CYBER CORE

A FOOD DELIVERY APP

TITLE : CYBER CORE - A FOOD DELIVERY APP

PROBLEM DESCRIPTION :

In today's fast-paced lifestyle, individuals seek a seamless and efficient solution for their dining needs, leading to the development of Cyber Core Food – an innovative food delivery app designed to revolutionize the culinary experience. Cyber Core Food aims to simplify the food ordering process by offering a user-friendly interface that allows customers to effortlessly browse through an extensive array of dishes. Users can explore various types of cuisines and, specifically for a popular dish like biryani, access a comprehensive selection of biryani varieties from different restaurants. The app facilitates customization, enabling users to easily adjust the quantity of food items in their order or remove items as per their preferences.

Cyber Core Food goes beyond conventional food delivery platforms by incorporating a streamlined payment system, ensuring a secure and swift transaction process. Moreover, users can conveniently specify their delivery location, ensuring that the ordered meals reach them promptly. With the fusion of an intuitive design, diverse menu options, and seamless transaction capabilities, Cyber Core Food is poised to redefine the food delivery experience, offering a solution that caters to the evolving needs of modern consumers.

USERS OF THE APPLICATION:

Cyber Core Food is designed to cater to a diverse range of users, including:

Busy Professionals: Individuals with demanding work schedules who may not have the time to cook or dine out can rely on Cyber Core Food to conveniently order meals and have them delivered to their doorstep.

Students: Students, whether in college or high school, often juggle hectic academic schedules. Cyber Core Food provides them with a quick and efficient way to access a variety of food options without disrupting their study routines.

Families: Families looking for a convenient solution for mealtime, especially on busy days or when they want to enjoy a diverse range of cuisines at home, can use Cyber Core Food to simplify the process of ordering and receiving meals.

Food Enthusiasts: Those who appreciate exploring diverse culinary options and trying new dishes will find Cyber Core Food appealing. The app offers a wide array of choices, allowing food enthusiasts to discover and indulge in different cuisines and specialties.

Tech-Savvy Individuals: People who prefer the ease and convenience of mobile applications for various tasks will be drawn to Cyber Core Food's user friendly interface, making it accessible to individuals with varying levels of technological proficiency.

Social Gatherings and Events: Cyber Core Food can also be utilized by individuals organizing or attending social gatherings, parties, or events. It provides a hassle-free way to cater to diverse food preferences without the need for elaborate meal planning.

FRONTEND CODE:

INDEX.JS

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import {
  createBrowserRouter,
  createRoutesFromElements,
  Route,
  RouterProvider,
} from "react-router-dom";
import Home from "./page/Home";
import Menu from "./page/Menu";
import About from "./page/About";
import Contact from "./page/Contact";
import Login from "./page/login";
import ListMenu from "./page/ListMenu";
import Newproduct from "./page/Newproduct";
import Signup from "./page/Signup";
import { store } from "./redux/index";
import { Provider } from "react-redux";
import Cart from "./page/Cart";
import Success from "./page/Success";
import Cancel from "./page/Cancel";


const router = createBrowserRouter(
  createRoutesFromElements(

    <Route path="/" element={<App />}>
      <Route index element={<Home />} />
      {/* <Route path="menu" element={<Menu />} /> */}
      <Route path="menu/:filterby" element={<Menu />} />
      <Route path="ListMenu" element={<ListMenu/>}/>
      <Route path="about" element={<About />} />
      <Route path="contact" element={<Contact />} />
      <Route path="login" element={<Login />} />
      <Route path="newproduct" element={<Newproduct />} />
      <Route path="signup" element={<Signup />} />
      <Route path="cart" element={<Cart />} />
      <Route path="success" element={<Success/>}/>
      <Route path="cancel" element={<Cancel/>}/>
    </Route>

  )
);

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <Provider store={store}>
    <RouterProvider router={router} />
  </Provider>
);
reportWebVitals();
```

HEADER.JS

```
import React, { useState } from "react";
import { Link } from "react-router-dom";
import logo from "../assest/logo.png";
import { BsCartFill } from "react-icons/bs";
import { useDispatch, useSelector } from "react-redux";
import { logoutRedux } from "../redux/userSlice";
import { toast } from "react-hot-toast";
import { LiaUserCircle } from "react-icons/lia";

const Header = () => {
  const [showMenu, setShowMenu] = useState(false);
  const userData = useSelector((state) => state.user);
  const dispatch = useDispatch();

  const handleShowMenu = () => {
    setShowMenu((preve) => !preve);
  };

  const handleLogout = () => {
    dispatch(logoutRedux());
    toast("Logout successfully");
  };
  const cartItemNumber =
useSelector((state)=>state.product.cartItem)
  return (
    <header className="fixed bg-white bg-opacity-20 backdrop-blur-lg
drop-shadow-lg shadow-md w-full h-16 px-2 md:px-6 z-50 bg-white">
      {/* desktop */}

      <div className="flex items-center h-full justify-between">
        <Link to={""}>
          <div className="h-9">
            <img src={logo} alt="EveryDay Eats" className="h-full"
/>
          </div>
        </Link>

        <div className="flex items-center gap-8 md:gap-16">
          <nav className="gap-8 md:gap-12 text-base md:text-lg
hidden md:flex">
            <Link to={""}>Home</Link>
            <Link to={"listmenu"}>Menu</Link>
            <Link to={"about"}>About</Link>
            <Link to={"contact"}>Profile</Link>
          </nav>
          <div className="text-2xl text-slate-600 relative">
            <Link to={"cart"}>
              <BsCartFill />
              <div className="absolute -top-2 -right-3 text-white
bg-lime-900 h-4 w-4 rounded-full m-0 p-0 text-xs text-center ">
                {cartItemNumber.length}
              </div>
            </Link>
          </div>
          <div className=" text-slate-600" onClick={handleShowMenu}>
            <div className="text-4xl cursor-pointer w-8 h-8 rounded-
full overflow-hidden drop-shadow-md">
              {userData.image ? (
                <img src={userData.image} alt="User Profile"
className="h-full w-full" />
              ) : (
                <LiaUserCircle />
              )}
            </div>
            {showMenu && (
              <div className="absolute right-2 bg-white py-2 gap-2
rounded-lg shadow drop-shadow-md flex flex-col min-w-[120px] text-
center">
                {userData.email ===
process.env.REACT_APP_ADMIN_EMAIL && (
                  <Link to={"newproduct"} className="whitespace-
nowrap cursor-pointer px-2">
                    New product
                  </Link>
                )}

                {userData.image ? (
                  <p
                    className="cursor-pointer text-white px-2 bg-
lime-900"
                    onClick={handleLogout}
                  >
                    Logout ({userData.firstName}){" "}
                  </p>
                ) : (
                  <Link
                    to={"login"}
                    className="whitespace-nowrap cursor-pointer px-
2"
                  >
                    Login
                  </Link>
                )}
                <nav className="text-base md:text-lg flex flex-col
md:hidden">
                  <Link to={""} className="px-2 py-1">
                    Home
                  </Link>
                  <Link
                    to={"listmenu"}
                    className="px-2 py-1"
                  >
                    Menu
                  </Link>
                  <Link to={"about"} className="px-2 py-1">
                    About
                  </Link>
                  <Link to={"contact"} className="px-2 py-1">
                    Contact
                  </Link>
                </nav>
              </div>
            )}
          </div>
        </div>
      </div>

      {/* mobile */}
```

```
    </header>
  );
};

export default Header;
```

HOME.JS

```
import React, { useEffect, useRef, useState } from "react";
import { useSelector } from "react-redux";
import CardFeature from "../component/CardFeature";
import HomeCard from "../component/HomeCard";
import { GrPrevious, GrNext } from "react-icons/gr";
import FilterProduct from "../component/FilterProduct";
import AllProduct from "../component/AllProduct";
import { Link } from "react-router-dom";
import logo from "../assest/FRONT.png";
const Home = () => {
  const productData = useSelector((state) =>
state.product.productList);
  const homeProductCartList = productData.slice(1,5);
  const homeProductCartListVegetables = productData.filter(
    (el) => el.category === "Biryani" || "Fried Rice",
    []
  );
  const loadingArray = new Array(4).fill(null);
  const loadingArrayFeature = new Array(10).fill(null);

  const slideProductRef = useRef();
  const nextProduct = () => {
    slideProductRef.current.scrollLeft += 200;
  };
  const preveProduct = () => {
    slideProductRef.current.scrollLeft -= 200;
  };
  return (

    <div className="p-2 h-full min-h-screen w-full bg-blue-200
rounded-none bg-clip-padding backdrop-filter backdrop-blur-sm bg-
opacity-30 border border-gray-100 md:p-4"    >
      <div className="md:flex gap-4 py-2">
        <div className="md:w-1/2">
          <div className="flex gap-3 bg-slate-300 w-36 px-2 items-
center rounded-full">
            <p className="text-sm font-medium text-slate-900">Bike
Delivery</p>
            <img
```

```
              src="https://cdn-icons-
png.flaticon.com/512/2972/2972185.png"
              alt="Doorstep Delivery"
              className="h-7"
            />
          </div>
          <h2 className="text-4xl md:text-7xl font-bold py-3">
           Hungry?{" "}<br></br>
            <span className=" text-green-900 text-"> Order in
Seconds, Delivered in Minutes!</span>
          </h2>
          <p className="py-3 text-base "></p>
          <div className="p-6 rounded-lg text-gray-900">
  <h2 className="text-4xl   font-extrabold mb-4">
    <span className="text-yellow-500 ">SAVOR </span>
    <span className="text-blue-800">THE </span>
    <span className="text-indigo-700">FLAVOR, </span>
    <span className="text-green-500">SWIFTLY!</span>
  </h2>
  <h4>
  Cyber Core Foods, a renowned establishment nestled in the heart of
Coimbatore, is more than just a restaurant - it's an epicurean
haven. Our commitment is centered around crafting and delivering
culinary masterpieces within a generous 30-kilometer radius,
guaranteeing not just meals, but experiences, promptly and
efficiently to your doorstep.
  </h4>
</div>


<button className="mt-8 transition ease-in-out delay-100
hover:scale-110 font-bold bg-green-900 text-slate-200 px-4 py-2
rounded-md">
        <Link to="/listmenu" className="text-slate-200">
          Order Now
        </Link>
      </button>

          </div>
          <div
  className="relative h-44.5 w-50"
  style={{
    position: 'relative',
    overflow: 'hidden',
  }}
>
  {/* Blurred Background */}
  <div
```

```
    className="absolute p-5  ml-80 left-40 inset-4 bg-green-900
rounded-md bg-clip-padding backdrop-filter backdrop-blur-md bg-
opacity-40 "
    style={{
      position: 'absolute',
    }}
  ></div>

  {/* Image in front */}
  <img
    src={logo}
    alt="EveryDay Eats"
    className="ml-60 h-[550px] transition ease-in-out delay-
100  hover:scale-110  rounded-md"
    style={{
      position: 'relative',
      zIndex: 1,
    }}
  />
</div>
      </div>

       <div className="">
        <div className="flex w-full  items-center">
          <h2 className="font-bold  text-2xl text-slate-800 mb-4">
          </h2>
          <div className="ml-auto flex gap-4">
            <button
              onClick={preveProduct}
              className="bg-slate-300 hover:bg-slate-400 text-lg  p-
1 rounded"
            >
              <GrPrevious />
            </button>
            <button
              onClick={nextProduct}
              className="bg-slate-300 hover:bg-slate-400 text-lg p-1
rounded "
            >
              <GrNext />
            </button>
          </div>
        </div>

        <div
          className="flex pt-8 gap-5  overflow-scroll overflow-
hidden scrollbar-none scroll-smooth transition-all"
          ref={slideProductRef}
        >
          {homeProductCartListVegetables[0]
            ? homeProductCartListVegetables.map((el) => {
                return (
                  <CardFeature
                    key={el._id+"Fried Rice"}
                    id={el._id}
                    name={el.name}
                    category={el.category}
                    price={el.price}
                    image={el.image}
                  />
                );
              })
            : loadingArrayFeature.map((el,index) => (
                <CardFeature loading="Loading..."
key={index+"cartLoading"} />
              ))}
        </div>
      </div>


      <AllProduct heading={""}/>
    </div>

  );
};

export default Home;
```

MENU.JS

```
import React, { useEffect, useRef, useState } from "react";
import { useSelector } from "react-redux";
import CardFeature from "../component/CardFeature";
import HomeCard from "../component/HomeCard";
import { GrPrevious, GrNext } from "react-icons/gr";
import FilterProduct from "../component/FilterProduct";
import AllProduct from "../component/AllProduct";
import { Link } from "react-router-dom";
import logo from "../assest/logo12.png";

const Home = () => {
  const productData = useSelector((state) =>
state.product.productList);

  const homeProductCartListVegetables = productData.filter(
    (el) => el.category === "",
    []
  );
  const loadingArray = new Array(4).fill(null);
  const loadingArrayFeature = new Array(10).fill(null);

  const slideProductRef = useRef();
  const nextProduct = () => {
    slideProductRef.current.scrollLeft += 200;
  };
  const preveProduct = () => {
    slideProductRef.current.scrollLeft -= 200;
  };
  return (
    <div className="p-2 min-h-screen bg-blue-200 rounded-none bg-
clip-padding backdrop-filter backdrop-blur-sm bg-opacity-30 md:p-4">
      <AllProduct heading={""}/>
    </div>

  );
};

export default Home;
```

CART.JS

```
import React from "react";
import { useSelector } from "react-redux";
import CartProduct from "../component/cartProduct";
import emptyCartImage from "../assest/empty.gif"
import { toast } from "react-hot-toast";
import {loadStripe} from '@stripe/stripe-js';
import { useNavigate } from "react-router-dom";

const Cart = () => {
  const productCartItem = useSelector((state) =>
state.product.cartItem);
  const user = useSelector(state => state.user)
  const navigate = useNavigate()

  const totalPrice = productCartItem.reduce(
    (acc, curr) => acc + parseInt(curr.total),
    0
  );
  const totalQty = productCartItem.reduce(
    (acc, curr) => acc + parseInt(curr.qty),
    0
  );
  const handlePayment = async()=>{

      if(user.email){

          const stripePromise = await
loadStripe(process.env.REACT_APP_STRIPE_PUBLIC_KEY)
          const res = await
fetch(`${process.env.REACT_APP_SERVER_DOMIN}/create-checkout-
session`,{
            method : "POST",
            headers  : {
              "content-type" : "application/json"
            },
            body  : JSON.stringify(productCartItem)
          })
          if(res.statusCode === 500) return;

          const data = await res.json()
          console.log(data)

          toast("Redirect to payment Gateway...!")
          stripePromise.redirectToCheckout({sessionId : data})
      }
      else{
        toast("You have not Login!")
        setTimeout(()=>{
          navigate("/login")
        },1000)
      }

  }
  return (
    <>

    <div className="min-h-screen  items-center justify-center p-2
bg-blue-200 bg-opacity-30 backdrop-blur-lg drop-shadow-lg md:p-4">
        <h2 className="text-lg md:text-2xl font-bold text-slate-
800">
          Your Cart Items
        </h2>

        {productCartItem[0] ?
        <div className="my-4 flex gap-3">
          {/* display cart items  */}
          <div className="w-full grid gap-4 bg-white bg-opacity-20
backdrop-blur-lg rounded-lg drop-shadow-lg max-w-3xl ">
            {productCartItem.map((el) => {
```

```
       return (
        <CartProduct
         key={el._id}
         id={el._id}
         name={el.name}
         image={el.image}
         category={el.category}
         qty={el.qty}
         total={el.total}
         price={el.price}
        />
       );
      }))}
     </div>

     {/* total cart item  */}
     <div className="w-full  max-w-md  ml-auto">
      <h2 className="bg-blue-500 text-center rounded-lg text-
white p-2 text-lg">Summary</h2>
      <div className="flex w-full py-2 text-lg border-b">
       <p>Total Qty :</p>
       <p className="ml-auto w-32 font-bold">{totalQty}</p>
      </div>
      <div className="flex w-full py-2 text-lg border-b">
       <p>Total Price</p>
       <p className="ml-auto w-32 font-bold">
        <span className="text-red-500">₹</span> {totalPrice}
       </p>
      </div>
      <button className="bg-red-500 rounded-lg w-full text-lg
font-bold py-2 text-white" onClick={handlePayment}>
       Payment
      </button>
     </div>
    </div>

    :
    <>
     <div className="flex w-full justify-center items-center
flex-col">
      <img src={emptyCartImage} className="w-full mt-40 top-40
rounded-3xl max-w-sm"/>
      <p className="text-slate-500 text-3xl font-bold">Empty
Cart</p>
     </div>
    </>
   }
   </div>
```

```
    </>
  );
};

export default Cart;
```

SIGNUP.JS

```
import React, { useState } from "react";
import loginSignupImage from "../assest/login-animation.gif";
import { BiShow, BiHide } from "react-icons/bi";
import { Link, useNavigate } from "react-router-dom";
import { BsEmojiSmileUpsideDown } from "react-icons/bs";
import { ImagetoBase64 } from "../utility/ImagetoBase64";
import { toast } from "react-hot-toast";

function Signup() {
  const navigate = useNavigate();
  const [showPassword, setShowPassword] = useState(false);
  const [showConfirmPassword, setShowConfirmPassword] =
useState(false);
  const [data, setData] = useState({
    firstName: "",
    lastName: "",
    email: "",
    password: "",
    confirmPassword: "",
    address:"",
    image : ""
  });

  const handleShowPassword = () => {
    setShowPassword((preve) => !preve);
  };
  const handleShowConfirmPassword = () => {
    setShowConfirmPassword((preve) => !preve);
  };

  const handleOnChange = (e) => {
    const { name, value } = e.target;
    setData((preve) => {
      return {
        ...preve,
        [name]: value,
      };
    });
  };
  const handleUploadProfileImage = async(e)=>{
      const data = await ImagetoBase64(e.target.files[0])
      setData((preve)=>{
          return{
            ...preve,
            image : data
          }
      })
  }
  console.log(process.env.REACT_APP_SERVER_DOMIN)
  const handleSubmit = async(e) => {
    e.preventDefault();
    const { firstName, email, password, confirmPassword } = data;
    if (firstName && email && password && confirmPassword) {
      if (password === confirmPassword) {
          const fetchData = await
fetch(`${process.env.REACT_APP_SERVER_DOMIN}/signup`,{
            method : "POST",
            headers : {
              "content-type" : "application/json"
            },
            body : JSON.stringify(data)
          })
          console.log(data)
          const dataRes = await fetchData.json()
        // alert(dataRes.message);
        toast(dataRes.message)
        if(dataRes.alert){
          navigate("/login");
        }
      } else {
        alert("password and confirm password not equal");
      }
    } else {
      alert("Please Enter required fields");
    }
  };

  return (
    <div className="p-3 bg-blue-200  bg-opacity-30 backdrop-blur-lg
drop-shadow-lg  md:p-20">
      <div className="w-full rounded-lg bg-gradient-to-r from-
yellow-200 to-blue-200 bg-opacity-30 backdrop-blur-lg drop-shadow-
lg  rounded-lg max-w-sm bg-slate-300 m-auto flex  flex-col p-4">
        {/* <h1 className='text-center text-2xl font-bold'>Sign
up</h1> */}
        <div className="w-13 overflow-hidden  m-auto relative ">
```

```
          <img src={data.image ? data.image :  loginSignupImage}
className="w-full h-full" />

          <label htmlFor="profileImage">
            <div className=" bottom-0 h-1/3  bg-slate-500 bg-
opacity-50 w-full text-center cursor-pointer">
              <p className=" text-sm p-3 text-white">Upload</p>
            </div>
            <input type={"file"} id="profileImage" accept="image/*"
className="hidden" onChange={handleUploadProfileImage}/>
          </label>
        </div>

        <form className="w-full py-3 flex flex-col"
onSubmit={handleSubmit}>
          <label htmlFor="firstName">First Name</label>
          <input
            type={"text"}
            id="firstName"
            name="firstName"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-1
rounded focus-within:outline-blue-300"
            value={data.firstName}
            onChange={handleOnChange}
          />

          <label htmlFor="lastName">Last Name</label>
          <input
            type={"text"}
            id="lastName"
            name="lastName"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-1
rounded focus-within:outline-blue-300"
            value={data.lastName}
            onChange={handleOnChange}
          />
          <label htmlFor="phoneno">Mobile No</label>
          <input
            type={"tel"}
            id="phoneno"
            name="phoneno"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-1
rounded focus-within:outline-blue-300"
            value={data.phoneno}
            onChange={handleOnChange}
          />
          <label htmlFor="email">Email</label>
          <input
```

```
            type={"email"}
            id="email"
            name="email"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-1
rounded focus-within:outline-blue-300"
            value={data.email}
            onChange={handleOnChange}
          />

          <label htmlFor="address">Address</label>
          <textarea
            rows={2}
            type={"address"}
            id="address"
            name="address"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-2
resize-none rounded focus-within:outline-blue-300"
            value={data.address}
            onChange={handleOnChange}
          />

          <label htmlFor="password">Password</label>
          <div className="flex px-2 py-1 bg-slate-200 rounded mt-1
mb-2 focus-within:outline focus-within:outline-blue-300">
            <input
              type={showPassword ? "text" : "password"}
              id="password"
              name="password"
              className=" w-full bg-slate-200 border-none outline-
none "
              value={data.password}
              onChange={handleOnChange}
            />
            <span
              className="flex text-xl cursor-pointer"
              onClick={handleShowPassword}
            >
              {showPassword ? <BiShow /> : <BiHide />}
            </span>
          </div>

          <label htmlFor="confirmpassword">Confirm Password</label>
          <div className="flex px-2 py-1 bg-slate-200 rounded mt-1
mb-2  focus-within:outline focus-within:outline-blue-300">
            <input
              type={showConfirmPassword ? "text" : "password"}
              id="confirmpassword"
              name="confirmPassword"
              className=" w-full bg-slate-200 border-none outline-
none "
              value={data.confirmPassword}
              onChange={handleOnChange}
            />
            <span
              className="flex text-xl cursor-pointer"
              onClick={handleShowConfirmPassword}
            >
              {showConfirmPassword ? <BiShow /> : <BiHide />}
            </span>
          </div>

          <button className="w-full transition ease-in-out delay-
100  hover:scale-110 bg-gradient-to-r from-green-400 to-blue-500
hover:from-pink-500 hover:to-yellow-500 bg-opacity-50 backdrop-blur-
lg drop-shadow-lg  max-w-[150px] m-auto  bg-blue-300 hover:bg-green-
800 cursor-pointer  text-white hover:text-white text-xl font-medium
text-center py-1.5 rounded-full mt-4">
            Sign up
          </button>
        </form>
        <p className="text-left text-sm mt-2">
          Already have account ?{" "}
          <Link to={"/login"} className="text-red-500 underline">
            Login
          </Link>
        </p>
      </div>
    </div>
  );
}

export default Signup;
```

LOGIN.JS

```
import React, { useState } from "react";
import loginSignupImage from "../assest/login-animation.gif";
import { BiShow, BiHide } from "react-icons/bi";
import { Link } from "react-router-dom";
import { BsEmojiSmileUpsideDown } from "react-icons/bs";
import { toast } from "react-hot-toast"
import { useNavigate } from "react-router-dom";
import { useDispatch, useSelector } from "react-redux";
import { loginRedux } from "../redux/userSlice";


const Login = () => {
  const [showPassword, setShowPassword] = useState(false);
  const [data, setData] = useState({
    email: "",
    password: "",
  });
  const navigate = useNavigate()
  const userData = useSelector(state => state)
  const dispatch = useDispatch()
  const handleShowPassword = () => {
    setShowPassword((preve) => !preve);
  };

  const handleOnChange = (e) => {
    const { name, value } = e.target
    setData((preve) => {
      return {
        ...preve,
        [name]: value
      }
    })
  }

  const handleSubmit = async (e) => {
    e.preventDefault()
    const { email, password } = data
    if (email && password) {
      const fetchData = await
fetch(`${process.env.REACT_APP_SERVER_DOMIN}/login`, {
        method: "POST",
        headers: {
          "content-type": "application/json"
        },
        body: JSON.stringify(data)
      })

      const dataRes = await fetchData.json()
      console.log(dataRes)

      toast(dataRes.message)

      if (dataRes.alert) {
        dispatch(loginRedux(dataRes))
        setTimeout(() => {
          navigate("/")
        }, 1000);
      }

      console.log(userData)
    }
    else {
      alert("Please Enter required fields")
    }
  }

  return (
    <div className=" min-h-screen flex bg-blue-200 items-center
justify-center p-2  bg-opacity-30 backdrop-blur-lg drop-shadow-lg
md:p-70">
      <div className="w-full rounded-lg bg-gradient-to-r from-
yellow-200 to-blue-200 bg-opacity-30 backdrop-blur-lg drop-shadow-
lg  bg-slate-300  max-w-sm bg-white m-auto flex  flex-col p-5">
        {/* <h1 className='text-center text-2xl font-bold'>Sign
up</h1> */}
        <div className="w-13  m-auto">
          <img src={loginSignupImage} className="w-full" />
        </div>

        <form className="w-full py-3 flex flex-col"
onSubmit={handleSubmit}>
          <label htmlFor="email">Email</label>
          <input
            type={"email"}
            id="email"
            name="email"
            className="mt-1 mb-2 w-full bg-slate-200 px-2 py-1
rounded focus-within:outline-blue-300"
            value={data.email}
            onChange={handleOnChange}

          />

          <label htmlFor="password">Password</label>
          <div className="flex px-2 py-1 bg-slate-200 rounded mt-1
mb-2 focus-within:outline focus-within:outline-blue-300">
            <input
              type={showPassword ? "text" : "password"}
              id="password"
              name="password"
              className=" w-full bg-slate-200 border-none outline-
none "
              value={data.password}
              onChange={handleOnChange}
            />
            <span
              className="flex text-xl cursor-pointer"
              onClick={handleShowPassword}
            >
              {showPassword ? <BiShow /> : <BiHide />}
            </span>
          </div>

          <button className="w-full bg-gradient-to-r from-green-400
to-blue-500 hover:from-pink-500 hover:to-yellow-500 transition ease-
in-out delay-100  hover:scale-110  max-w-[150px] m-auto  bg-green-
900 hover:bg-teal-500 cursor-pointer  text-white text-xl font-medium
text-center py-1 rounded-full mt-4">
            Login
          </button>
        </form>
        <p className="text-left text-sm mt-2">
          Don't  have account ?{" "}
          <Link to={"/signup"} className="text-red-500 underline">
            Sign Up
          </Link>
        </p>
      </div>
    </div>
  )
}

export default Login
```

PROFILE.JS

```
import React from 'react';
import toast from 'react-hot-toast';
import { useSelector } from 'react-redux';
import { useNavigate } from 'react-router-dom';

const UserProfile = () => {
  const user = useSelector((state) => state.user);
  const navigate = useNavigate();

  if (!user.email) {
    toast('You have not logged in!');
    setTimeout(() => {
      navigate('/login');
    }, 10);
  }

  return (
    <div className="bg-blue-200 min-h-screen bg-opacity-30 flex
justify-center items-center backdrop-blur-lg shadow-sm p-4">
      <div className="mb-80 justify-center max-w-3xl w-full lg:max-
w-full lg:flex">
```

```
        <div className="h-80 bg-opacity-30 flex backdrop-blur-sm
shadow-lg lg:h-auto lg:w-80 flex-none bg-cover rounded-t lg:rounded-
t-none lg:rounded-l text-center overflow-hidden">
          <img src={user.image} alt="User" className="h-full w-full
object-cover" />
        </div>
        <div className="border-r bg-blue-200 bg-opacity-30 flex
backdrop-blur-sm shadow-lg border-b border-l border-gray-400
lg:border-l-0 lg:border-t lg:border-gray-400 bg-white rounded-b
lg:rounded-b-none lg:rounded-r p-6 flex flex-col justify-between
leading-normal">
          <div className="mb-8">
            <p className="text-2xl text-gray-600 flex items-center">
              {/* Additional greetings or messages can be inserted
here */}
              Hi, {user.firstName}!
            </p>
            <div className="text-gray-900 font-bold text-2xl mb-
4"></div>
            <p className="text-gray-700 py-2 text-base">
              <span className="font-bold">Name:</span>
{user.firstName} {user.lastName}
            </p>
            <p className="text-gray-700 py-2 text-base">
              <span className="font-bold">Email:</span> {user.email}
            </p>
            <p className="text-gray-700 py-2 text-base">
              <span className="font-bold">Address:</span>
{user.address}
            </p>
            <p className="text-gray-700 py-2 text-base">
              <span className="font-bold">Phone:</span>
{user.phoneno}
            </p>
          </div>
          <div className="flex items-center">
            <img className="w-12 h-12 rounded-full mr-4"
src={user.image} alt="Avatar" />
            <div className="text-sm">
              <p className="text-gray-900 leading-
none">{user.firstName} {user.lastName}</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};
```

```
export default UserProfile;
```

ABOUT.JS

```
import React from 'react';

const About = () => {
  const redirectToMail = (email) => {
    window.location.href = `mailto:${email}`;
  };

  const redirectToPhone = (phoneNumber) => {
    window.location.href = `tel:${phoneNumber}`;
  };

  return (
    <div className="flex flex-col bg-blue-200 min-h-screen bg-
opacity-30 justify-center items-center backdrop-blur-sm">
      <div className="w-1/3 p-4 justify-center m-2 bg-blue-200 bg-
opacity-30 backdrop-filter backdrop-blur-sm rounded-md">
        <h1 className="text-3xl font-bold mb-4 text-center">CYBER
CORE FOOD</h1>
        <section className="text-center mb-4">
          <h2 className="text-lg font-bold mb-2">About</h2>
          <div className="mb-2">
            <p>
              Cyber Core Foods is a renowned restaurant situated in
Sai Karan Nagar, Singanallur, Coimbatore.
              Our dedication lies in delivering exquisite meals
within a 30km radius, ensuring swift and timely delivery.
            </p>
          </div>
        </section>
      </div>
      <div className="w-1/3 p-4 m-2 bg-blue-200 bg-opacity-30
backdrop-filter backdrop-blur-sm rounded-md">
        <section className="text-center mb-4">
          <h2 className="text-lg font-bold mb-2">Contact</h2>
          <p className="mb-2">Email:</p>
          <ul className="list-disc list-inside mb-2">
            <li>
              <span className="text-blue-500 cursor-pointer"
onClick={() =>
redirectToMail('abarnaananthan17@gmail.com')}>abarnaananthan@gmail.c
om</span>
            </li>
            <li>
```

```
              <span className="text-blue-500 cursor-pointer"
onClick={() =>
redirectToMail('nikitta2411@gmail.com')}>nikitta2411@gmail.com</span
>
            </li>
          </ul>
          <p className="mb-2">Mobile:</p>
          <ul className="list-disc list-inside mb-2">
            <li>
              <span className="text-blue-500 cursor-pointer"
onClick={() => redirectToPhone('+91 8870416782')}>+91
8870416700</span>
            </li>
            <li>
              <span className="text-blue-500 cursor-pointer"
onClick={() => redirectToPhone('+91 9043045347')}>+91
9043045347</span>
            </li>
          </ul>
          <p className="text-center mt-2 text-left">
            Address : Cyber Core Foods, Sai Karan Nagar,
Singanallur, Coimbatore
          </p>
        </section>
      </div>
    </div>
  );
};

export default About;
```

.env

```
REACT_APP_SERVER_DOMIN = http://localhost:8080
REACT_APP_ADMIN_EMAIL = nikitta2411@gmail.com
REACT_APP_STRIPE_PUBLIC_KEY
="pk_test_51OKBrOSCfxpYiAKSbhEo8UfP50hbSqAwb5zUt58wk8wLLeCvQ1OQgVP0C
euuMR2qOA4TQKD9uYAY4wJA6OV3w1Km00bFqLXU0m"
```

BACKEND CODE:

INDEX.JS

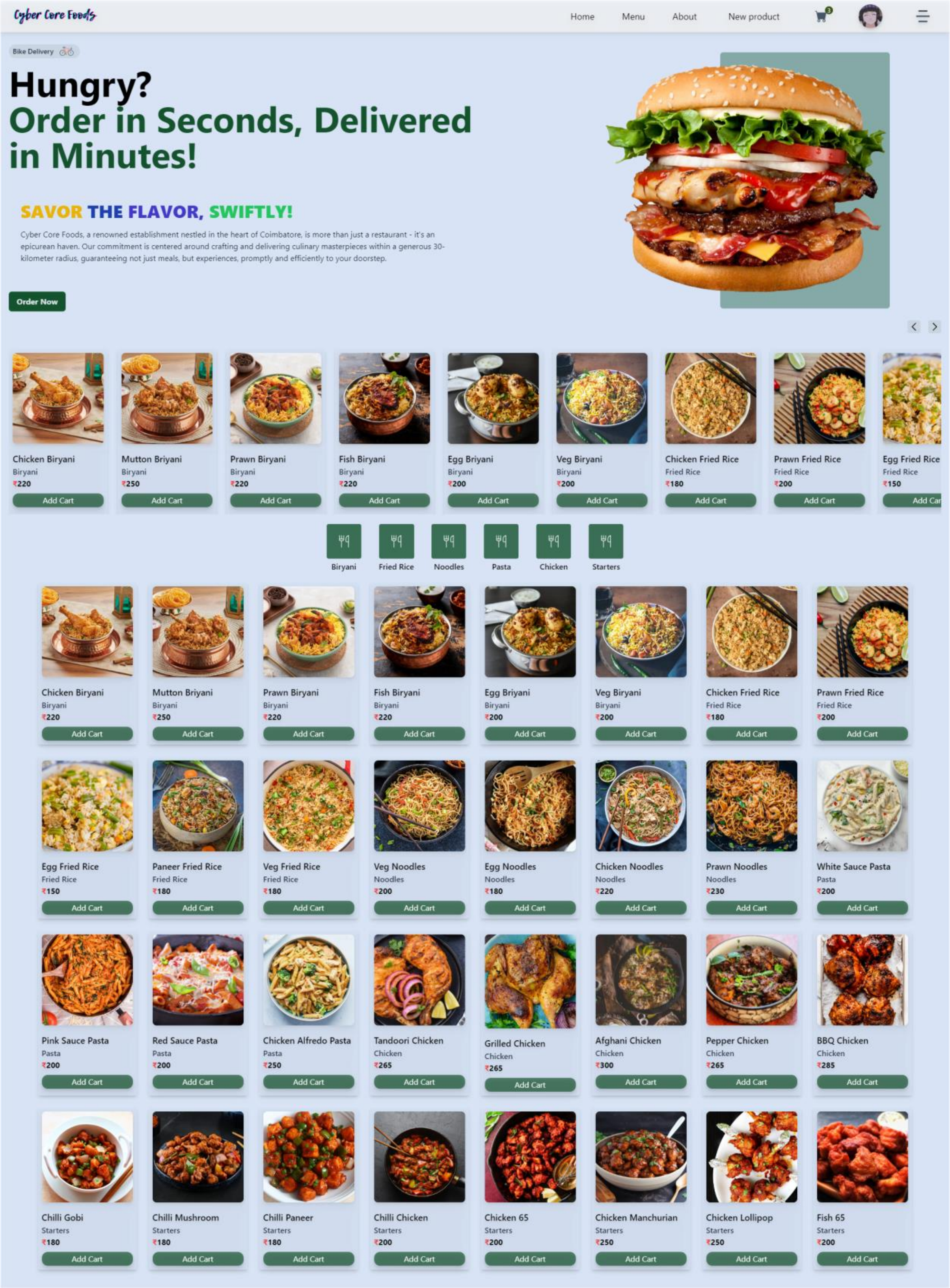
```
const express = require("express");
const cors = require("cors");
const mongoose = require("mongoose");
const dotenv = require("dotenv").config();
const Stripe = require('stripe')
const app = express();
app.use(cors());
app.use(express.json({ limit: "10mb" }));

const PORT = process.env.PORT || 8080;

//mongodb connection
mongoose.set("strictQuery", false);
mongoose
  .connect(process.env.MONGODB_URL)
  .then(() => console.log("Connected to Databse"))
  .catch((err) => console.log(err));

//schema
const userSchema = mongoose.Schema({
  firstName: String,
  lastName: String,
  email: {
    type: String,
    unique: true,
  },
  password: String,
  confirmPassword: String,
  address: String,
  phoneno: Number,
  image: String,
});

//
const userModel = mongoose.model("user", userSchema);

//api
app.get("/", (req, res) => {
  res.send("Server is running");
});

//sign up
app.post("/signup", async (req, res) => {
  // console.log(req.body);
  const { email } = req.body;

  userModel.findOne({ email: email }, (err, result) => {
    // console.log(result);
    console.log(err);
    if (result) {
      res.send({ message: "Email id is already register", alert:
false });
    } else {
      const data = userModel(req.body);
      const save = data.save();
      res.send({ message: "Successfully sign up", alert: true });
    }
  });
});

//api login
app.post("/login", (req, res) => {
  // console.log(req.body);
  const { email } = req.body;
  userModel.findOne({ email: email }, (err, result) => {
    if (result) {
      const dataSend = {
        _id: result._id,
        firstName: result.firstName,
        lastName: result.lastName,
        email: result.email,
        phoneno: result.phoneno,
        address: result.address,
        image: result.image,
      };
      console.log(dataSend);
      res.send({
        message: "Login is successfully",
        alert: true,
        data: dataSend,
      });
    } else {
      res.send({
        message: "Email is not available, please sign up",
        alert: false,
      });
    }
  });
});

//product section

const schemaProduct = mongoose.Schema({
  name: String,
  category: String,
  image: String,
  price: String,
  description: String,
});
const productModel = mongoose.model("product", schemaProduct)
```

```
//save product in data
//api
app.post("/uploadProduct", async (req, res) => {
  // console.log(req.body)
  const data = await productModel(req.body)
  const datasave = await data.save()
  res.send({ message: "Upload successfully" })
})

//
app.get("/product", async (req, res) => {
  const data = await productModel.find({})
  res.send(JSON.stringify(data))
})

/*****payment getWay */
console.log(process.env.STRIPE_SECRET_KEY)


const stripe = new Stripe(process.env.STRIPE_SECRET_KEY)

app.post("/create-checkout-session", async (req, res) => {

  try {
    const params = {
      submit_type: 'pay',
      mode: "payment",
      payment_method_types: ['card'],
      billing_address_collection: "auto",
      shipping_options: [{ shipping_rate:
"shr_1OKDlXSCfxpYiAKS3r0vNyxN" }],

      line_items: req.body.map((item) => {
        return {
          price_data: {
            currency: "inr",
            product_data: {
              name: item.name,
              // images : [item.image]
            },
            unit_amount: item.price * 100,
          },
          adjustable_quantity: {
            enabled: true,
            minimum: 1,
          },
          quantity: item.qty
        }
      }),

      success_url: `${process.env.FRONTEND_URL}/success`,
      cancel_url: `${process.env.FRONTEND_URL}/cancel`,

    }


    const session = await stripe.checkout.sessions.create(params)
    // console.log(session)
    res.status(200).json(session.id)
  }
  catch (err) {
    res.status(err.statusCode || 500).json(err.message)
  }

})


//server is ruuning
app.listen(PORT, () => console.log("server is running at port : " +
PORT));
```

.env

```
MONGODB_URL = "mongodb://localhost:27017"
STRIPE_SECRET_KEY
="sk_test_51OKBrOSCfxpYiAKSYHqZxKraDLb1ye1JBgSdrdBMNK4JLitcnkrGl1yL1
60eMZ5Ub1NAE6O0OWicuqnRGUUceNLZ00thHgkKZP"
FRONTEND_URL = "http://localhost:3000"
```

OUTPUT:



Hungry?
Order in Seconds, Delivered in Minutes!

SAVOR THE FLAVOR, SWIFTLY!

Cyber Core Foods, a renowned establishment nestled in the heart of Coimbatore, is more than just a restaurant - it's an epicurean haven. Our commitment is centered around crafting and delivering culinary masterpieces within a generous 30-kilometer radius, guaranteeing not just meals, but experiences, promptly and efficiently to your doorstep.

Chicken Biryani
Biryani
₹220

Mutton Briyani
Biryani
₹250

Prawn Biryani
Biryani
₹220

Fish Biryani
Biryani
₹220

Egg Bryani
Biryani
₹200

Veg Biryani
Biryani
₹200

Chicken Fried Rice
Fried Rice
₹180

Prawn Fried Rice
Fried Rice
₹200

Egg Fried Rice
Fried Rice
₹150

Chicken Biryani
Biryani
₹220

Mutton Briyani
Biryani
₹250

Prawn Biryani
Biryani
₹220

Fish Biryani
Biryani
₹220

Egg Bryani
Biryani
₹200

Veg Biryani
Biryani
₹200

Chicken Fried Rice
Fried Rice
₹180

Prawn Fried Rice
Fried Rice
₹200

Egg Fried Rice
Fried Rice
₹150

Paneer Fried Rice
Fried Rice
₹180

Veg Fried Rice
Fried Rice
₹180

Veg Noodles
Noodles
₹200

Egg Noodles
Noodles
₹180

Chicken Noodles
Noodles
₹220

Prawn Noodles
Noodles
₹230

White Sauce Pasta
Pasta
₹200

Pink Sauce Pasta
Pasta
₹200

Red Sauce Pasta
Pasta
₹200

Chicken Alfredo Pasta
Pasta
₹250

Tandoori Chicken
Chicken
₹265

Grilled Chicken
Chicken
₹265

Afghani Chicken
Chicken
₹300

Pepper Chicken
Chicken
₹265

BBQ Chicken
Chicken
₹285

Chilli Gobi
Starters
₹180

Chilli Mushroom
Starters
₹180

Chilli Paneer
Starters
₹180

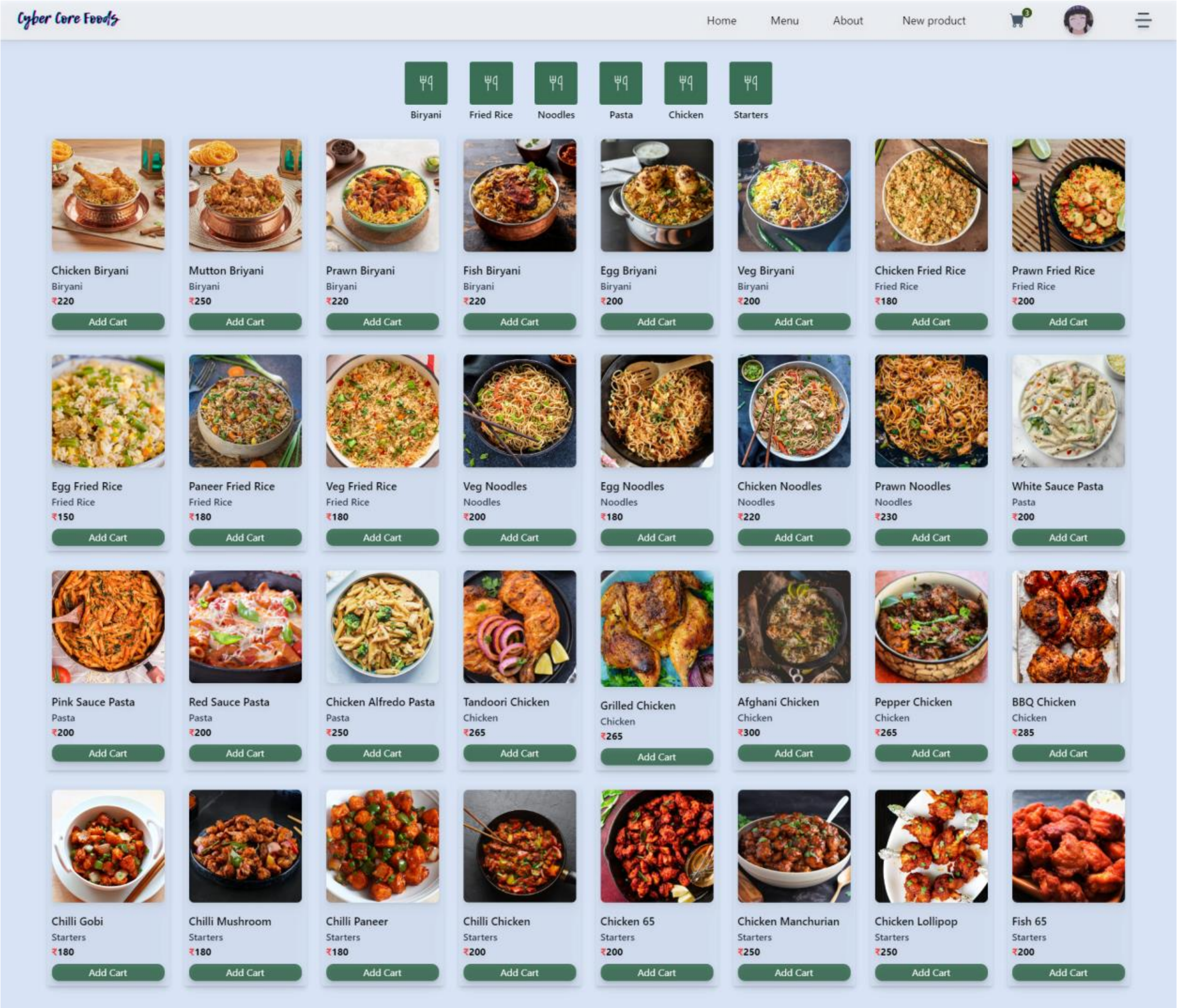
Chilli Chicken
Starters
₹200

Chicken 65
Starters
₹200

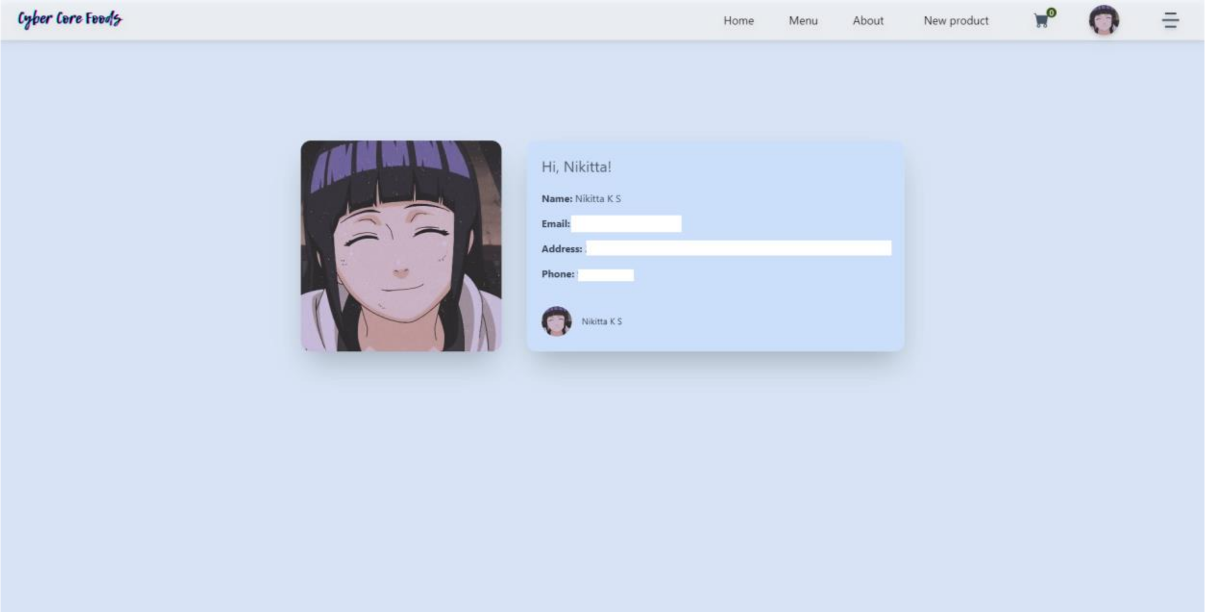
Chicken Manchurian
Starters
₹250

Chicken Lollipop
Starters
₹250

Fish 65
Starters
₹200



Chicken Briyani	Biryani	₹220
Mutton Briyani	Biryani	₹250
Prawn Briyani	Biryani	₹220
Fish Briyani	Biryani	₹220
Egg Briyani	Biryani	₹200
Veg Biryani	Biryani	₹200
Chicken Fried Rice	Fried Rice	₹180
Prawn Fried Rice	Fried Rice	₹200
Egg Fried Rice	Fried Rice	₹150
Paneer Fried Rice	Fried Rice	₹180
Veg Fried Rice	Fried Rice	₹180
Veg Noodles	Noodles	₹200
Egg Noodles	Noodles	₹180
Chicken Noodles	Noodles	₹220
Prawn Noodles	Noodles	₹230
White Sauce Pasta	Pasta	₹200
Pink Sauce Pasta	Pasta	₹200
Red Sauce Pasta	Pasta	₹200
Chicken Alfredo Pasta	Pasta	₹250
Tandoori Chicken	Chicken	₹265
Grilled Chicken	Chicken	₹265
Afghani Chicken	Chicken	₹300
Pepper Chicken	Chicken	₹265
BBQ Chicken	Chicken	₹285
Chilli Gobi	Starters	₹180
Chilli Mushroom	Starters	₹180
Chilli Paneer	Starters	₹180
Chilli Chicken	Starters	₹200
Chicken 65	Starters	₹200
Chicken Manchurian	Starters	₹250
Chicken Lollipop	Starters	₹250
Fish 65	Starters	₹200



Hi, Nikitta!

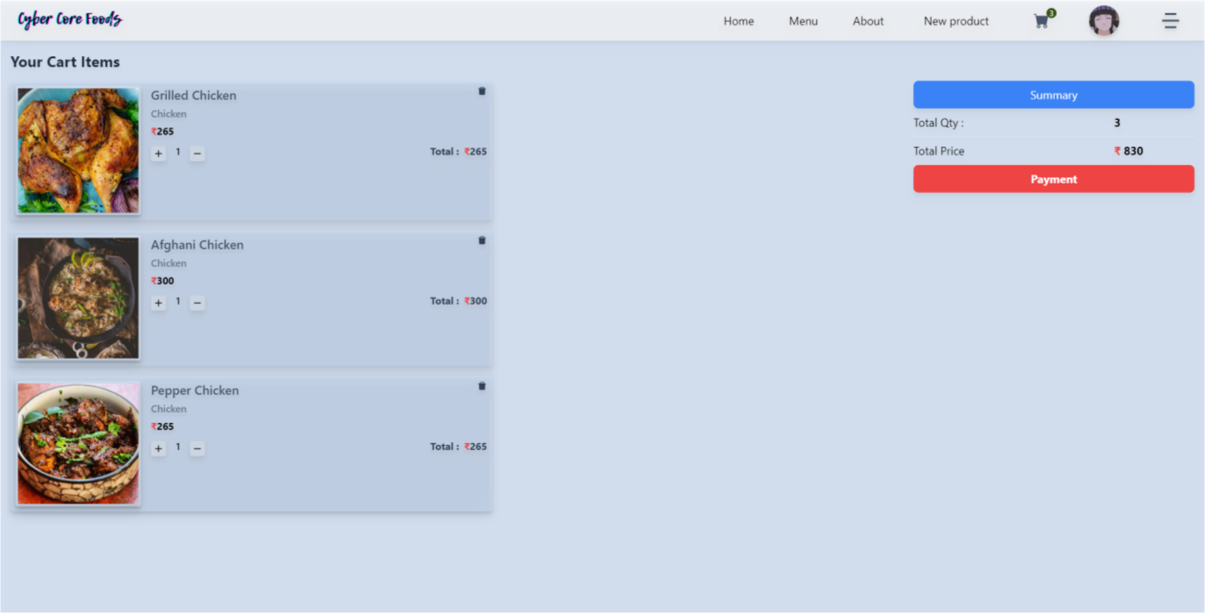
Name: Nikitta K S

Email: [redacted]

Address: [redacted]

Phone: [redacted]

Nikitta K S



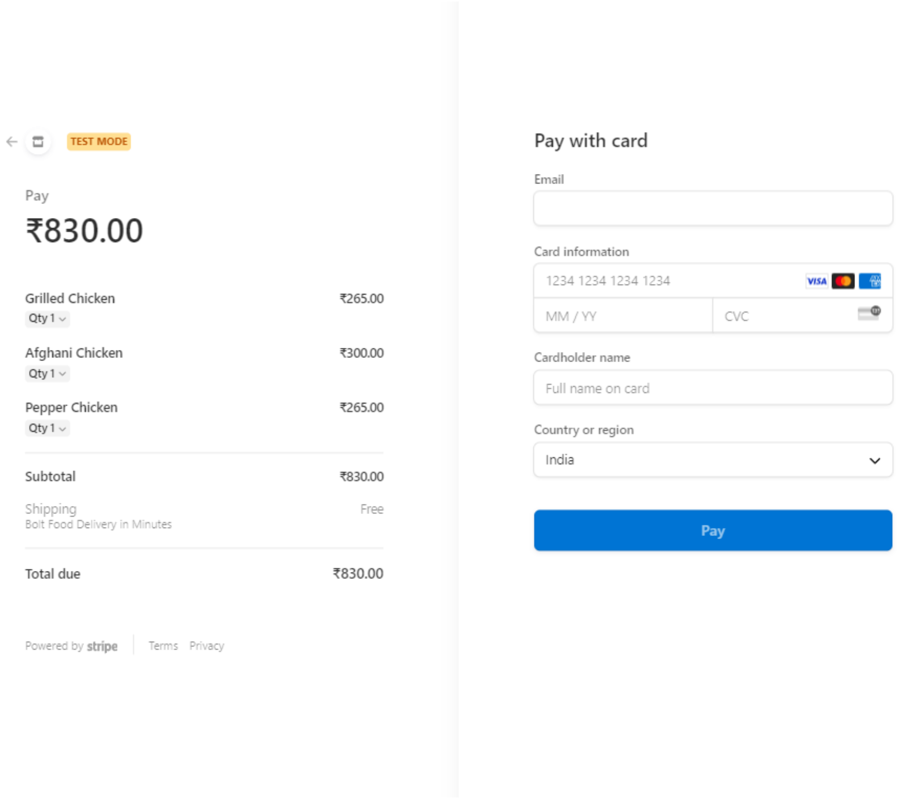
Your Cart Items

Grilled Chicken
Chicken
₹265
+ 1 –
Total : ₹265

Afghani Chicken
Chicken
₹300
+ 1 –
Total : ₹300

Pepper Chicken
Chicken
₹265
+ 1 –
Total : ₹265

Total Qty : 3
Total Price ₹ 830



Pay
₹830.00

Grilled Chicken Qty 1	₹265.00
Afghani Chicken Qty 1	₹300.00
Pepper Chicken Qty 1	₹265.00
Subtotal	₹830.00
Shipping Bolt Food Delivery in Minutes	Free
Total due	₹830.00

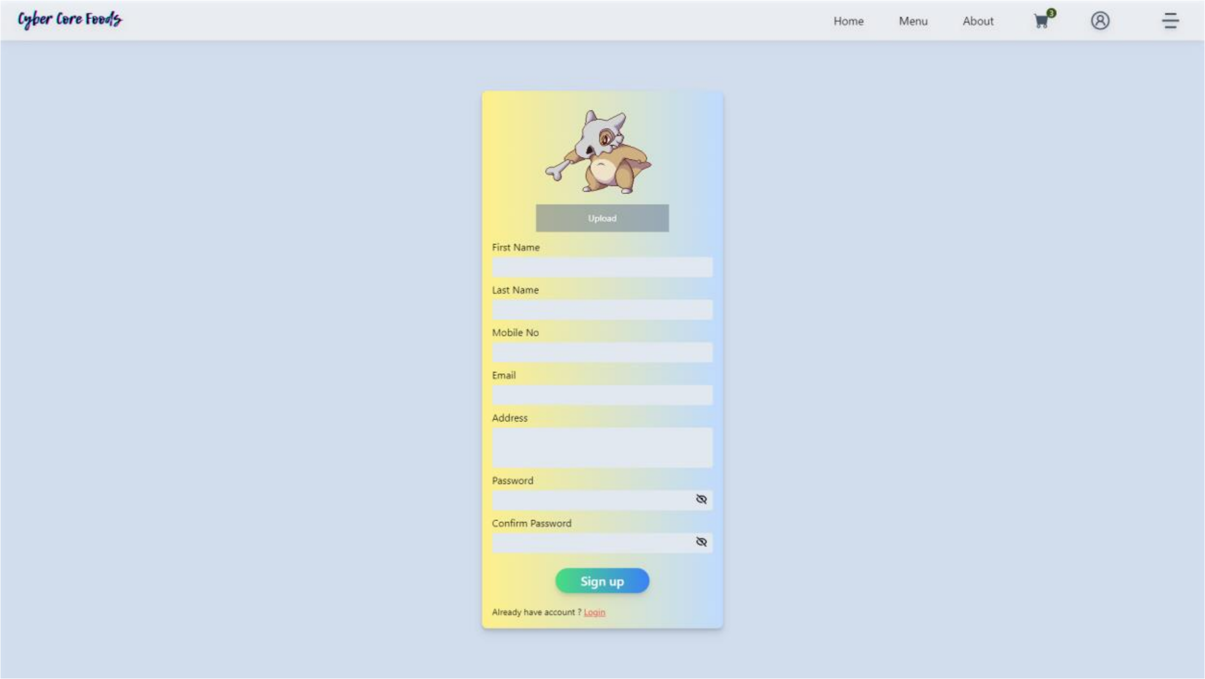
Pay with card

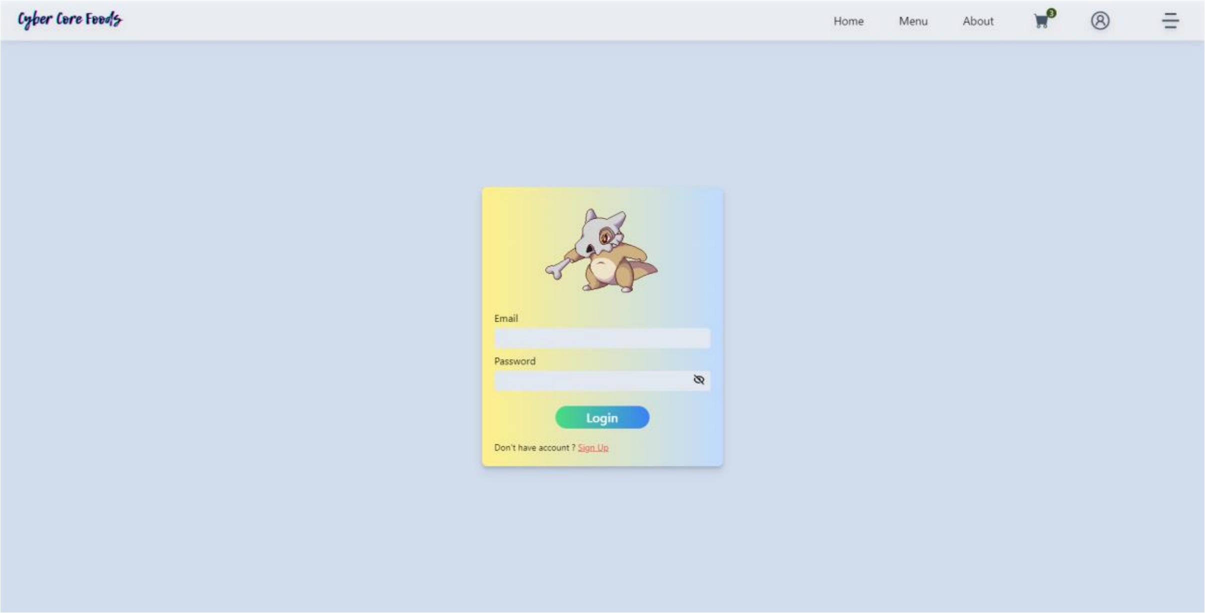
Email

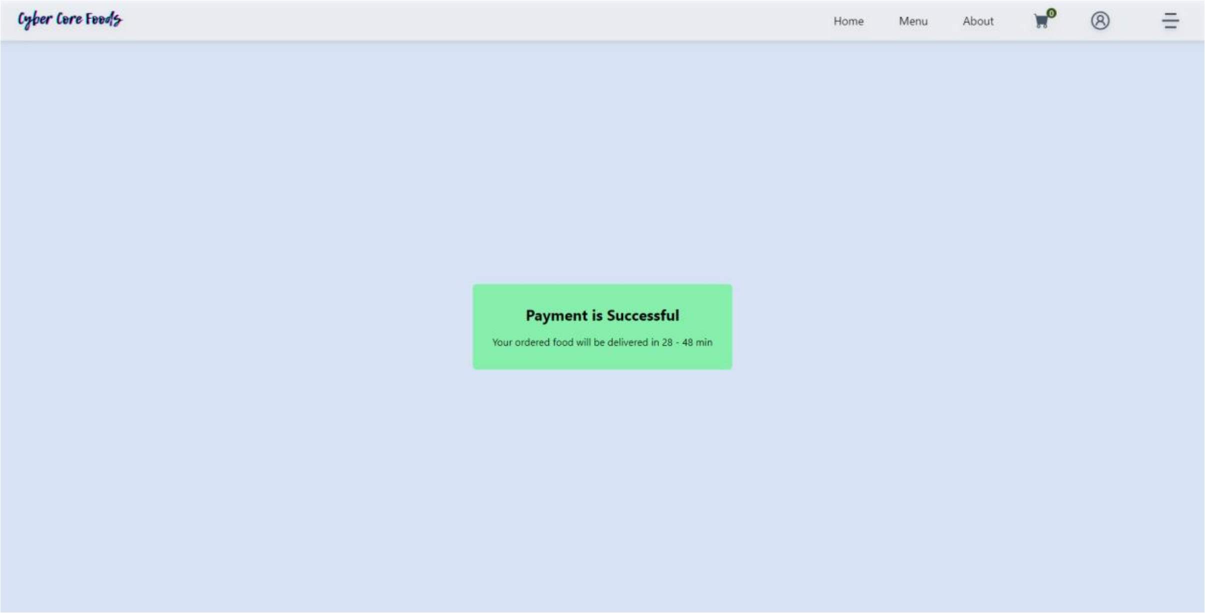
Card information
1234 1234 1234 1234
MM / YY
CVC

Cardholder name
Full name on card

Country or region
India

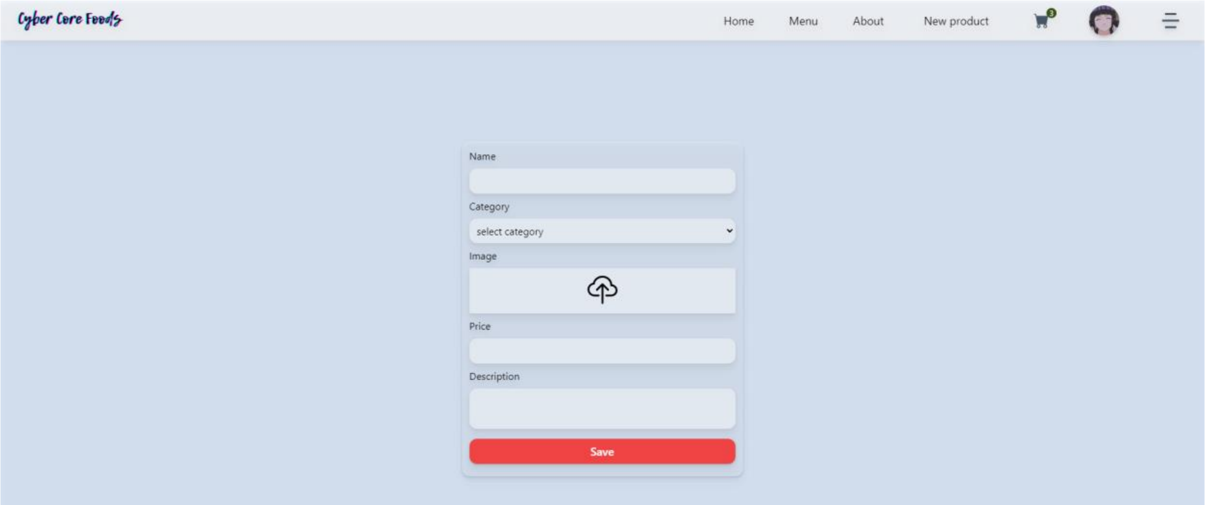

Cyber Core Foods
Home
Menu
About
Upload
First Name
Last Name
Mobile No
Email
Address
Password
Confirm Password
Sign up
Already have account ? Login

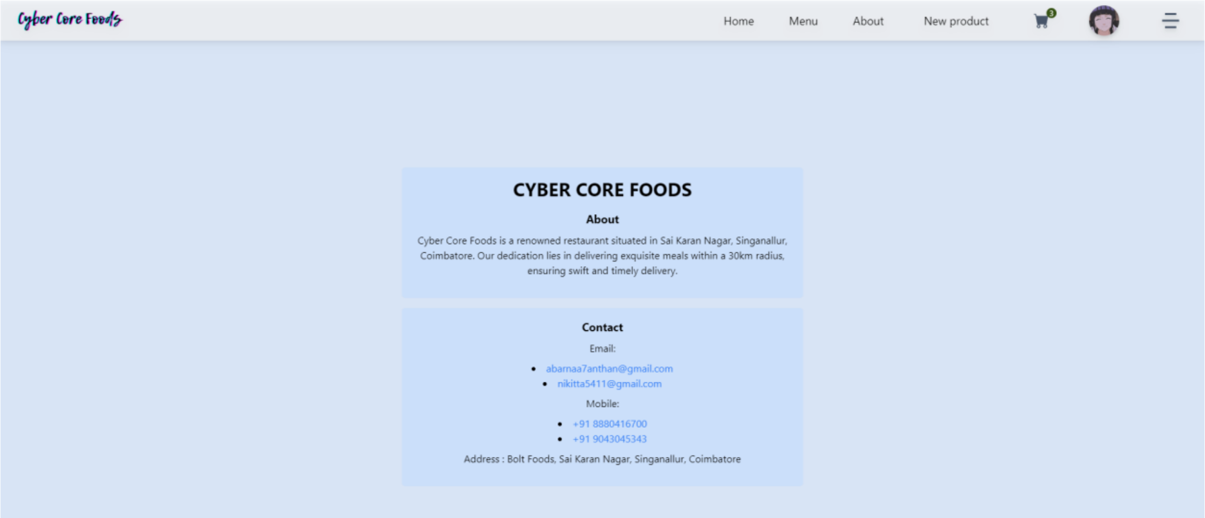

Cyber Core Foods
Home
Menu
About
Email
Password
Login
Don't have account ? Sign Up



Payment is Successful

Your ordered food will be delivered in 28 - 48 min





CYBER CORE FOODS

About

Cyber Core Foods is a renowned restaurant situated in Sai Karan Nagar, Singanallur, Coimbatore. Our dedication lies in delivering exquisite meals within a 30km radius, ensuring swift and timely delivery.

Contact

Email:

- abarnaa7anthan@gmail.com
- nikitta5411@gmail.com

Mobile:

- +91 8880416700
- +91 9043045343

Address : Bolt Foods, Sai Karan Nagar, Singanallur, Coimbatore