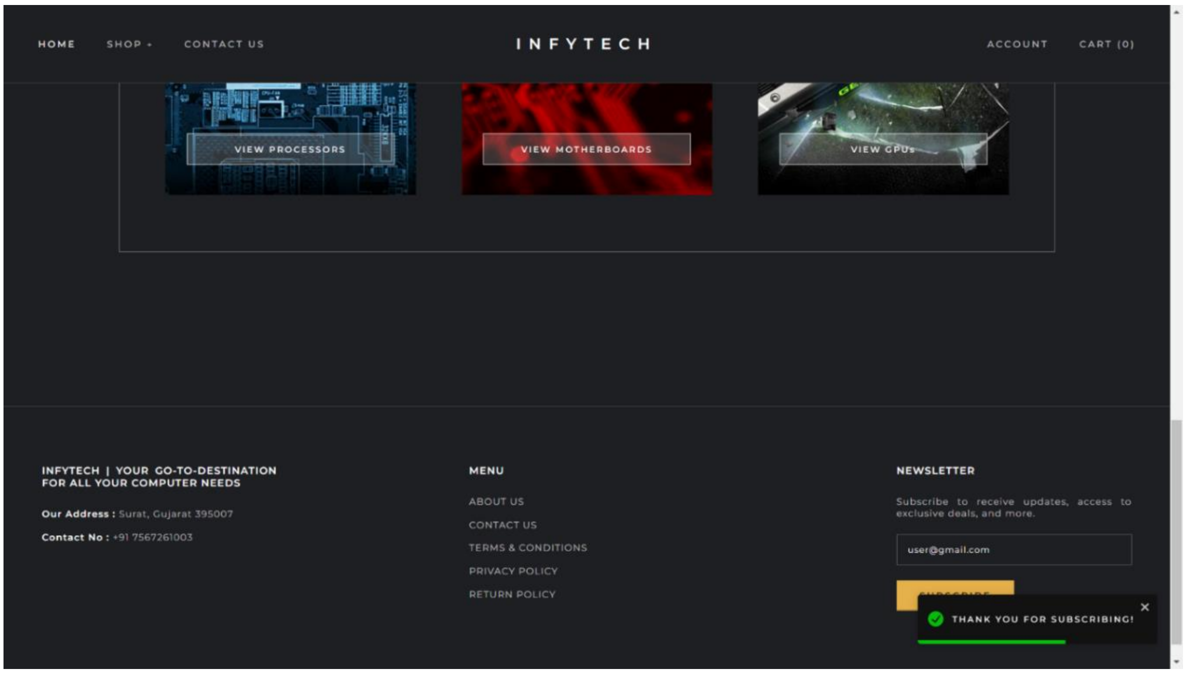
Home Page (1) :



Home Page (2) :



Home Page (3) :



INFYTECH | YOUR GO-TO-DESTINATION FOR ALL YOUR COMPUTER NEEDS

Our Address : Surat, Gujarat 395007

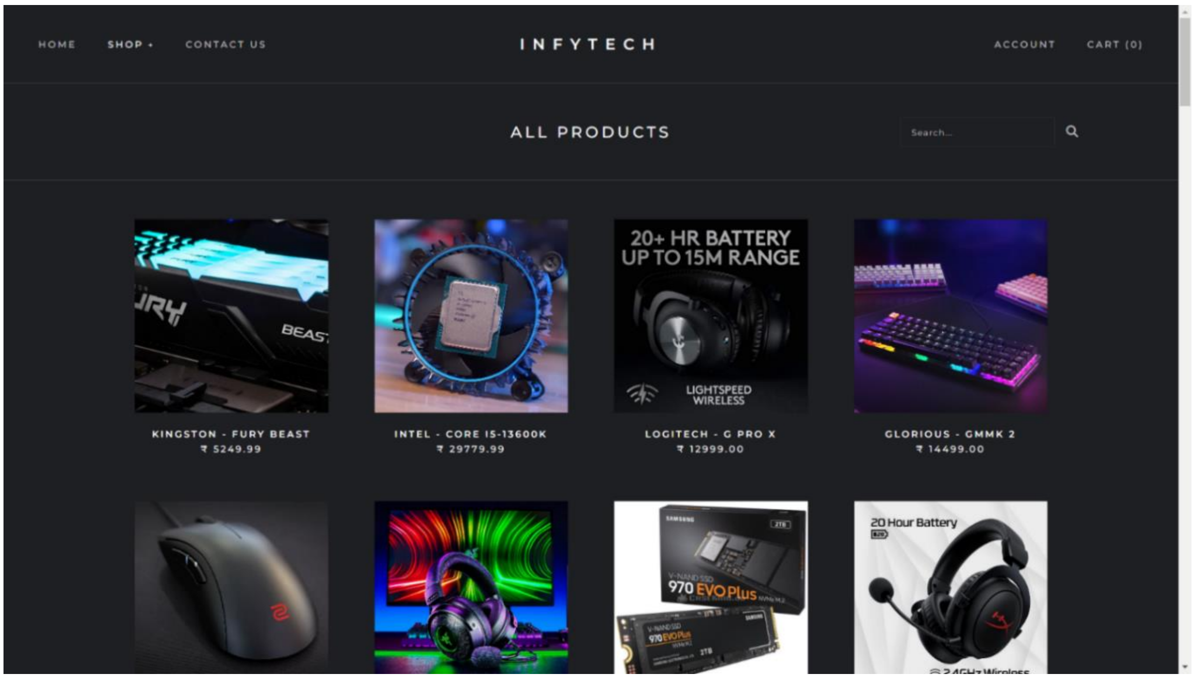
Contact No : +91 7567261003

Subscribe to receive updates, access to exclusive deals, and more.

user@gmail.com

THANK YOU FOR SUBSCRIBING!

All Products Page :



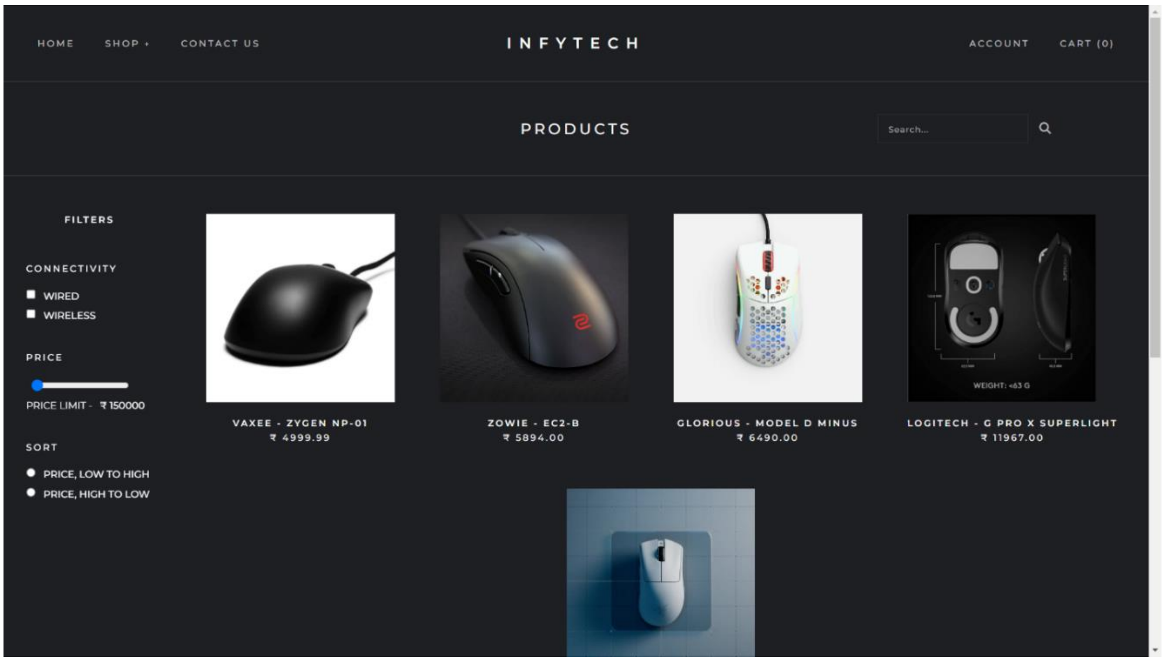
KINGSTON - FURY BEAST
₹ 5249.99

INTEL - CORE I5-13600K
₹ 29779.99

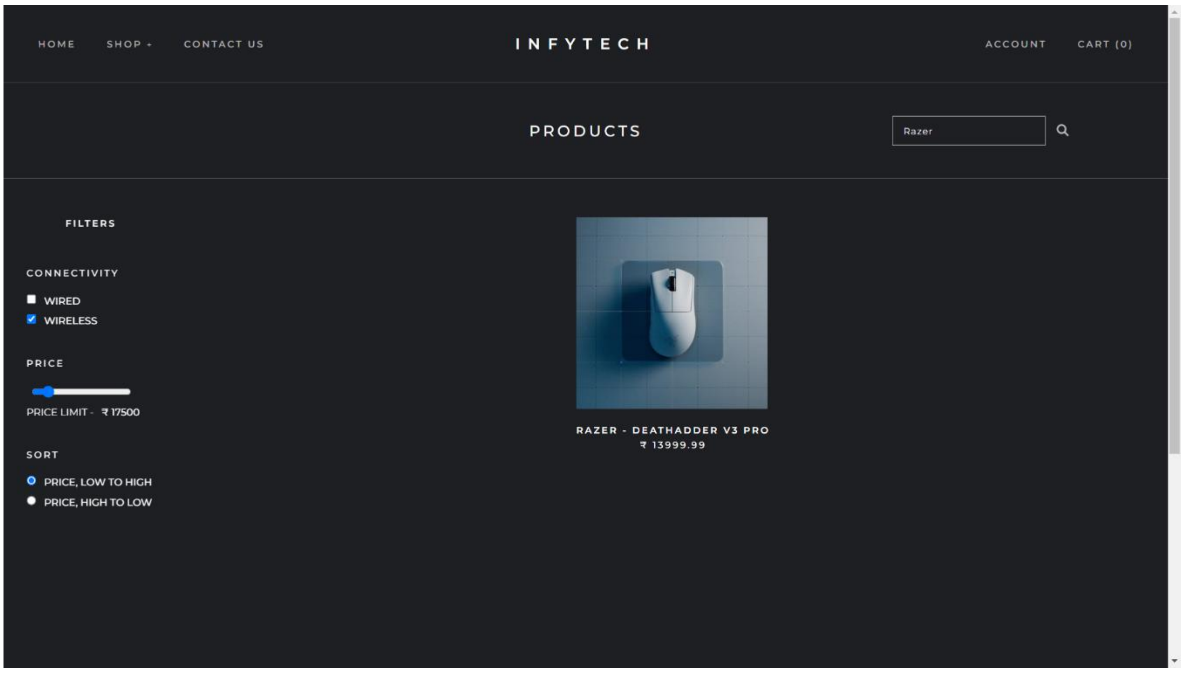
LOGITECH - G PRO X
₹ 12999.00

GLORIOUS - GMMK 2
₹ 14499.00

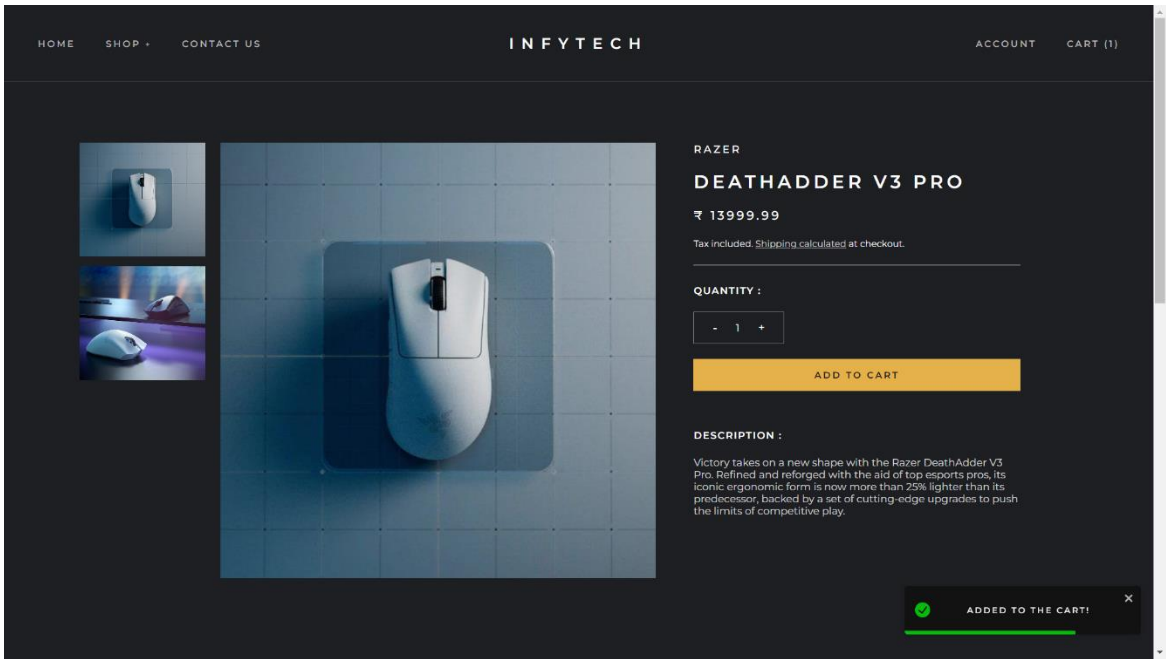
Category Page :



Filtered Category Page :



Product Details Page (1) :



RAZER

DEATHADDER V3 PRO

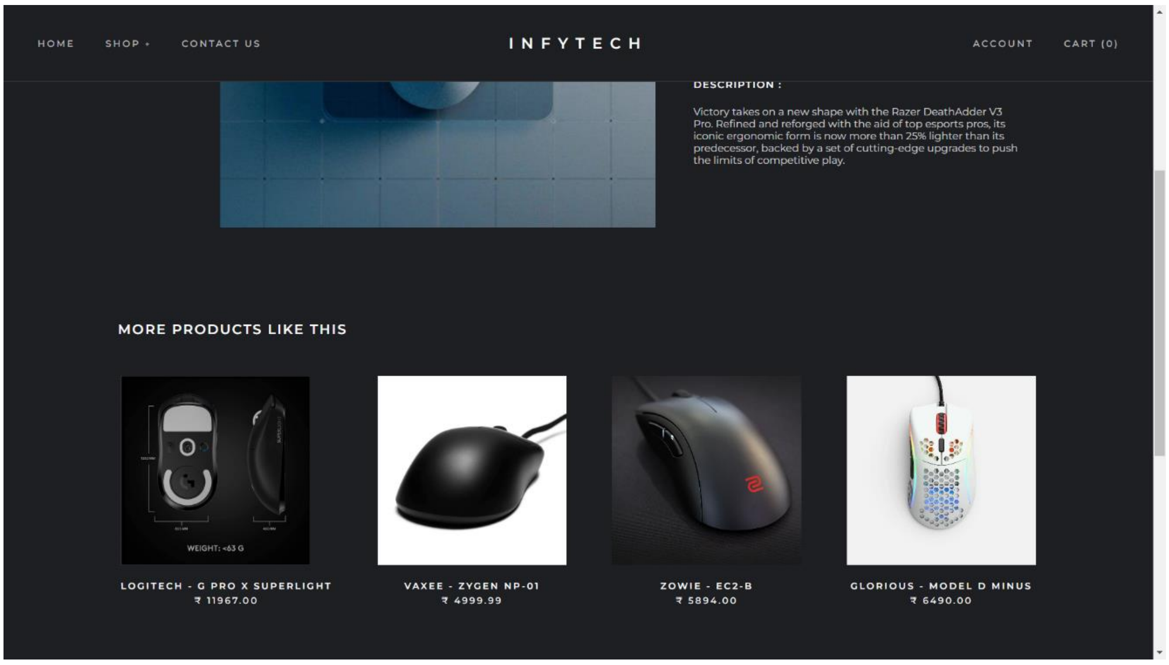
₹ 13999.99

Tax included. Shipping calculated at checkout.

DESCRIPTION :

Victory takes on a new shape with the Razer DeathAdder V3 Pro. Refined and reforged with the aid of top esports pros, its iconic ergonomic form is now more than 25% lighter than its predecessor, backed by a set of cutting-edge upgrades to push the limits of competitive play.

Product Details Page (2) :



DESCRIPTION :

Victory takes on a new shape with the Razer DeathAdder V3 Pro. Refined and reforged with the aid of top esports pros, its iconic ergonomic form is now more than 25% lighter than its predecessor, backed by a set of cutting-edge upgrades to push the limits of competitive play.

MORE PRODUCTS LIKE THIS

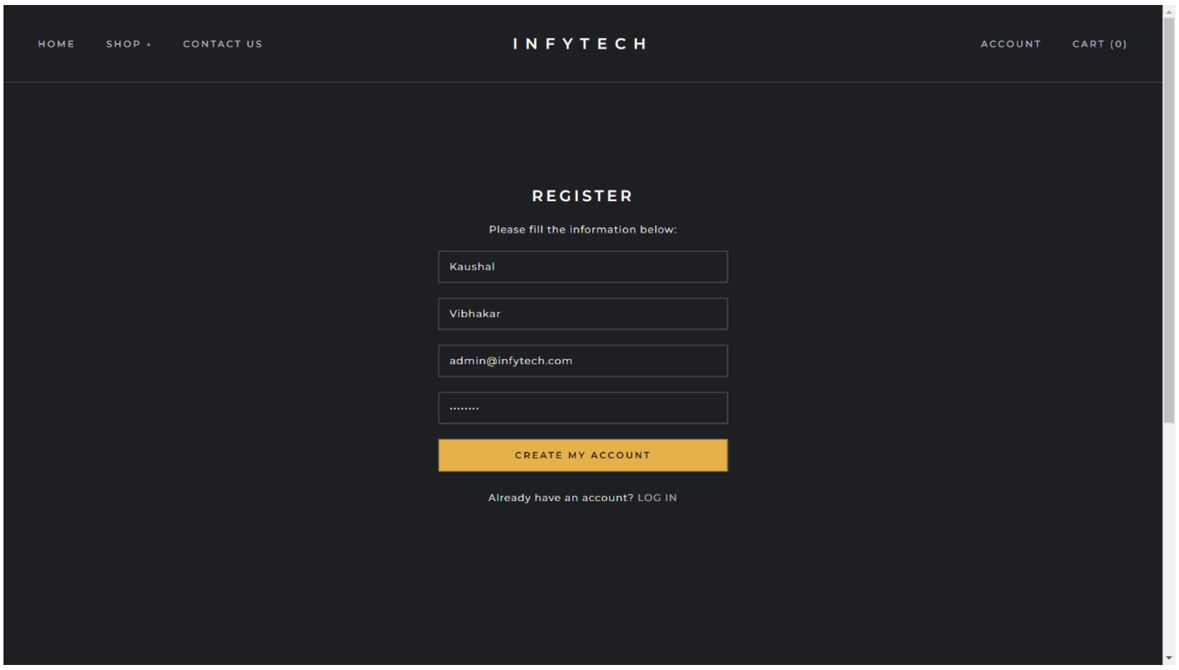
LOGITECH - G PRO X SUPERLIGHT
₹ 11967.00

VAXEE - ZYGEN NP-01
₹ 4999.99

ZOWIE - EC2-B
₹ 5894.00

GLORIOUS - MODEL D MINUS
₹ 6490.00

Register Page :



REGISTER

Please fill the information below:

Kaushal

Vibhakar

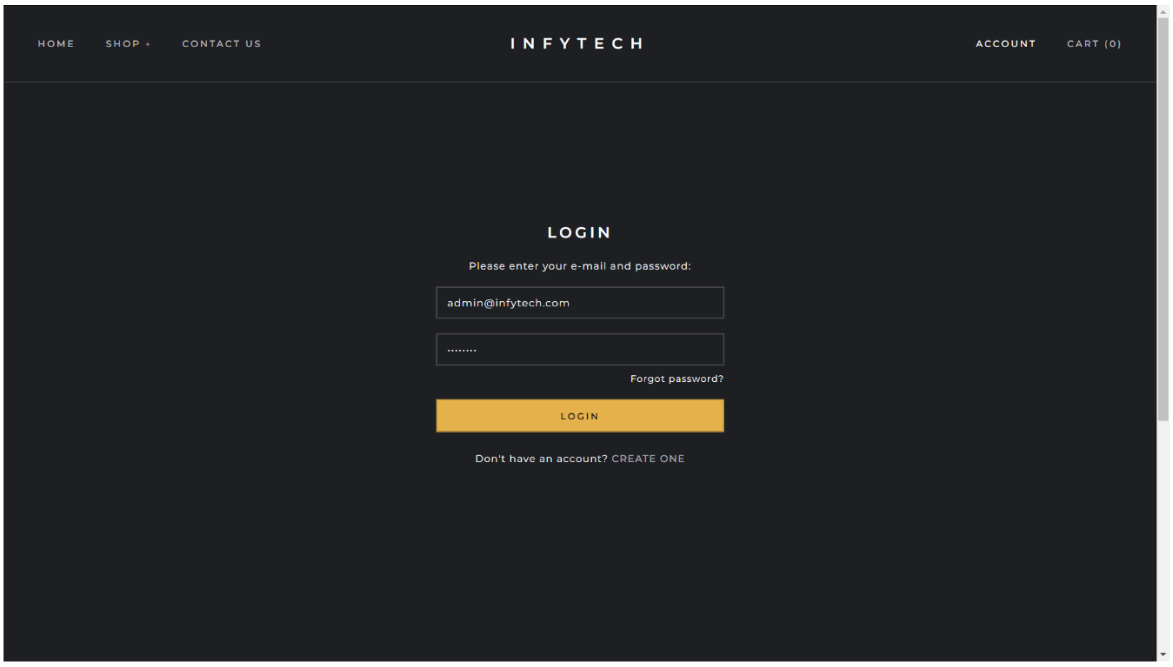
admin@infytech.com

....

CREATE MY ACCOUNT

Already have an account? LOG IN

Login Page :



LOGIN

Please enter your e-mail and password:

admin@infytech.com

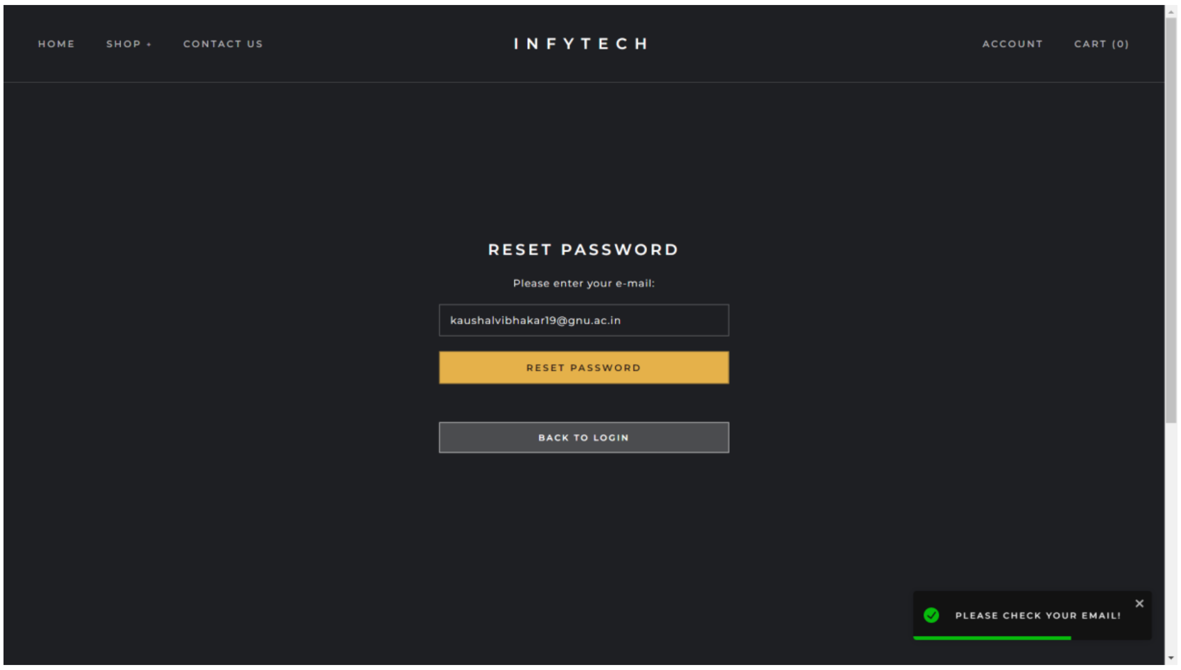
....

Forgot password?

LOGIN

Don't have an account? CREATE ONE

Reset Password Page :



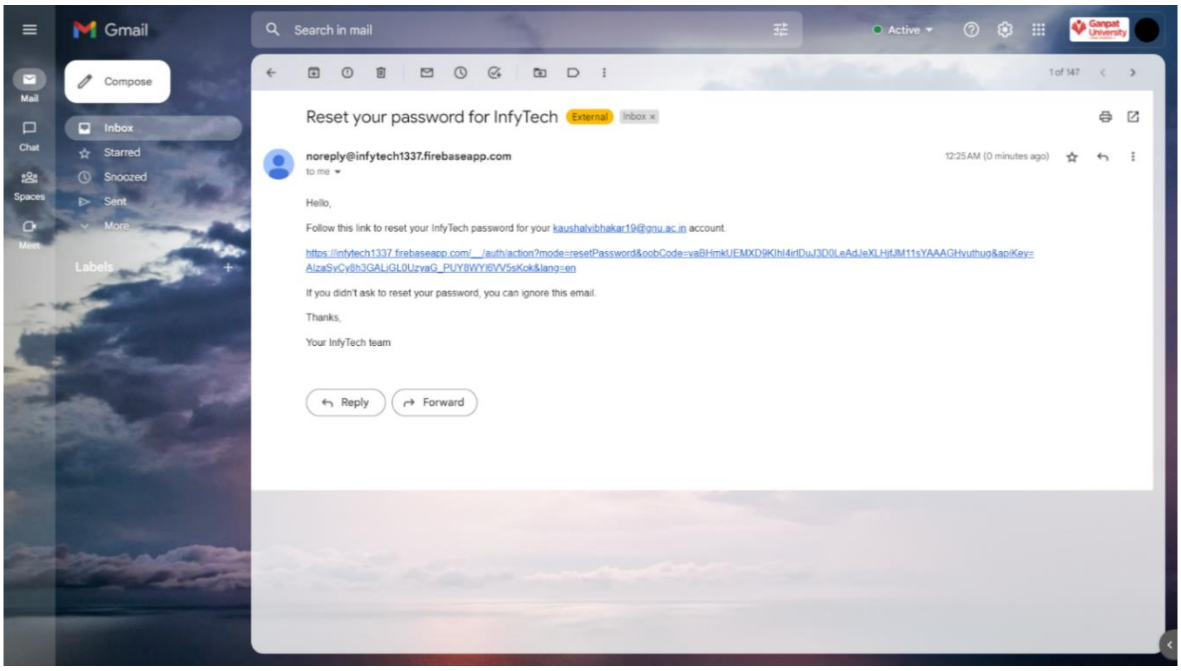
RESET PASSWORD

Please enter your e-mail:

kaushalvibhakar19@gnu.ac.in

PLEASE CHECK YOUR EMAIL!

Reset Password Mail :



Reset your password for InfyTech

noreply@infytech1337.firebaseapp.com
to me

12:25 AM (0 minutes ago)

Hello,

Follow this link to reset your InfyTech password for your kaushalvibhakar19@gnu.ac.in account.

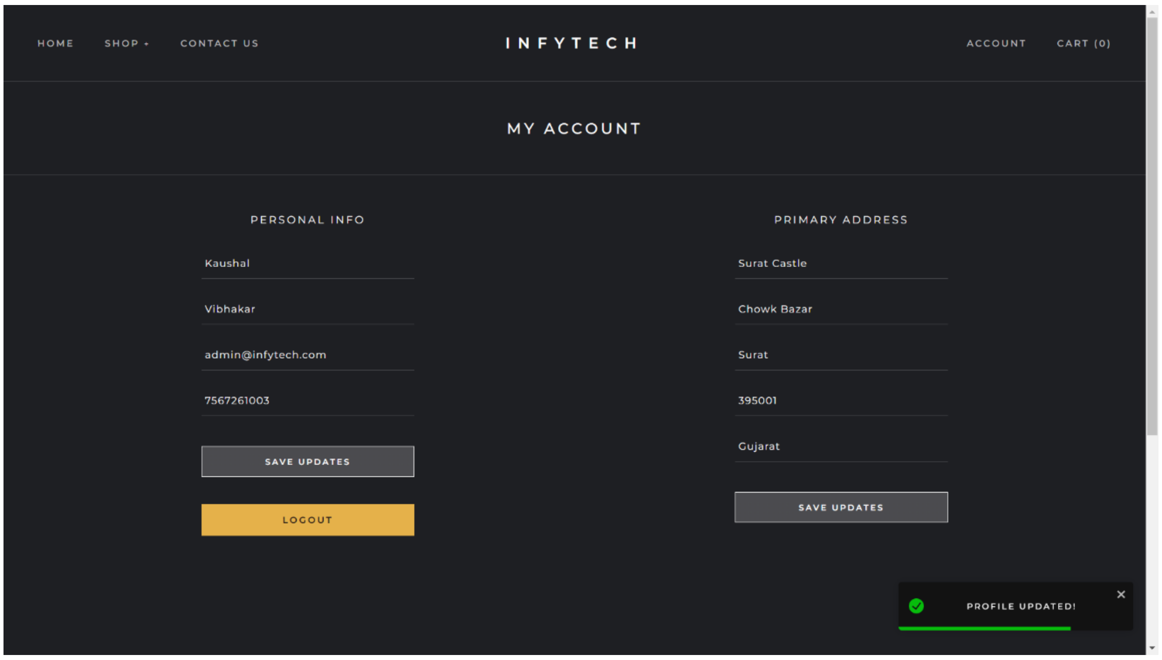
https://infytech1337.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=vaBHmkUEMXD9KIhl4irlDuJ3D0LeAdJeXLHjfJM11sYAAAGHvuthug&apiKey=AIzaSyCy8h3GALjGL0UzyaG_PUY8WYI6VV5sKok&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your InfyTech team

Profile Page :



MY ACCOUNT

PERSONAL INFO

Kaushal

Vibhakar

admin@infytech.com

7567261003

SAVE UPDATES

LOGOUT

PRIMARY ADDRESS

Surat Castle

Chowk Bazar

Surat

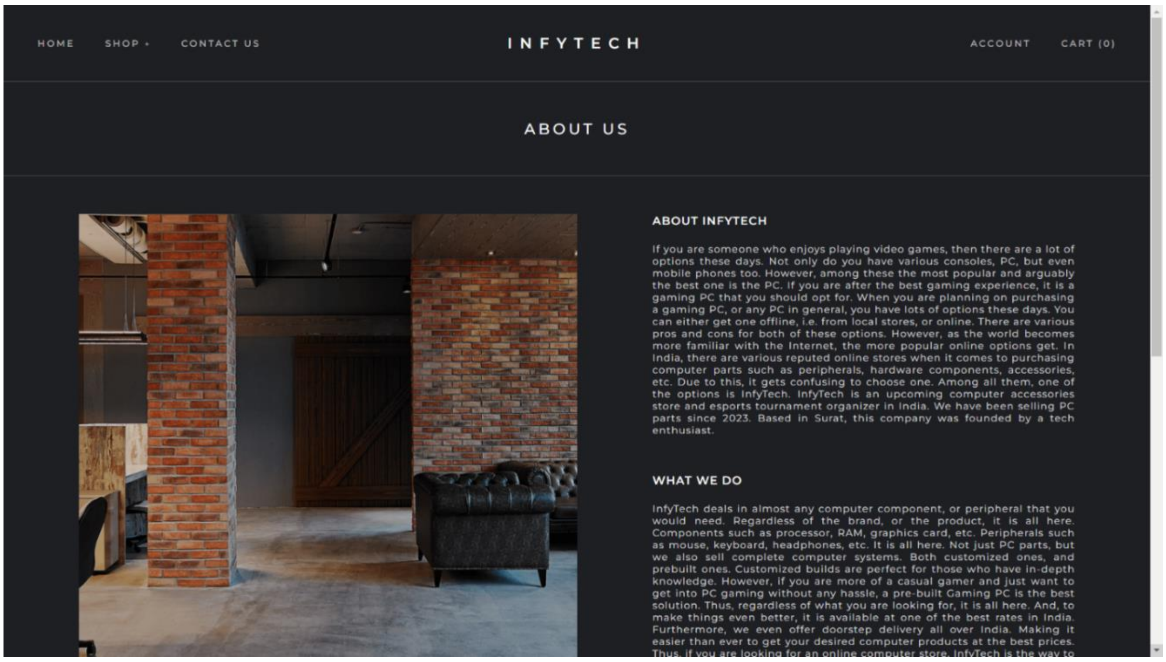
395001

Gujarat

SAVE UPDATES

PROFILE UPDATED!

About Us Page :



ABOUT US

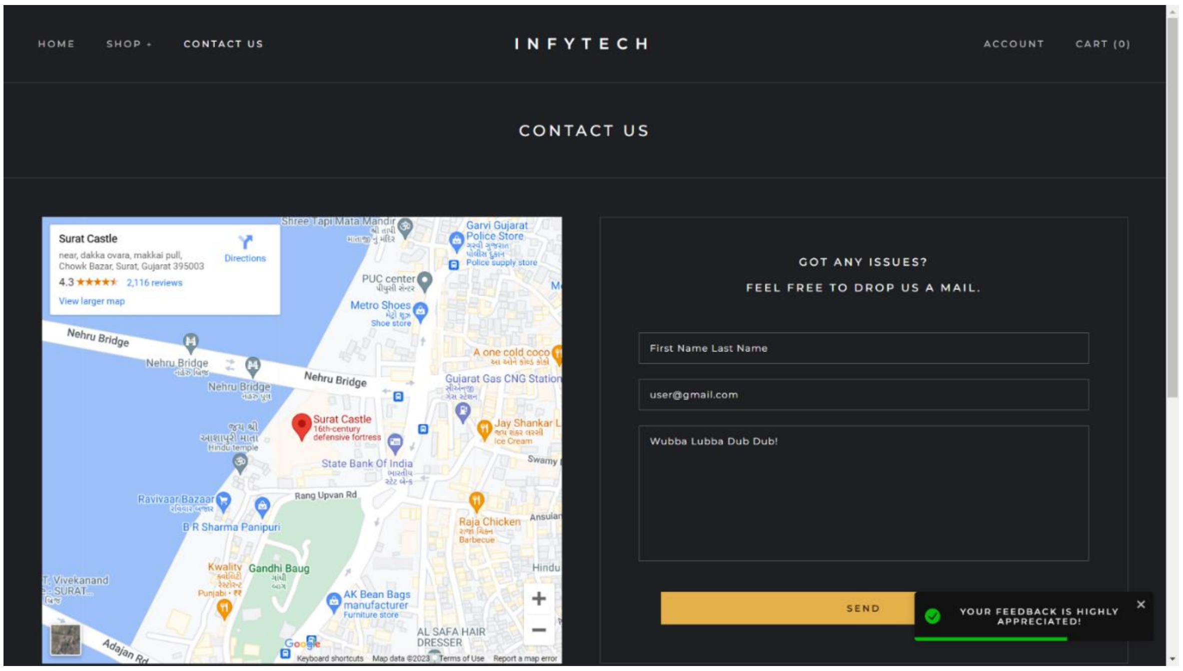
ABOUT INFYTECH

If you are someone who enjoys playing video games, then there are a lot of options these days. Not only do you have various consoles, PC, but even mobile phones too. However, among these the most popular and arguably the best one is the PC. If you are after the best gaming experience, it is a gaming PC that you should opt for. When you are planning on purchasing a gaming PC, or any PC in general, you have lots of options these days. You can either get one offline, i.e. from local stores, or online. There are various pros and cons for both of these options. However, as the world becomes more familiar with the Internet, the more popular online options get. In India, there are various reputed online stores when it comes to purchasing computer parts such as peripherals, hardware components, accessories, etc. Due to this, it gets confusing to choose one. Among all them, one of the options is InfyTech. InfyTech is an upcoming computer accessories store and esports tournament organizer in India. We have been selling PC parts since 2023. Based in Surat, this company was founded by a tech enthusiast.

WHAT WE DO

InfyTech deals in almost any computer component, or peripheral that you would need. Regardless of the brand, or the product, it is all here. Components such as processor, RAM, graphics card, etc. Peripherals such as mouse, keyboard, headphones, etc. It is all here. Not just PC parts, but we also sell complete computer systems. Both customized ones, and prebuilt ones. Customized builds are perfect for those who have in-depth knowledge. However, if you are more of a casual gamer and just want to get into PC gaming without any hassle, a pre-built Gaming PC is the best solution. Thus, regardless of what you are looking for, it is all here. And, to make things even better, it is available at one of the best rates in India. Furthermore, we even offer doorstep delivery all over India. Making it easier than ever to get your desired computer products at the best prices. Thus, if you are looking for an online computer store, InfyTech is the way to

Contact Us Page :

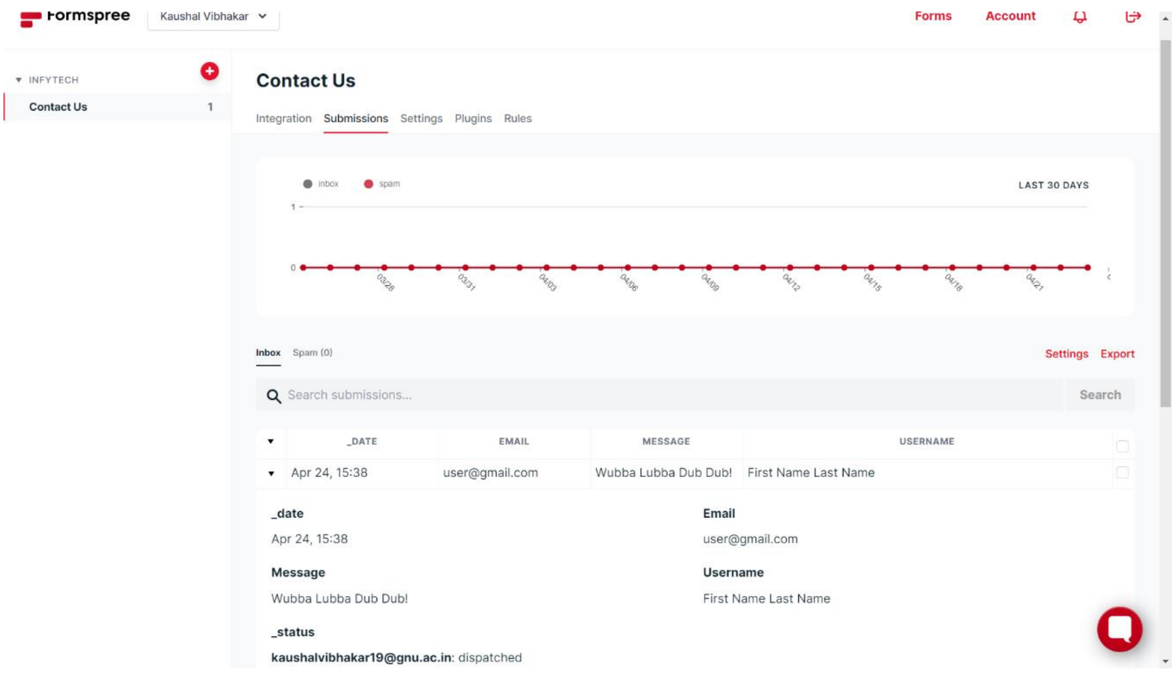


First Name Last Name

user@gmail.com

Wubba Lubba Dub Dub!

Contact Us Form :



_DATE	EMAIL	MESSAGE	USERNAME
Apr 24, 15:38	user@gmail.com	Wubba Lubba Dub Dub!	First Name Last Name

_date

Apr 24, 15:38

Email

user@gmail.com

Message

Wubba Lubba Dub Dub!

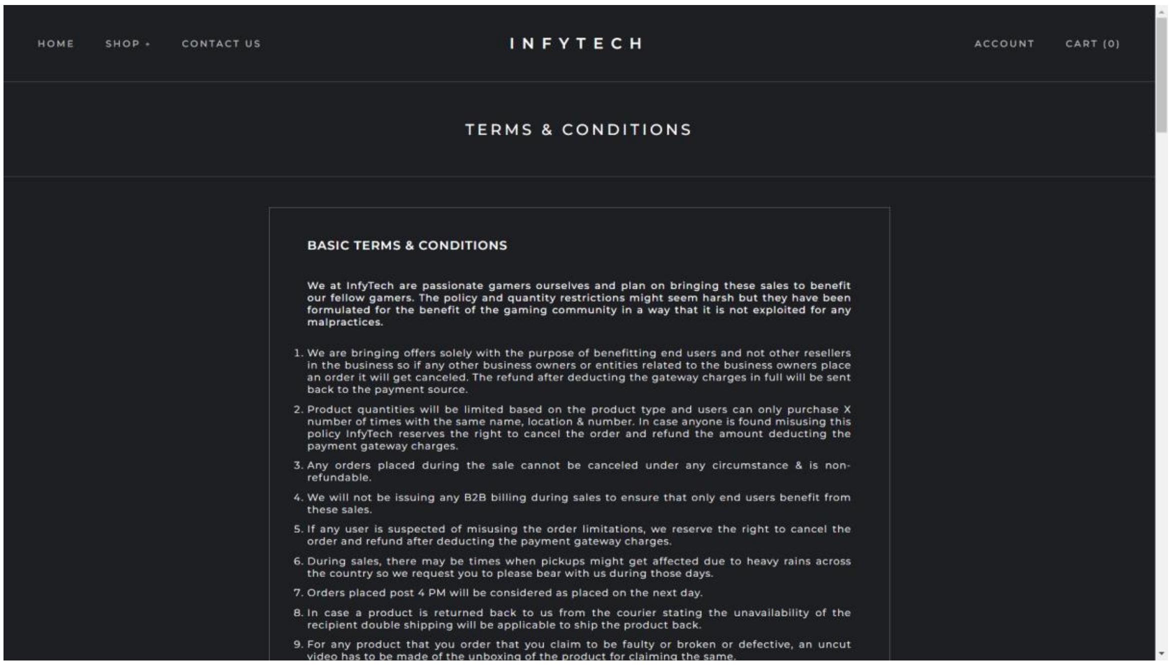
Username

First Name Last Name

_status

kaushalvibhakar19@gnu.ac.in: dispatched

Terms & Conditions Page :



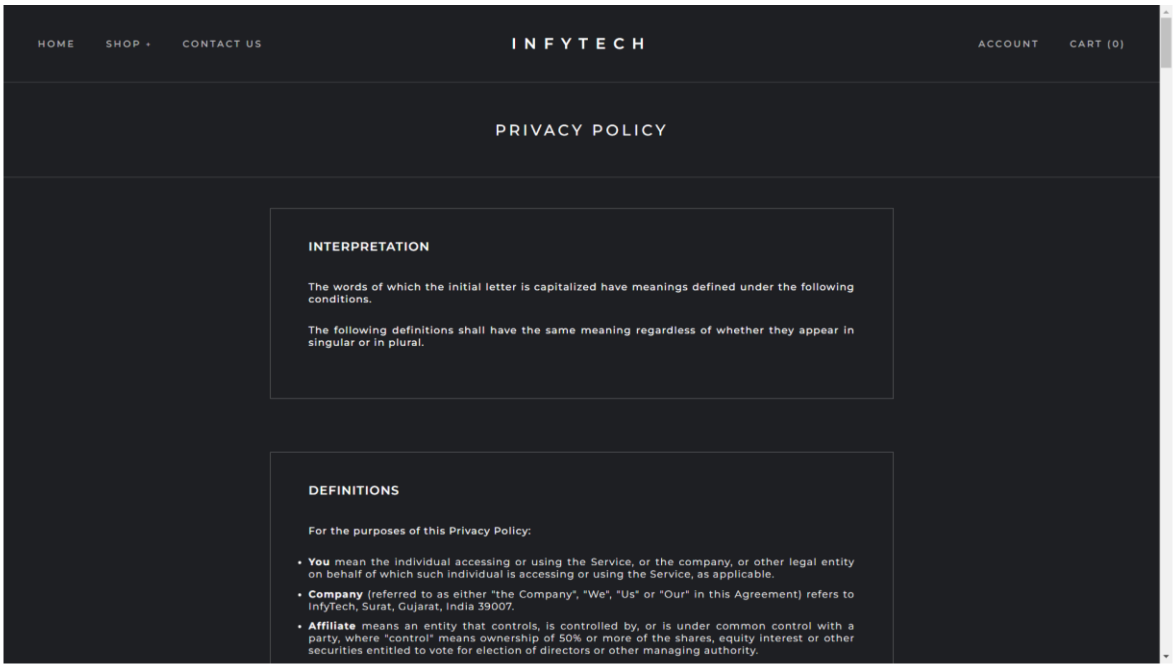
TERMS & CONDITIONS

BASIC TERMS & CONDITIONS

We at InfyTech are passionate gamers ourselves and plan on bringing these sales to benefit our fellow gamers. The policy and quantity restrictions might seem harsh but they have been formulated for the benefit of the gaming community in a way that it is not exploited for any malpractices.

1. We are bringing offers solely with the purpose of benefitting end users and not other resellers in the business so if any other business owners or entities related to the business owners place an order it will get canceled. The refund after deducting the gateway charges in full will be sent back to the payment source.
2. Product quantities will be limited based on the product type and users can only purchase X number of times with the same name, location & number. In case anyone is found misusing this policy InfyTech reserves the right to cancel the order and refund the amount deducting the payment gateway charges.
3. Any orders placed during the sale cannot be canceled under any circumstance & is non-refundable.
4. We will not be issuing any B2B billing during sales to ensure that only end users benefit from these sales.
5. If any user is suspected of misusing the order limitations, we reserve the right to cancel the order and refund after deducting the payment gateway charges.
6. During sales, there may be times when pickups might get affected due to heavy rains across the country so we request you to please bear with us during those days.
7. Orders placed post 4 PM will be considered as placed on the next day.
8. In case a product is returned back to us from the courier stating the unavailability of the recipient double shipping will be applicable to ship the product back.
9. For any product that you order that you claim to be faulty or broken or defective, an uncut video has to be made of the unboxing of the product for claiming the same.

Privacy Policy Page :



PRIVACY POLICY

INTERPRETATION

The words of which the initial letter is capitalized have meanings defined under the following conditions.

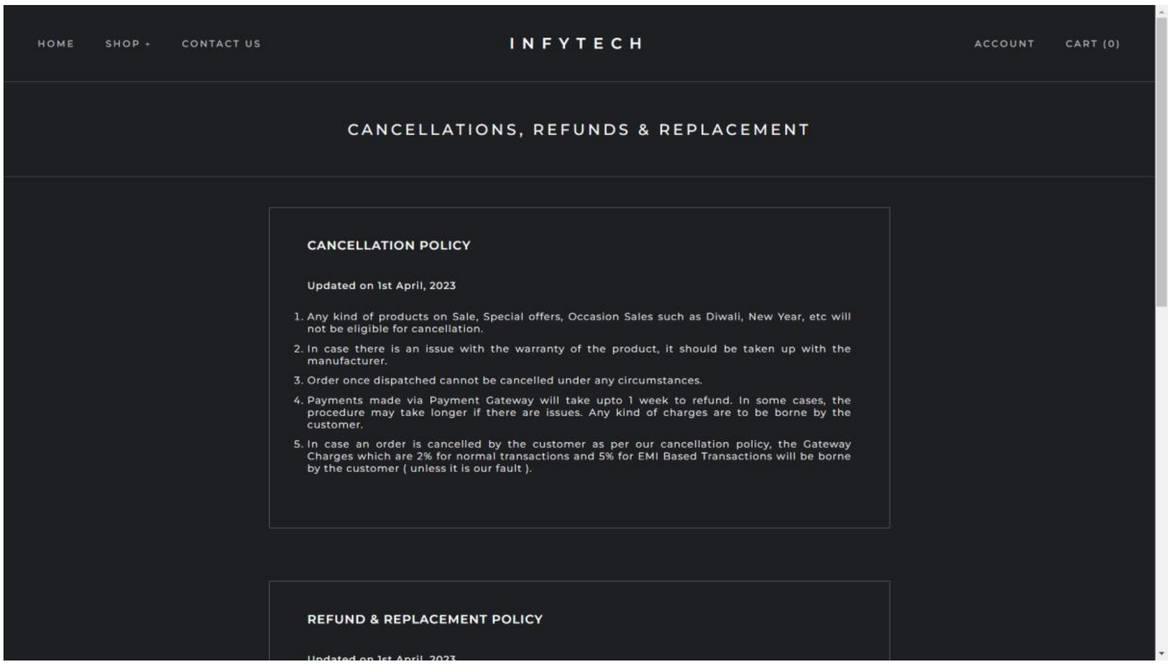
The following definitions shall have the same meaning regardless of whether they appear in singular or in plural.

DEFINITIONS

For the purposes of this Privacy Policy:

- **You** mean the individual accessing or using the Service, or the company, or other legal entity on behalf of which such individual is accessing or using the Service, as applicable.
- **Company** (referred to as either "the Company", "We", "Us" or "Our" in this Agreement) refers to InfyTech, Surat, Gujarat, India 39007.
- **Affiliate** means an entity that controls, is controlled by, or is under common control with a party, where "control" means ownership of 50% or more of the shares, equity interest or other securities entitled to vote for election of directors or other managing authority.

Return Policy Page :



CANCELLATIONS, REFUNDS & REPLACEMENT

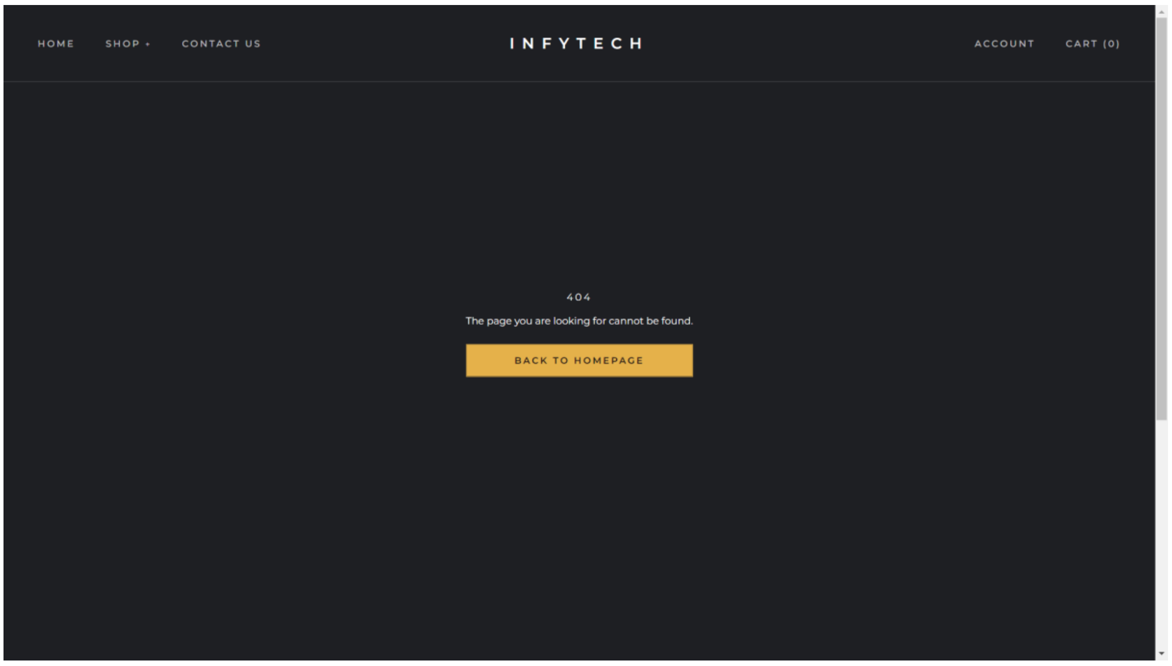
CANCELLATION POLICY

Updated on 1st April, 2023

1. Any kind of products on Sale, Special offers, Occasion Sales such as Diwali, New Year, etc will not be eligible for cancellation.
2. In case there is an issue with the warranty of the product, it should be taken up with the manufacturer.
3. Order once dispatched cannot be cancelled under any circumstances.
4. Payments made via Payment Gateway will take upto 1 week to refund. In some cases, the procedure may take longer if there are issues. Any kind of charges are to be borne by the customer.
5. In case an order is cancelled by the customer as per our cancellation policy, the Gateway Charges which are 2% for normal transactions and 5% for EMI Based Transactions will be borne by the customer (unless it is our fault).

REFUND & REPLACEMENT POLICY

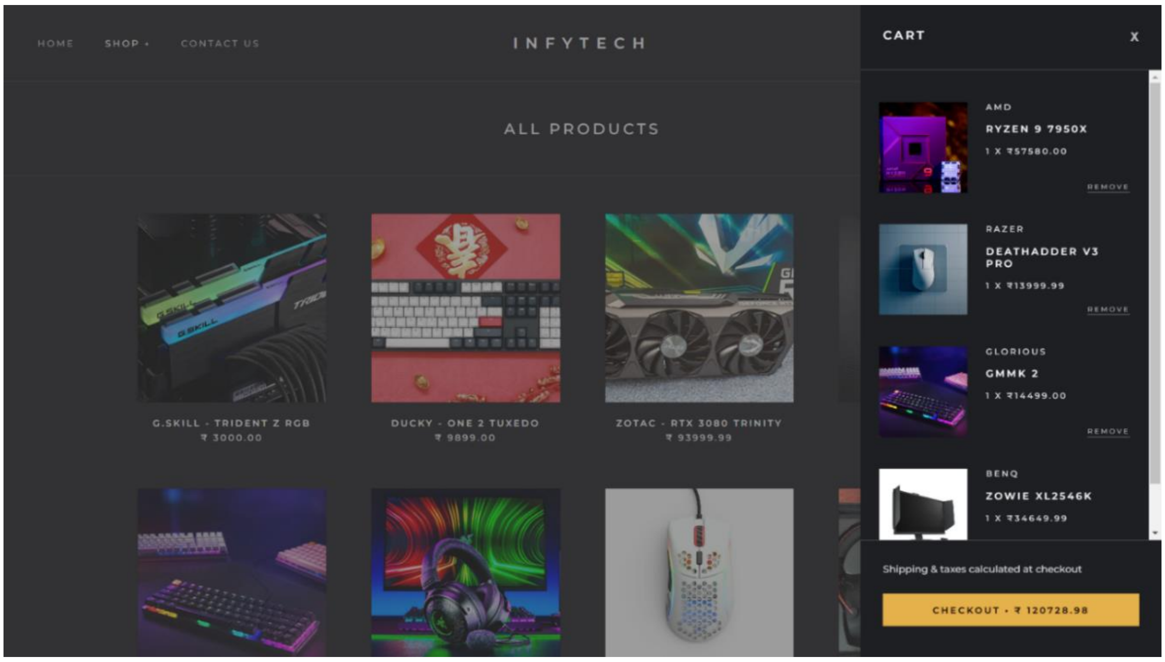
Error Page :



404

The page you are looking for cannot be found.

Cart Component :



CART X

AMD
RYZEN 9 7950X
1 X ₹57580.00
REMOVE

RAZER
DEATHADDER V3 PRO
1 X ₹13999.99
REMOVE

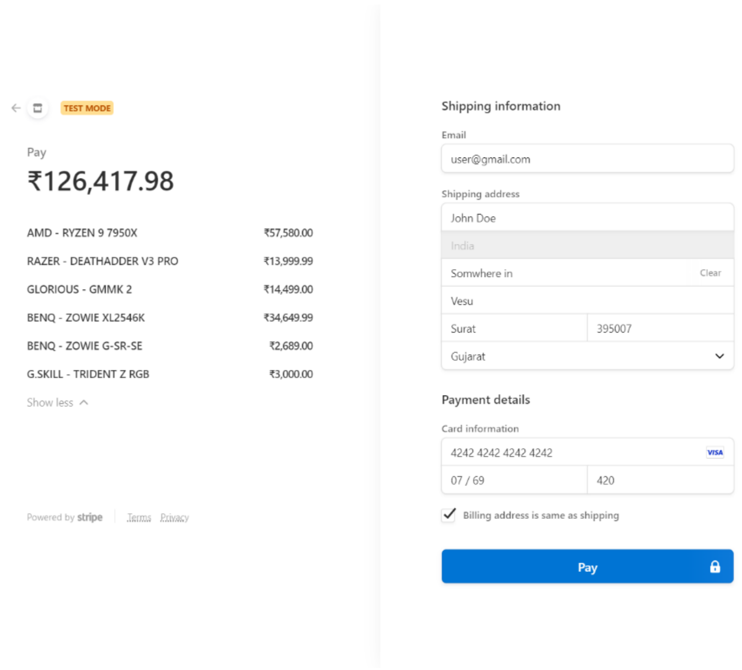
GLORIOUS
GMMK 2
1 X ₹14499.00
REMOVE

BENQ
ZOWIE XL2546K
1 X ₹34649.99

Shipping & taxes calculated at checkout

CHECKOUT • ₹ 120728.98

Checkout Page :


TEST MODE

Pay
₹126,417.98

AMD - RYZEN 9 7950X	₹57,580.00
RAZER - DEATHADDER V3 PRO	₹13,999.99
GLORIOUS - GMMK 2	₹14,499.00
BENQ - ZOWIE XL2546K	₹34,649.99
BENQ - ZOWIE G-SR-SE	₹2,689.00
G.SKILL - TRIDENT Z RGB	₹3,000.00

Show less

Powered by stripe | Terms Privacy

Shipping information

Email: user@gmail.com

Shipping address:
John Doe
India
Somwhere in
Vesu
Surat 395007
Gujarat

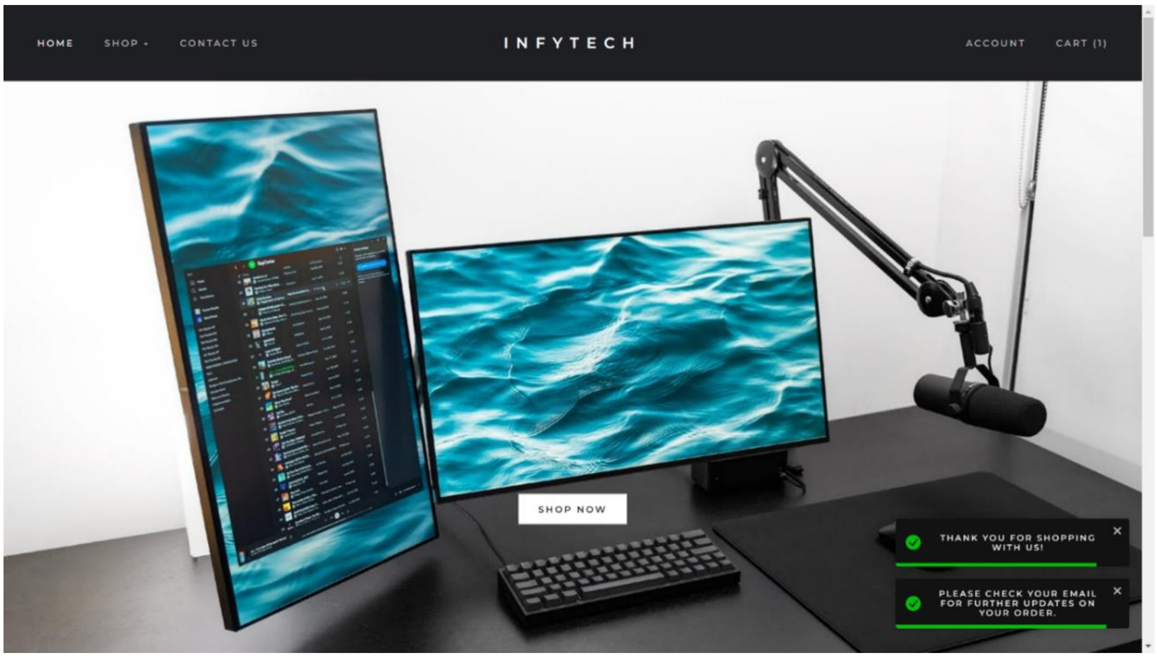
Payment details

Card information:
4242 4242 4242 4242 VISA
07 / 69 420

✓ Billing address is same as shipping

Pay

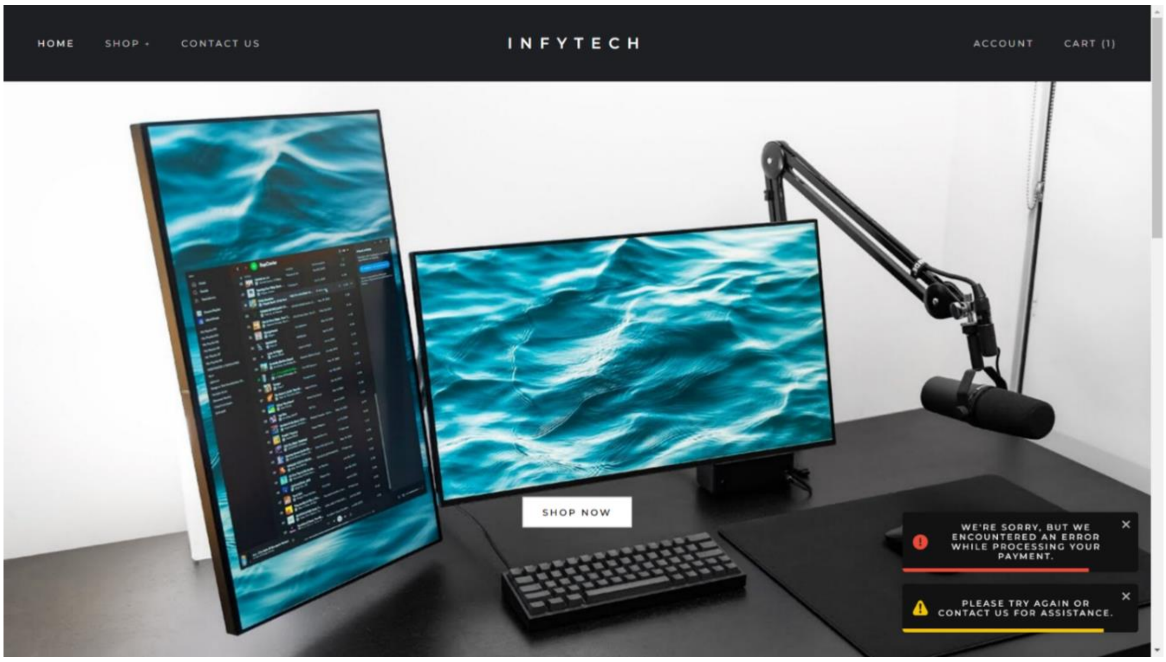
On Order Success :



THANK YOU FOR SHOPPING WITH US!

PLEASE CHECK YOUR EMAIL FOR FURTHER UPDATES ON YOUR ORDER.

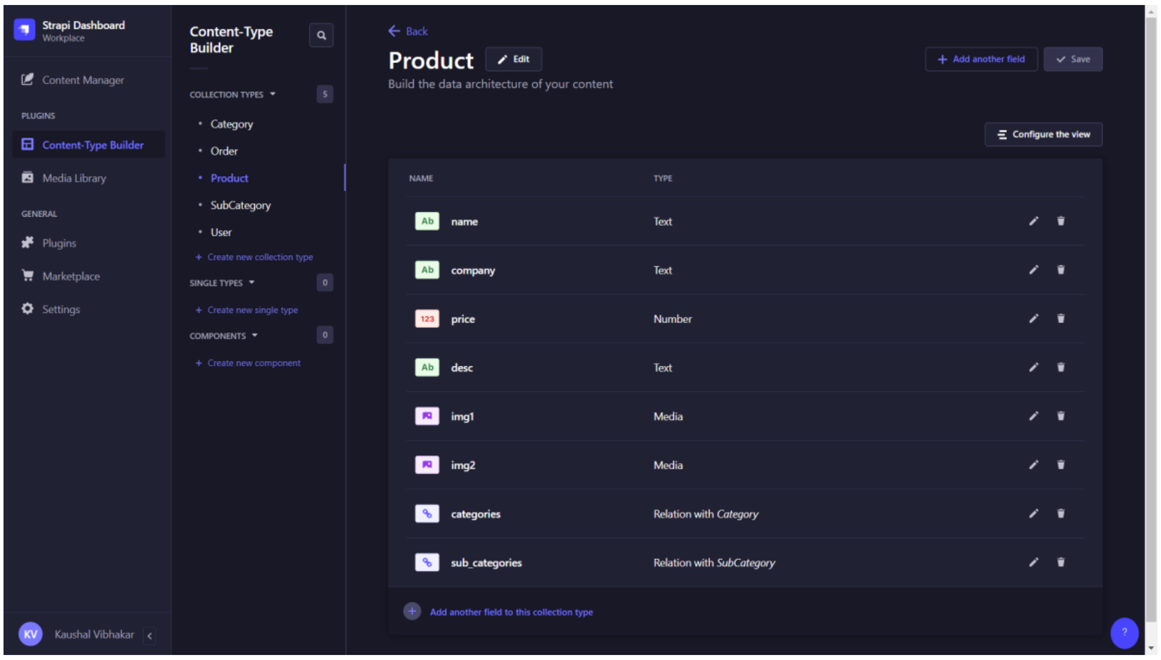
On Order Failure :



WE'RE SORRY, BUT WE ENCOUNTERED AN ERROR WHILE PROCESSING YOUR PAYMENT.

PLEASE TRY AGAIN OR CONTACT US FOR ASSISTANCE.

Strapi Content Type Builder :

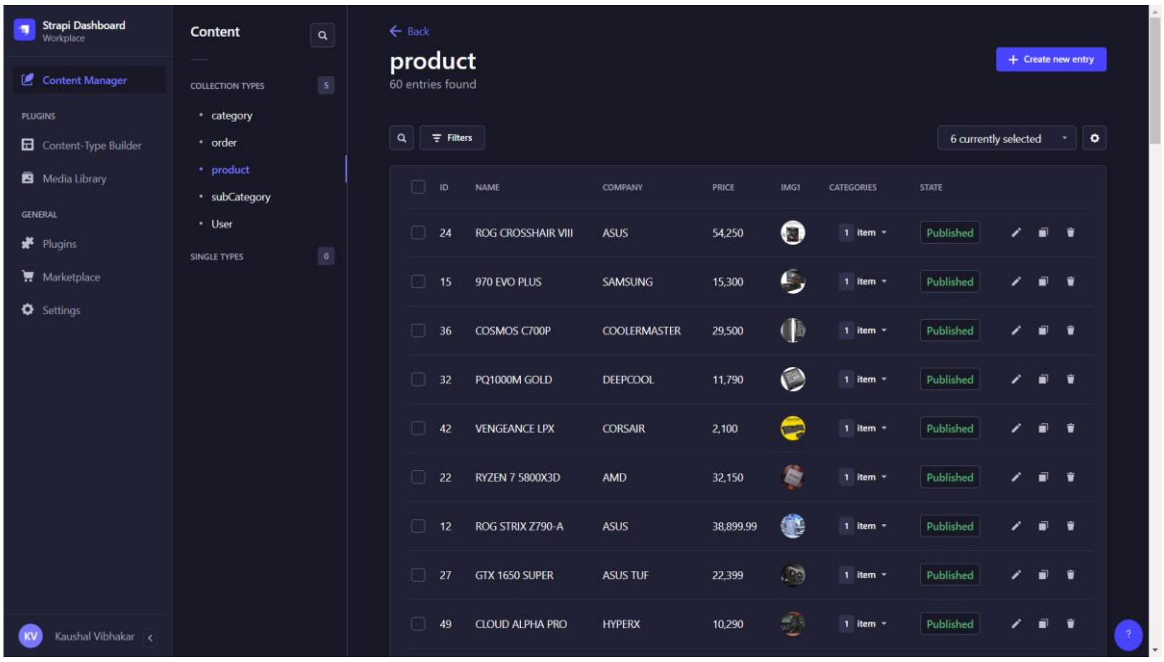


Product

Build the data architecture of your content

NAME	TYPE
name	Text
company	Text
price	Number
desc	Text
img1	Media
img2	Media
categories	Relation with *Category*
sub_categories	Relation with *SubCategory*

Strapi Content Manager Page :

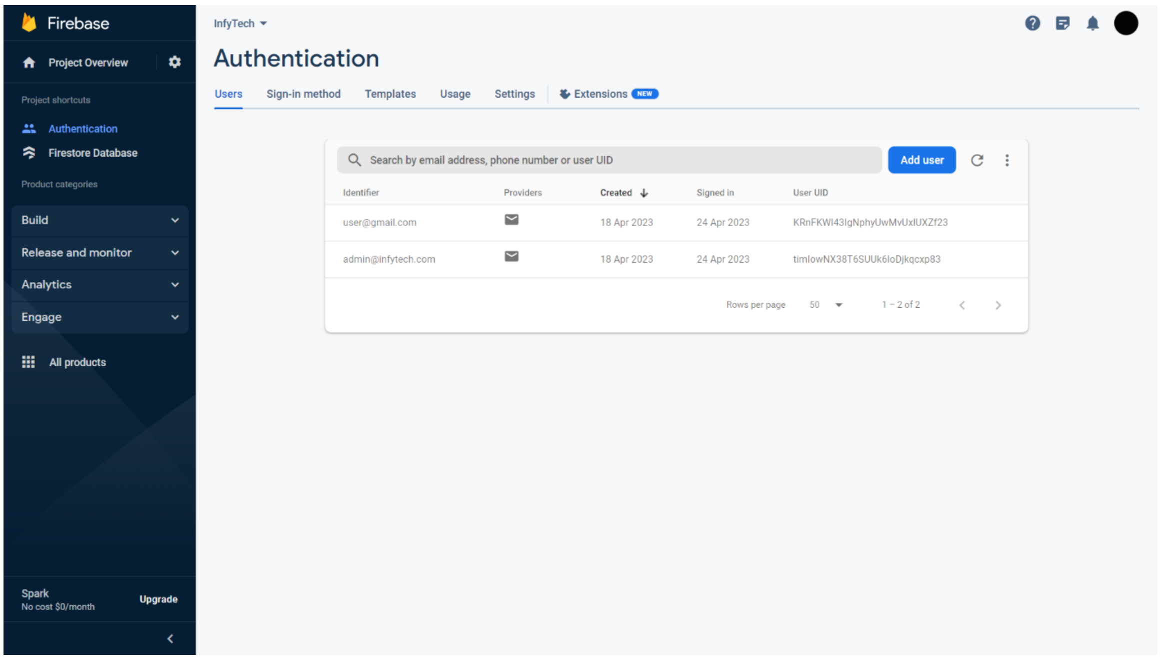


product

60 entries found

ID	NAME	COMPANY	PRICE	IMG1	CATEGORIES	STATE
24	ROG CROSSHAIR VIII	ASUS	54,250		1 item	Published
15	970 EVO PLUS	SAMSUNG	15,300		1 item	Published
36	COSMOS C700P	COOLERMASTER	29,500		1 item	Published
32	PQ1000M GOLD	DEEPCOOL	11,790		1 item	Published
42	VENGEANCE LPX	CORSAIR	2,100		1 item	Published
22	RYZEN 7 5800X3D	AMD	32,150		1 item	Published
12	ROG STRIX Z790-A	ASUS	38,899.99		1 item	Published
27	GTX 1650 SUPER	ASUS TUF	22,399		1 item	Published
49	CLOUD ALPHA PRO	HYPERX	10,290		1 item	Published

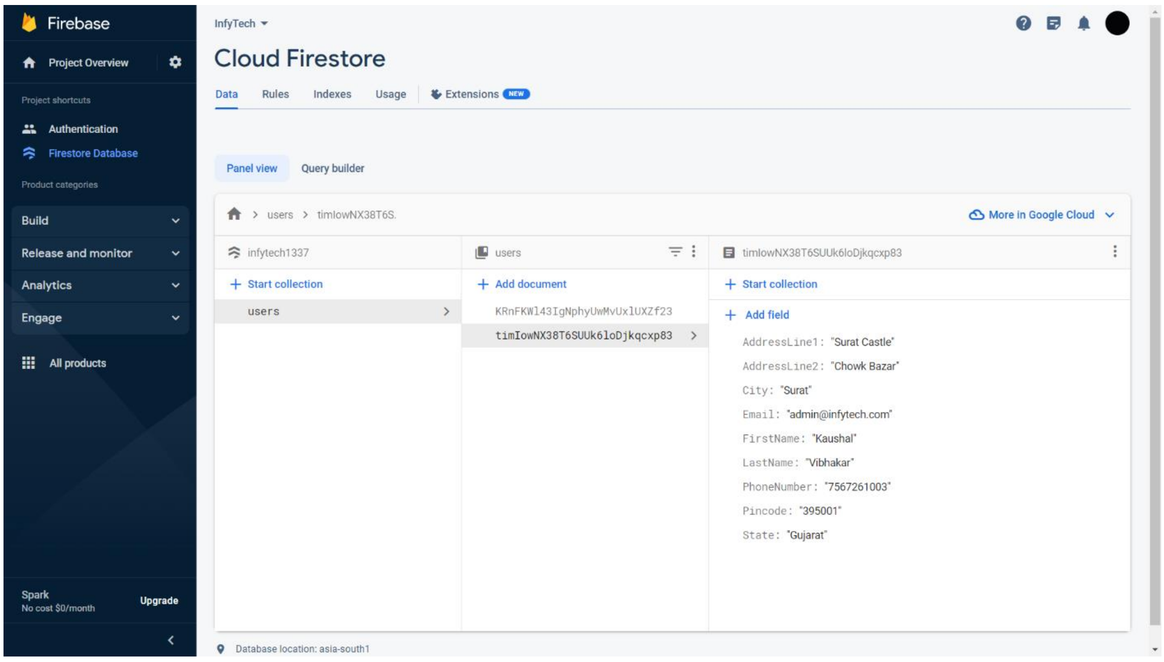
Firebase Authentication :



Identifier	Providers	Created	Signed in	User UID
user@gmail.com	✉	18 Apr 2023	24 Apr 2023	KRnFKWI43IgNphyUwMvUxlUXZf23
admin@infytech.com	✉	18 Apr 2023	24 Apr 2023	timlowNX38T6SUUk6loDjkqcxp83

Rows per page 50 1 – 2 of 2

Firestore Database :



```
AddressLine1: "Surat Castle"
AddressLine2: "Chowk Bazar"
City: "Surat"
Email: "admin@infytech.com"
FirstName: "Kaushal"
LastName: "Vibhakar"
PhoneNumber: "7567261003"
Pincode: "395001"
State: "Gujarat"
```

Database location: asia-south1

9. Testing

9.1 Testing Plan :

The objective is to test the functionality of the ‘InfyTech’ Web Application. Testing is done on both frontend and backend of the application in the Windows environments. The features, which are to be tested, are Login Page, Signup Page and Profile Page. All the major functionality of the application should work as intended and the pass percentage of test cases should be more than 95% and there should not be any critical bugs.

9.2 Test Cases :

The following are the test cases for all modules :

Login Module :

Id	Test Senario	Test Steps	Test Data	Expected Results	Actual Results	Pass /Fail
1	Login with Valid Data	Enter Email Enter Password Click On Login Button	Valid Email Id Valid Password	User Should Login into the Application	As Expected	Pass
2	Login with Valid Data	Enter Email Enter Password Click On Login Button	Valid Email Id Invalid Password	User Should Login into the Application	Incorrect Password	Fail
3	Login with Valid Data	Enter Email Enter Password Click On Login Button	Invalid Email Id Valid Password	User Should Login into the Application	Incorrect Email Id	Fail
4	Login with Valid Data	Enter Email Enter Password Click On Login Button	Invalid Email Id Invalid Password	User Should Login into the Application	Incorrect Email Id and Password	Fail

Table 9.2.1 – Login Test Case Table

Signup Module :

Id	Test Senario	Test_Steps	Test_Data	Expected Results	Actual Results	Pass /Fail
1	Signup with Valid Data	Enter Email id Enter Password Enter Confirm Password Click On Sign up Button	Filled Email Filled Password Filled Confirm Password	User Should Login into the Application	As Expected	Pass
2	Signup with Valid Data	Enter Email id Enter Password Enter Confirm Password Click On Signup Button	Not filled Email Filled Password Filled Confirm Password	User Should Login into the Application	Empty field Email Id	Fail
3	Signup with Valid Data	Enter Email id Enter Password Enter Confirm Password Click On Signup Button	Filled Email Not filled Password Filled Confirm Password	User Should Login into the Application	Empty field Password	Fail
4	Signup with Valid Data	Enter Name Enter Email Enter Password Click On Submit Button	Filled Email Filled Password Not Filled Confirm Password	User Should Login into the Application	Empty field Confirm Password	Fail

Table 9.2.2 – Register Test Case Table

10. Conclusion & Future Scope

10.1 Conclusion :

In conclusion, our project has successfully achieved its objective of developing a fully functional e-commerce website using React, Strapi, and Stripe. We implemented a product catalog with search and filtering options, a shopping cart, and a user authentication system, all integrated with Stripe for secure payment processing. Additionally, we created an admin dashboard with the help of Strapi for managing products, orders, and customers.

Our website provides a seamless shopping experience for customers, with easy-to-use and visually appealing interfaces. While we did not implement order tracking functionality and responsive design, we believe that our website still provides significant value to our users. We recommend future development of the platform to include these features. Overall, our project has made a valuable contribution to the e-commerce industry, and we hope that it will inspire further innovation and development in this field.

10.2 Future Scope :

- More Products as well as Categories to choose from.
- Add more Filters to help Search for the desired Products.
- Customers can Compare Products.
- Customers can Track their Orders.
- Add more Payment methods.

Annexure

References :

- **React Docs -** https://beta.reactjs.org/
- **Redux Docs -** https://redux.js.org/introduction/getting-started
- **Strapi Docs -** https://docs.strapi.io/
- **Firebase Docs -** https://firebase.google.com/docs
- **Stripe Docs -** https://stripe.com/docs

About Tools and Technology :

Information	**Image**
Visual Studio Code – Source Code Editor	Visual Studio Code
React – Frontend Javascript Library	React JS
Sass – Preprocessor Scripting Language Interpreted into CSS	Sass
Strapi – Headless CMS	strapi
Stripe – Payment Processing Platform	stripe
Firebase – Set of Backend Cloud Computing Services	Firebase

Table 11.1 –Tools and Technology Table

About College :

U. V. Patel College of Engineering , Ganpat University

Ganpat University
U.V. Patel College of Engineering
|| विद्यया समाजोत्कर्षः ||
CELEBRATING 25 YEARS OF EXCELLENCE 1997 - 2022
25 Years excellence in innovative technical education in shaping engineers

Ganpat University - U. V. Patel College of Engineering (GUNI-UVPCE) is situated in Ganpat Vidyanagar campus. It was established in September 1997 with the aim of providing educational opportunities to students from various strata of society. It was armed with the vision of educating and training young talented students of Gujarat in the field of Engineering and Technology so that they could meet the demands of Industries in Gujarat and across the globe.

The College is named after Shri Ugarchandbhai Varanasibhai Patel, a leading industrialist of Gujarat, for his generous support. It is a self-financed institute approved by All India Council for Technical Education (AICTE), New Delhi and the Commissionerate of Technical Education, Government of Gujarat.

The College is spread over 25 acres of land and is a part of Ganpat Vidyanagar Campus. It has six ultra-modern buildings of architectural splendor, class rooms, tutorial rooms, seminar halls, offices, drawing hall, workshop, library, well equipped departmental laboratories andseveral computer laboratories with internet connectivity through 1 GBPS Fiber link, satellite link education center with two-way audio and one-way video link.

The Institute offers various undergraduate programs, postgraduate programs, and Ph.D. programs. We aim to create a generation of students that possess technical expertise and are adept at utilizing the technical 'know-hows' in the service of mankind.

We strive towards these Aims and Objectives :

- To offer guidance, motivation, and inspiration to the students for well-rounded development of their personality.
- To impart technical and need-based education by conducting elaborated trainingprograms.
- To shape and mold the personality of the future generation.
- To construct fertile ground for adapting to dire challenges.
- To cultivate the feeling of belongingness amongst the faction of engineers.