

COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

**Mobile Application Development semester project**

**Documentation for Food-Panda- Food App**

**Irfan Khan FA21-BSE-027**

**Shahzaib FA21-BSE-034**

Submitted To:

**Sir Jawad Khan**

Date: 29 december-2023

Contents

[1. Introduction 3](#_Toc154840026)

[1.1 Purpose and Overview 3](#_Toc154840027)

[1.2 Target Audience 3](#_Toc154840028)

[2. Technology Stack 4](#_Toc154840029)

[2.1 Frontend 4](#_Toc154840030)

[2.2 Backend 4](#_Toc154840031)

[2.3 Additional Libraries/Tools 4](#_Toc154840032)

[3. Authentication Module 4](#_Toc154840033)

[3.1 Description 4](#_Toc154840034)

[3.2 Implementation 4](#_Toc154840035)

[3.3 Key Features 4](#_Toc154840036)

[4. Home Screen 5](#_Toc154840037)

[4.1 Description 5](#_Toc154840038)

[4.2 Navigation 5](#_Toc154840039)

[4.3 Screenshots 6](#_Toc154840040)

[5. Services Module 6](#_Toc154840041)

[5.1 Description 6](#_Toc154840042)

[5.2 Service Types 6](#_Toc154840043)

[6. Pick-up Screen 7](#_Toc154840044)

[6.1 Description 7](#_Toc154840045)

[6.3 Visuals 7](#_Toc154840046)

[7. Cart Screen 8](#_Toc154840047)

[7.1 Description 8](#_Toc154840048)

[7.3 Screenshot 8](#_Toc154840049)

[9. Code Snippets 9](#_Toc154840050)

[9.1 app.js 9](#_Toc154840051)

[9.2 Login.js 10](#_Toc154840052)

[9.3 Register.js 14](#_Toc154840053)

[9.4 home.js 18](#_Toc154840054)

[9.5 Pick-Up 24](#_Toc154840055)

[9. 6 Cart 30](#_Toc154840056)

[9.7 Orderscreen.js 38](#_Toc154840057)

[9.8 profile screen.js 39](#_Toc154840058)

[9.9 dress item services 40](#_Toc154840059)

[9.10 carousel.js 44](#_Toc154840060)

[9.11 Services.js 44](#_Toc154840061)

[9.12 Cart Reducer.js 46](#_Toc154840062)

[9.13 Product reducer.js 47](#_Toc154840063)

[9.14 Stack Navigator 48](#_Toc154840064)

# 1. Introduction

## 1.1 Purpose and Overview

Food-Panda is a mobile application meticulously crafted to revolutionize the way users experience food delivery services. Boasting a seamless user interface and robust features, the app is designed to overcome the common hurdles associated with traditional food delivery platforms.

## 1.2 Target Audience

Food-Panda is catered to individuals with a penchant for convenience and a craving for diverse culinary experiences. It specifically targets busy professionals, families, and students who are in search of a hassle-free and efficient solution for their food cravings. Whether it's a quick lunch at the office, a family dinner, or a late-night study session, Food-Panda aims to provide a delightful and streamlined food delivery experience for everyone.

# 2. Technology Stack

## 2.1 Frontend

* **Framework:** React Native
* **State Management:** Redux

## 2.2 Backend

* **Database:** Firebase Firestore
* **Authentication:** Firebase Authentication

## 2.3 Additional Libraries/Tools

* React Native Paper
* React Native Scroll View
* React Native Location
* React Native Safe Area Context
* Async Storage

# 3. Authentication Module

## 3.1 Description

The authentication module ensures a secure and personalized experience for users. Leveraging the Redux, the app manages user authentication seamlessly.

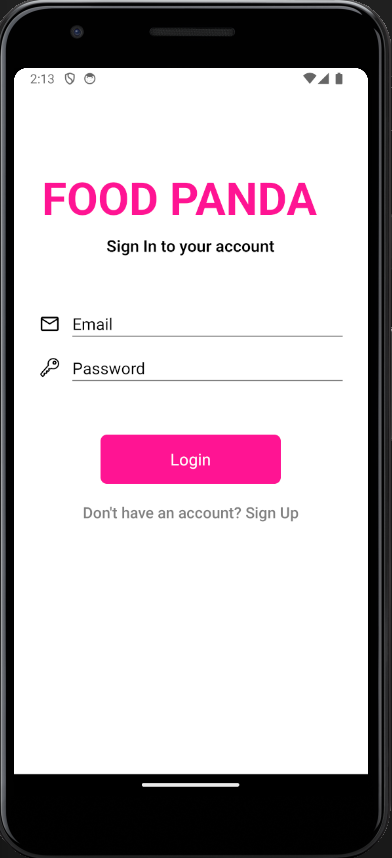
## 3.2 Implementation

Explain the implementation details of user authentication, including the flow and security measures in place.

## 3.3 Key Features

* Secure user authentication.
* Integration with Firebase Authentication for enhanced security.

A screen shot of a phone

Description automatically generated

# 4. Home Screen

## 4.1 Description

The home screen serves as the central hub for navigation. Provide your location after you allow the application to access than show to menu.

## 4.2 Navigation

* Services: Explore food services.
* Profile: User account and settings
* Selection: Select items to order.

## 4.3 Screenshots

# 5. Services Module

## 5.1 Description

The services module introduces users to the available Food services, offering a variety of options to suit different needs.

## 5.2 Service Types

Egg Rice, Chicken Rice, Chilly Paneer and Soup.

# 6. Pick-up Screen

## 6.1 Description

When a user selects items, detailed information is presented, highlighting the unique features and benefits of each option.

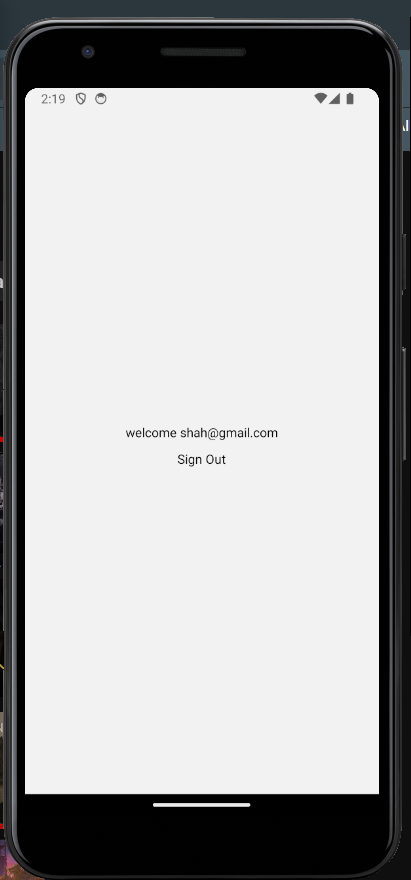
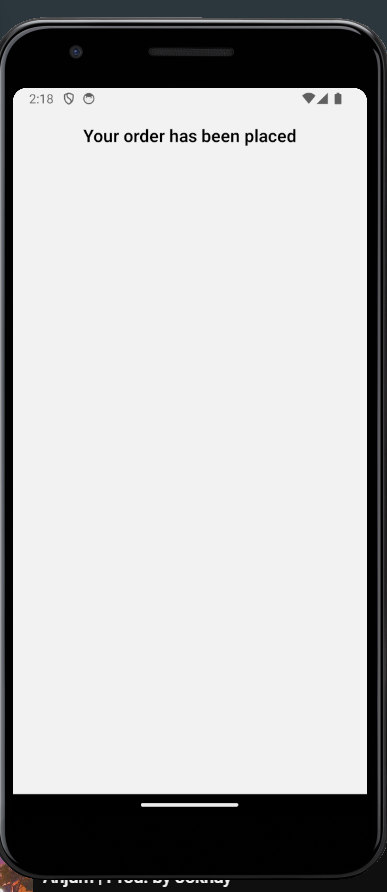
## A screenshot of a phone Description automatically generated6.3 Visuals

# 7. Cart Screen

## 7.1 Description

The Cart Screen simplifies the process of confirming the order.

## 7.3 Screenshot



# 9. Code Snippets

## 9.1 app.js

// App.js

import { StatusBar } from 'expo-status-bar';

import { StyleSheet, Text, View } from 'react-native';

import HomeScreen from './screens/HomeScreen';

import { Provider } from 'react-redux';

import store from './store';

import StackNavigator from './StackNavigator';

export default function App() {

  return (

    <Provider store={store}>

      <StackNavigator/>

      <StatusBar />

    </Provider>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    marginTop:25

  },

});

## 9.2 Login.js

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

import {

  StyleSheet,

  Text,

  View,

  SafeAreaView,

  KeyboardAvoidingView,

  TextInput,

  Pressable,

  ActivityIndicator,

} from "react-native";

import { Ionicons } from "@expo/vector-icons";

import { MaterialCommunityIcons } from "@expo/vector-icons";

import { useNavigation } from "@react-navigation/native";

import { signInWithEmailAndPassword } from "firebase/auth";

import { auth } from "../firebase";

const LoginScreen = () => {

  const [email, setEmail] = useState("");

  const [loading, setLoading] = useState(false);

  const [password, setPassword] = useState("");

  const navigation = useNavigation();

  useEffect(() => {

    setLoading(true);

    const unsubscribe = auth.onAuthStateChanged((authUser) => {

      if (!authUser) {

        setLoading(false);

      }

      if (authUser) {

        navigation.replace("Home");

      }

    });

    return unsubscribe;

  }, []);

  const login = () => {

    signInWithEmailAndPassword(auth, email, password).then((userCredential) => {

      console.log("user credential", userCredential);

      const user = userCredential.user;

      console.log("user details", user);

    });

  };

  return (

    <SafeAreaView style={styles.container}>

      {loading ? (

        <View style={styles.loadingContainer}>

          <Text style={styles.loadingText}>Loading</Text>

          <ActivityIndicator size="large" color={"red"} />

        </View>

      ) : (

        <KeyboardAvoidingView style={styles.contentContainer}>

          <View style={styles.header}>

            <Text style={styles.headerText}>FOOD PANDA  </Text>

            <Text style={styles.subHeaderText}>Sign In to your account</Text>

          </View>

          <View style={styles.inputContainer}>

            <View style={styles.inputWrapper}>

              <MaterialCommunityIcons

                name="email-outline"

                size={24}

                color="black"

              />

              <TextInput

                placeholder="Email"

                value={email}

                onChangeText={(text) => setEmail(text)}

                placeholderTextColor="black"

                style={styles.input}

              />

            </View>

            <View style={styles.inputWrapper}>

              <Ionicons name="key-outline" size={24} color="black" />

              <TextInput

                value={password}

                onChangeText={(text) => setPassword(text)}

                secureTextEntry={true}

                placeholder="Password"

                placeholderTextColor="black"

                style={styles.input}

              />

            </View>

            <Pressable

              onPress={login}

              style={styles.loginButton}

            >

              <Text style={styles.loginButtonText}>Login</Text>

            </Pressable>

            <Pressable onPress={() => navigation.navigate("Register")} style={styles.signUpText}>

              <Text style={styles.signUpButtonText}>

                Don't have an account? Sign Up

              </Text>

            </Pressable>

          </View>

        </KeyboardAvoidingView>

      )}

    </SafeAreaView>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: "white",

    alignItems: "center",

    padding: 10,

  },

  loadingContainer: {

    flex: 1,

    alignItems: "center",

    justifyContent: "center",

  },

  loadingText: {

    marginRight: 10,

  },

  contentContainer: {

    flex: 1,

  },

  header: {

    justifyContent: "center",

    alignItems: "center",

    marginTop: 100,

  },

  headerText: {

    fontSize: 50,

    color: "#ff1493",

    fontWeight: "bold",

  },

  subHeaderText: {

    fontSize: 18,

    marginTop: 8,

    fontWeight: "600",

  },

  inputContainer: {

    marginTop: 50,

  },

  inputWrapper: {

    flexDirection: "row",

    alignItems: "center",

  },

  input: {

    fontSize: 18,

    borderBottomWidth: 1,

    borderBottomColor: "gray",

    marginLeft: 13,

    width: 300,

    marginVertical: 10,

  },

  loginButton: {

    width: 200,

    backgroundColor: "#ff1493",

    padding: 15,

    borderRadius: 7,

    marginTop: 50,

    marginLeft: "auto",

    marginRight: "auto",

  },

  loginButtonText: {

    fontSize: 18,

    textAlign: "center",

    color: "white",

  },

  signUpText: {

    marginTop: 20,

  },

  signUpButtonText: {

    textAlign: "center",

    fontSize: 17,

    color: "gray",

    fontWeight: "500",

  },

});

export default LoginScreen;

## 9.3 Register.js

import {

  StyleSheet,

  Text,

  View,

  Pressable,

  TextInput,

  SafeAreaView,

  KeyboardAvoidingView,

  Alert

} from "react-native";

import { Feather } from '@expo/vector-icons';

import { Ionicons } from "@expo/vector-icons";

import React, { useState } from "react";

import { MaterialCommunityIcons } from "@expo/vector-icons";

import { useNavigation } from "@react-navigation/native";

import { createUserWithEmailAndPassword } from "firebase/auth";

import { auth, db } from "../firebase";

import { doc, setDoc } from "firebase/firestore";

const RegisterScreen = () => {

    const [email,setEmail] = useState("");

    const [password,setPassword] = useState("");

    const [phone,setPhone] = useState("");

    const navigation = useNavigation();

    const register = () => {

      if(email === "" || password === "" || phone === ""){

        Alert.alert(

          "Invalid Details",

          "Please fill all the details",

          [

            {

              text: "Cancel",

              onPress: () => console.log("Cancel Pressed"),

              style: "cancel"

            },

            { text: "OK", onPress: () => console.log("OK Pressed") }

          ],

          { cancelable: false }

        );

      }

      createUserWithEmailAndPassword(auth,email,password).then((userCredential) => {

        console.log("user credential",userCredential);

        const user = userCredential.\_tokenResponse.email;

        const myUserUid = auth.currentUser.uid;

        setDoc(doc(db,"users",`${myUserUid}`),{

          email:user,

          phone:phone

        })

      })

    }

  return (

    <SafeAreaView

      style={{

        flex: 1,

        backgroundColor: "white",

        alignItems: "center",

        padding: 10,

      }}

    >

      <KeyboardAvoidingView>

        <View

          style={{

            justifyContent: "center",

            alignItems: "center",

            marginTop: 100,

          }}

        >

          <Text style={{ fontSize: 20, color: "#662d91", fontWeight: "bold" }}>

            Register

          </Text>

          <Text style={{ fontSize: 18, marginTop: 8, fontWeight: "600" }}>

            Create a new Account

          </Text>

        </View>

        <View style={{ marginTop: 50 }}>

          <View style={{ flexDirection: "row", alignItems: "center" }}>

            <MaterialCommunityIcons

              name="email-outline"

              size={24}

              color="black"

            />

            <TextInput

              placeholder="Email"

              value={email}

              onChangeText={(text) => setEmail(text)}

              placeholderTextColor="black"

              style={{

                fontSize: email ? 18 : 18,

                borderBottomWidth: 1,

                borderBottomColor: "gray",

                marginLeft: 13,

                width: 300,

                marginVertical: 10,

              }}

            />

          </View>

          <View style={{ flexDirection: "row", alignItems: "center" }}>

            <Ionicons name="key-outline" size={24} color="black" />

            <TextInput

              value={password}

              onChangeText={(text) => setPassword(text)}

              secureTextEntry={true}

              placeholder="Password"

              placeholderTextColor="black"

              style={{

                fontSize: password ? 18 : 18,

                borderBottomWidth: 1,

                borderBottomColor: "gray",

                marginLeft: 13,

                width: 300,

                marginVertical: 20,

              }}

            />

          </View>

          <View style={{ flexDirection: "row", alignItems: "center" }}>

          <Feather name="phone" size={24} color="black" />

            <TextInput

              value={phone}

              onChangeText={(text) => setPhone(text)}

              placeholder="Phone No"

              placeholderTextColor="black"

              style={{

                fontSize: password ? 18 : 18,

                borderBottomWidth: 1,

                borderBottomColor: "gray",

                marginLeft: 13,

                width: 300,

                marginVertical: 10,

              }}

            />

          </View>

          <Pressable

          onPress={register}

            style={{

              width: 200,

              backgroundColor: "#318CE7",

              padding: 15,

              borderRadius: 7,

              marginTop: 50,

              marginLeft: "auto",

              marginRight: "auto",

            }}

          >

            <Text style={{ fontSize: 18, textAlign: "center", color: "white" }}>

              Register

            </Text>

          </Pressable>

          <Pressable onPress={() => navigation.goBack()} style={{ marginTop: 20 }}>

            <Text

              style={{

                textAlign: "center",

                fontSize: 17,

                color: "gray",

                fontWeight: "500",

              }}

            >

              Already have a account? Sign in

            </Text>

          </Pressable>

        </View>

      </KeyboardAvoidingView>

    </SafeAreaView>

  );

};

export default RegisterScreen;

const styles = StyleSheet.create({});

## 9.4 home.js

import {

    StyleSheet,

    Text,

    View,

    SafeAreaView,

    Alert,

    Pressable,

    Image,

    TextInput,

    ScrollView

} from 'react-native'

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

import \* as Location from "expo-location";

import { MaterialIcons } from '@expo/vector-icons';

import { Feather } from '@expo/vector-icons';

import Carousel from '../components/Carousel';

import Services from '../components/Services';

import DressItem from '../components/DressItem';

import { useDispatch, useSelector } from 'react-redux';

import { getProducts } from '../ProductReducer';

import { useNavigation } from "@react-navigation/native";

import { collection, getDoc, getDocs } from "firebase/firestore";

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

const HomeScreen = () => {

    const cart = useSelector((state) => state.cart.cart);

    const total = cart.map((item) => item.quantity \* item.price).reduce((curr, prev) => curr + prev, 0);

    const navigation = useNavigation();

    console.log(cart);

    const [displayCurrentAddress, setdisplayCurrentAddress] = useState("we are loading your location");

    const [locationServicesEnabled, setlocalServicesEnabled] = useState(false);

    useEffect(() => {

        checkIfLocationEnabled();

        getCurrentLocation();

    }, [])

    const checkIfLocationEnabled = async () => {

        let enabled = await Location.hasServicesEnabledAsync();

        if (!enabled) {

            Alert.alert(

                "Location Service not enabled",

                "Please enable the location service",

                [

                    {

                        text: 'Cancel',

                        onPress: () => console.log('Cancel Pressed'),

                        style: 'cancel',

                    },

                    { text: 'OK', onPress: () => console.log('OK Pressed') },

                ],

                { cancelable: false }

            );

        } else {

            setlocalServicesEnabled(enabled);

        }

    }

    const getCurrentLocation = async () => {

        let { status } = await Location.requestForegroundPermissionsAsync();

        if (status !== "granted") {

            Alert.alert(

                "Permission denied",

                "Allow the app to use the location services",

                [

                    {

                        text: 'Cancel',

                        onPress: () => console.log('Cancel Pressed'),

                        style: 'cancel',

                    },

                    { text: 'OK', onPress: () => console.log('OK Pressed') },

                ]);

        }

        const { coords } = await Location.getCurrentPositionAsync();

        // console.log(coords);

        if (coords) {

            const { latitude, longitude } = coords;

            let response = await Location.reverseGeocodeAsync({

                latitude,

                longitude

            })

            // console.log(response);

            for (let item of response) {

                let address = `${item.name} ${item.city} ${item.postalCode}`;

                setdisplayCurrentAddress(address);

            }

        }

    };

    const product = useSelector((state) => state.product.product);

    const dispatch = useDispatch();

    useEffect(() => {

      if (product.length > 0) return;

      const fetchProducts = async () => {

        const colRef = collection(db,"types");

        const docsSnap = await getDocs(colRef);

        docsSnap.forEach((doc) => {

          item.push(doc.data());

        });

        item?.map((service) => dispatch(getProducts(service)));

      };

      fetchProducts();

    }, []);

    console.log(product);

    // console.log(product)

    const services = [

        {

            id: "0",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/ry3c3f518z10t4olu4l7 ",

            name: "Paneer 65",

            quantity: 1,

            price: 210,

        },

        {

            id: "11",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/druwjzmfmz7qvepq3bkr",

            name: "Chilly Chicken (Boneless)",

            quantity: 0,

            price: 110,

        },

        {

            id: "12",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/fsitbray4gq1kxcndqqx",

            name: "Spl Veg Biryani",

            quantity: 0,

            price: 140,

        },

        {

            id: "13",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/byonwwb8mzxyqluxlqpq",

            name: "Chilly Paneer",

            quantity: 0,

            price: 160,

        },

        {

            id: "14",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/x0jegvlf4h7wrgaaqdqi",

            name: "Chicken 65",

            quantity: 0,

            price: 130,

        },

        {

            id: "15",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/akmx533z73jjbq8avy6v",

            name: "Chicken Fried Rice",

            quantity: 0,

            price: 110,

        },

        {

            id: "16",

            image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/lv6jl9qdscekjmwkxm9l",

            name: "Egg Fried Rice",

            quantity: 0,

            price: 100,

        },

    ];

    return (

        <>

            <ScrollView style={{ backgroundColor: "#F0F0F0", flex: 1, marginTop: 50 }}>

                {/\* Location and profile bar \*/}

                <View style={{ flexDirection: "row", alignItems: "center", padding: 10 }}>

                    <MaterialIcons name="location-on" size={30} color="#fd5c63" />

                    <View>

                        <Text style={{ fontSize: 18, fontWeight: "600" }}>Home</Text>

                        <Text>{displayCurrentAddress}</Text>

                    </View>

                    <Pressable onPress={() => navigation.navigate("Profile")}

                    style={{ marginLeft: "auto", marginRight: 7 }} >

                        <Image

                        style={{ width: 40, height: 40, borderRadius: 20 }}

                            source={{

                                uri: "https://yt3.ggpht.com/hu807DZYWt8SwtGVUsLJ3loDPy-SGpkren2wcinFGJwfM5da2fnNiA9G7wE6U4k0kbTmhtfs\_3Y=s88-c-k-c0x00ffffff-no-rj"

                            }} />

                    </Pressable>

                </View>

                {/\* Search Bar Starts from here \*/}

                <View

                    style={{

                        padding: 10,

                        margin: 10,

                        flexDirection: "row",

                        alignItems: "center",

                        justifyContent: "space-between",

                        borderWidth: 0.8,

                        borderColor: "#C0C0C0",

                        borderRadius: 7

                    }}>

                    <TextInput placeholder="Search for items or more" />

                    <Feather name="search" size={24} color="#fd5c63" />

                </View>

                {/\* Image Carousel \*/}

                <Carousel />

                {/\* Services  \*/}

                <Services />

                {/\* Render all the Products \*/}

                {

                    services.map((item) => (

                        <DressItem item={item} />

                    )

                    )

}

            </ScrollView>

            {total === 0 ? (

                null

            ) : (

                <Pressable

                style={{

                    backgroundColor: "#088F8F",

                    padding: 10,

                    marginBottom: 40,

                    margin: 15,

                    borderRadius: 7,

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between"

                }}>

                <View>

                    <Text style={{fontSize:17, fontWeight:"600", color:"white"}}>{cart.length} item | $ {total}</Text>

                    <Text style={{fontSize:12, fontWeight:"400", color:"white", marginVertical:6}}>Extra Charges might be Apply</Text>

                </View>

                <Pressable onPress={() => navigation.navigate("PickUpScreen")}>

                    <Text style={{fontSize:17, fontWeight:"600", color:"white"}}>Proceed to Pickup!</Text>

                </Pressable>

            </Pressable>

            )}

        </>

    )

}

export default HomeScreen

const styles = StyleSheet.create({})

## 9.5 Pick-Up

import {

    StyleSheet,

    Text,

    View,

    SafeAreaView,

    TextInput,

    Pressable,

    ScrollView,

    Alert

 } from 'react-native'

import React, { useState } from 'react'

import HorizontalDatepicker from '@awrminkhodaei/react-native-horizontal-datepicker'

import { useSelector } from 'react-redux';

import { useNavigation } from '@react-navigation/native';

const PickUpScreen = () => {

    const cart = useSelector((state) => state.cart.cart);

    const total = cart.map((item) => item.quantity \* item.price).reduce((curr, prev) => curr + prev, 0);

    const [selectDate, setSelectedDate] = useState("");

    const [selectedTime, setSelectedTime] = useState([]);

    const [delivery, setDelivery] = useState([]);

    const deliveryTime = [

        {

            id: "0",

            name: "2-3 Days",

        },

        {

            id: "1",

            name: "3-4 Days",

        },

        {

            id: "2",

            name: "4-5 Days",

        },

        {

            id: "3",

            name: "5-6 Days",

        },

        {

            id: "4",

            name: "Tommorrow",

        },

    ];

    const times = [

        {

            id: "0",

            time: "11:00 PM",

        },

        {

            id: "1",

            time: "12:00 PM",

        },

        {

            id: "2",

            time: "1:00 PM",

        },

        {

            id: "2",

            time: "2:00 PM",

        },

        {

            id: "4",

            time: "3:00 PM",

        },

        {

            id: "5",

            time: "4:00 PM",

        },

    ];

    const navigation = useNavigation();

    const proceedToCart = () =>{

        if(!selectDate || !selectedTime || !delivery){

            Alert.alert(

                "Empty or Invalid",

                "Please Select ",

                [

                {

                  text: 'Cancel',

                  onPress: () => console.log('Cancel Pressed'),

                  style: 'cancel',

                },

                {text: 'OK', onPress: () => console.log('OK Pressed')},

              ]);

        }

        if(selectDate && selectedTime && delivery){

           navigation.replace("CartScreen")

        }

    }

    return (

        <>

        <SafeAreaView>

            <Text style={{ marginTop: 25, fontSize: 16, fontWeight: "500", marginHorizontal: 10 }}>

                Enter Address

            </Text>

            <TextInput

                style={{

                    padding: 40,

                    borderColor: "gray",

                    borderWidth: 0.8,

                    paddingVertical: 80,

                    borderRadius: 9,

                    margin: 10

                }} />

            <Text style={{ marginTop: 25, fontSize: 16, fontWeight: "500", marginHorizontal: 10 }}>

                Pick Up Date

            </Text>

            <HorizontalDatepicker

                mode="gregorian"

                startDate={new Date('2023-09-20')}

                endDate={new Date('2024-01-15')}

                initialSelectedDate={new Date('2020-08-22')}

                onSelectedDateChange={(date) => setSelectedDate(date)}

                selectedItemWidth={170}

                unselectedItemWidth={38}

                itemHeight={38}

                itemRadius={10}

                selectedItemTextStyle={styles.selectedItemTextStyle}

                unselectedItemTextStyle={styles.selectedItemTextStyle}

                selectedItemBackgroundColor="#222831"

                unselectedItemBackgroundColor="#ececec"

                flatListContainerStyle={styles.flatListContainerStyle}

            />

            <Text style={{ marginTop: 25, fontSize: 16, fontWeight: "500", marginHorizontal: 10 }}>

                Select Time

            </Text>

            <ScrollView horizontal showsHorizontalScrollIndicator={false}>

                {times.map((item, index) => (

                    <Pressable

                        key={index}

                        onPress={() => setSelectedTime(item.time)}

                        style={

                            selectedTime.includes(item.time) ?

                                {

                                    margin: 10,

                                    borderRadius: 7,

                                    padding: 15,

                                    borderColor: "red",

                                    borderWidth: 0.7

                                } :

                                {

                                    margin: 10,

                                    borderRadius: 7,

                                    padding: 15,

                                    borderColor: "gray",

                                    borderWidth: 0.7

                                }

                        }

                    >

                        <Text> {item.time}</Text>

                    </Pressable>

                ))}

            </ScrollView>

            <Text style={{ marginTop: 25, fontSize: 16, fontWeight: "500", marginHorizontal: 10 }}>

                Delivery Date

            </Text>

            <ScrollView horizontal showsHorizontalScrollIndicator={false}>

                {deliveryTime.map((item, i) => (

                    <Pressable

                        onPress={() => setDelivery(item.name)}

                        key={i}

                        style={

                            delivery.includes(item.name) ?

                                {

                                    margin: 10,

                                    borderRadius: 7,

                                    padding: 15,

                                    borderColor: "red",

                                    borderWidth: 0.7

                                } :

                                {

                                    margin: 10,

                                    borderRadius: 7,

                                    padding: 15,

                                    borderColor: "gray",

                                    borderWidth: 0.7

                                }

                        }

                    >

                        <Text>{item.name}</Text>

                    </Pressable>

                )

                )}

            </ScrollView>

        </SafeAreaView>

        {total === 0 ? (

                null

            ) : (

                <Pressable

                style={{

                    backgroundColor: "#088F8F",

                    padding: 10,

                    marginBottom: 40,

                    margin: 15,

                    marginTop:"auto",

                    borderRadius: 7,

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between"

                }}>

                <View>

                    <Text style={{fontSize:17, fontWeight:"600", color:"white"}}>{cart.length} items | $ {total}</Text>

                    <Text style={{fontSize:12, fontWeight:"400", color:"white", marginVertical:6}}>Extra Charges might be Apply</Text>

                </View>

                <Pressable onPress={proceedToCart}>

                    <Text style={{fontSize:17, fontWeight:"600", color:"white"}}>Proceed to Cart!</Text>

                </Pressable>

            </Pressable>

            )}

        </>

    )

}

export default PickUpScreen

const styles = StyleSheet.create({})

## 9. 6 Cart

import {

  StyleSheet,

  Text,

  View,

  SafeAreaView,

  ScrollView,

  Pressable,

} from "react-native";

import React from "react";

import { useDispatch, useSelector } from "react-redux";

import { Ionicons } from "@expo/vector-icons";

import { useNavigation, useRoute } from "@react-navigation/native";

import {

  cleanCart,

  decrementQuantity,

  incrementQuantity,

} from "../CartReducer";

import { decrementQty, incrementQty } from "../ProductReducer";

import { doc, setDoc } from "firebase/firestore";

import { auth, db } from "../firebase";

const CartScreen = () => {

  const cart = useSelector((state) => state.cart.cart);

  const route = useRoute();

  const total = cart

    .map((item) => item.quantity \* item.price)

    .reduce((curr, prev) => curr + prev, 0);

  const navigation = useNavigation();

  const userUid = auth.currentUser.uid;

  const dispatch = useDispatch();

  const placeOrder = async () => {

    navigation.navigate("Order");

    dispatch(cleanCart());

    await setDoc(

      doc(db, "users", `${userUid}`),

      {

        orders: { ...cart },

        pickUpDetails: route.params,

      },

      {

        merge: true,

      }

    );

  };

  return (

    <>

      <ScrollView style={{ marginTop: 50 }}>

        {total === 0 ? (

          <View style={{ justifyContent: "center", alignItems: "center" }}>

            <Text style={{ marginTop: 40 }}>Your cart is empty</Text>

          </View>

        ) : (

          <>

            <View

              style={{

                padding: 10,

                flexDirection: "row",

                alignItems: "center",

              }}

            >

              <Ionicons

                onPress={() => navigation.goBack()}

                name="arrow-back"

                size={24}

                color="black"

              />

              <Text>Your Bucket</Text>

            </View>

            <Pressable

              style={{

                backgroundColor: "white",

                borderRadius: 12,

                marginLeft: 10,

                marginRight: 10,

                padding: 14,

              }}

            >

              {cart.map((item, index) => (

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                    marginVertical: 12,

                  }}

                  key={index}

                >

                  <Text style={{ width: 100, fontSize: 16, fontWeight: "500" }}>

                    {item.name}

                  </Text>

                  {/\* - + button \*/}

                  <Pressable

                    style={{

                      flexDirection: "row",

                      paddingHorizontal: 10,

                      paddingVertical: 5,

                      alignItems: "center",

                      borderColor: "#BEBEBE",

                      borderWidth: 0.5,

                      borderRadius: 10,

                    }}

                  >

                    <Pressable

                      onPress={() => {

                        dispatch(decrementQuantity(item)); // cart

                        dispatch(decrementQty(item)); // product

                      }}

                    >

                      <Text

                        style={{

                          fontSize: 20,

                          color: "#088F8F",

                          paddingHorizontal: 6,

                          fontWeight: "600",

                        }}

                      >

                        -

                      </Text>

                    </Pressable>

                    <Pressable>

                      <Text

                        style={{

                          fontSize: 19,

                          color: "#088F8F",

                          paddingHorizontal: 8,

                          fontWeight: "600",

                        }}

                      >

                        {item.quantity}

                      </Text>

                    </Pressable>

                    <Pressable

                      onPress={() => {

                        dispatch(incrementQuantity(item)); // cart

                        dispatch(incrementQty(item)); //product

                      }}

                    >

                      <Text

                        style={{

                          fontSize: 20,

                          color: "#088F8F",

                          paddingHorizontal: 6,

                          fontWeight: "600",

                        }}

                      >

                        +

                      </Text>

                    </Pressable>

                  </Pressable>

                  <Text style={{ fontSize: 16, fontWeight: "500" }}>

                    ${item.price \* item.quantity}

                  </Text>

                </View>

              ))}

            </Pressable>

            <View style={{ marginHorizontal: 10 }}>

              <Text style={{ fontSize: 16, fontWeight: "bold", marginTop: 30 }}>

                Billing Details

              </Text>

              <View

                style={{

                  backgroundColor: "white",

                  borderRadius: 7,

                  padding: 10,

                  marginTop: 15,

                }}

              >

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                  }}

                >

                  <Text

                    style={{ fontSize: 18, fontWeight: "400", color: "gray" }}

                  >

                    Item Total

                  </Text>

                  <Text style={{ fontSize: 18, fontWeight: "400" }}>

                    ₹{total}

                  </Text>

                </View>

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                    marginVertical: 8,

                  }}

                >

                  <Text

                    style={{ fontSize: 18, fontWeight: "400", color: "gray" }}

                  >

                    Delivery Fee | 1.2KM

                  </Text>

                  <Text

                    style={{

                      fontSize: 18,

                      fontWeight: "400",

                      color: "#088F8F",

                    }}

                  >

                    FREE

                  </Text>

                </View>

                <View style={{ flexDirection: "row", alignItems: "center" }}>

                  <Text

                    style={{ fontSize: 18, fontWeight: "500", color: "gray" }}

                  >

                    Free Delivery on Your order

                  </Text>

                </View>

                <View

                  style={{

                    borderColor: "gray",

                    height: 1,

                    borderWidth: 0.5,

                    marginTop: 10,

                  }}

                />

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                    marginVertical: 10,

                  }}

                >

                  <Text

                    style={{ fontSize: 18, fontWeight: "500", color: "gray" }}

                  >

                    selected Date

                  </Text>

                  <Text

                    style={{

                      fontSize: 18,

                      fontWeight: "400",

                      color: "#088F8F",

                    }}

                  >

                    {/\* {route.params.pickUpDate} \*/}

                  </Text>

                </View>

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                  }}

                >

                  <Text

                    style={{ fontSize: 18, fontWeight: "500", color: "gray" }}

                  >

                    No Of Days

                  </Text>

                  <Text

                    style={{

                      fontSize: 18,

                      fontWeight: "400",

                      color: "#088F8F",

                    }}

                  >

                    {13}

                  </Text>

                </View>

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                    marginVertical: 10,

                  }}

                >

                  <Text

                    style={{ fontSize: 18, fontWeight: "500", color: "gray" }}

                  >

                    selected Pick Up Time

                  </Text>

                  <Text

                    style={{

                      fontSize: 18,

                      fontWeight: "400",

                      color: "#088F8F",

                    }}

                  >

                    {'11:00'}

                  </Text>

                </View>

                <View

                  style={{

                    borderColor: "gray",

                    height: 1,

                    borderWidth: 0.5,

                    marginTop: 10,

                  }}

                />

                <View

                  style={{

                    flexDirection: "row",

                    alignItems: "center",

                    justifyContent: "space-between",

                    marginVertical: 8,

                  }}

                >

                  <Text style={{ fontSize: 18, fontWeight: "bold" }}>

                    To Pay

                  </Text>

                  <Text style={{ fontSize: 18, fontWeight: "bold" }}>

                    {total + 95}

                  </Text>

                </View>

              </View>

            </View>

          </>

        )}

      </ScrollView>

      {total === 0 ? null : (

        <Pressable

          style={{

            backgroundColor: "#088F8F",

            marginTop: "auto",

            padding: 10,

            marginBottom: 40,

            margin: 15,

            borderRadius: 7,

            flexDirection: "row",

            alignItems: "center",

            justifyContent: "space-between",

          }}

        >

          <View>

            <Text style={{ fontSize: 17, fontWeight: "600", color: "white" }}>

              {cart.length} items | $ {total}

            </Text>

            <Text

              style={{

                fontSize: 15,

                fontWeight: "400",

                color: "white",

                marginVertical: 6,

              }}

            >

              extra charges might apply

            </Text>

          </View>

          <Pressable onPress={placeOrder}>

            <Text style={{ fontSize: 17, fontWeight: "600", color: "white" }}>

              Place Order

            </Text>

          </Pressable>

        </Pressable>

      )}

    </>

  );

};

export default CartScreen;

const styles = StyleSheet.create({});

## 9.7 Orderscreen.js

import { StyleSheet, Text, View, SafeAreaView } from "react-native";

import React from "react";

import LottieView from "lottie-react-native";

const OrderScreen = () => {

  return (

    <SafeAreaView>

      <Text

        style={{

          marginTop: 40,

          fontSize: 19,

          fontWeight: "600",

          textAlign: "center",

        }}

      >

        Your order has been placed

      </Text>

    </SafeAreaView>

  );

};

const styles = StyleSheet.create({});

export default OrderScreen;

## 9.8 profile screen.js

import { StyleSheet, Text, View,SafeAreaView,Pressable } from 'react-native'

import React from 'react'

import { auth } from '../firebase'

import { signOut } from 'firebase/auth';

import { useNavigation } from '@react-navigation/native';

const ProfileScreen = () => {

    const user = auth.currentUser;

    const navigation = useNavigation();

    const signOutUser = () => {

        signOut(auth).then(() => {

            navigation.replace("Login");

        }).catch(err => {

            console.log(err);

        })

    }

  return (

    <SafeAreaView style={{flex:1,justifyContent:"center",alignItems:"center"}}>

      <Pressable style={{marginVertical:10}}>

        <Text>welcome {user.email}</Text>

      </Pressable>

      <Pressable onPress={signOutUser}>

          <Text>Sign Out</Text>

      </Pressable>

    </SafeAreaView>

  )

}

export default ProfileScreen

const styles = StyleSheet.create({})

## 9.9 dress item services

import { StyleSheet, Text, View, Pressable, Image } from "react-native";

import React from "react";

import { useDispatch, useSelector } from "react-redux";

import {

  addToCart,

  decrementQuantity,

  incrementQuantity,

} from "../CartReducer";

import { decrementQty, incrementQty } from "../ProductReducer";

const DressItem = ({ item }) => {

  const dispatch = useDispatch();

  const cart = useSelector((state) => state.cart.cart);

  const addItemToCart = () => {

    dispatch(addToCart(item)); // cart

    dispatch(incrementQty(item)); // product

  };

  return (

    <View>

      <Pressable

        style={{

          backgroundColor: "#F8F8F8",

          borderRadius: 8,

          padding: 10,

          flexDirection: "row",

          alignItems: "center",

          justifyContent: "space-between",

          margin: 14,

        }}

      >

        <View>

          <Image

            style={{ width: 70, height: 70 }}

            source={{ uri: item.image }}

          />

        </View>

        <View>

          <Text

            style={{

              width: 83,

              fontSize: 17,

              fontWeight: "500",

              marginBottom: 7,

            }}

          >

            {item.name}

          </Text>

          <Text style={{ width: 60, color: "gray", fontSize: 15 }}>

            ${item.price}

          </Text>

        </View>

        {cart.some((c) => c.id === item.id) ? (

          <Pressable

            style={{

              flexDirection: "row",

              paddingHorizontal: 10,

              paddingVertical: 5,

            }}

          >

            <Pressable

                        onPress={() => {

                            dispatch(decrementQuantity(item)); // cart

                            dispatch(decrementQty(item)); // product

                        }}

                        style={{

                            width: 26,

                            height: 26,

                            borderRadius: 13,

                            borderColor: "#BEBEBE",

                            backgroundColor: "#E0E0E0",

                            justifyContent: "center",

                            alignContent: "center",

                        }}

                    >

              <Text

                style={{

                  fontSize: 20,

                  color: "#088F8F",

                  paddingHorizontal: 6,

                  fontWeight: "600",

                  textAlign: "center",

                }}

              >

                -

              </Text>

            </Pressable>

            <Pressable>

              <Text

                style={{

                  fontSize: 19,

                  color: "#088F8F",

                  paddingHorizontal: 8,

                  fontWeight: "600",

                }}

              >

                {item.quantity}

              </Text>

            </Pressable>

            <Pressable

                        onPress={() => {

                            dispatch(incrementQuantity(item)); // cart

                            dispatch(incrementQty(item)); // product

                        }}

                        style={{

                            width: 26,

                            height: 26,

                            borderRadius: 13,

                            borderColor: "#BEBEBE",

                            backgroundColor: "#E0E0E0",

                            justifyContent: "center",

                            alignContent: "center",

                        }}

                    >

              <Text

                style={{

                  fontSize: 20,

                  color: "#088F8F",

                  paddingHorizontal: 6,

                  fontWeight: "600",

                  textAlign: "center",

                }}

              >

                +

              </Text>

            </Pressable>

          </Pressable>

        ) : (

          <Pressable onPress={addItemToCart} style={{ width: 80 }}>

            <Text

              style={{

                borderColor: "gray",

                borderRadius: 4,

                borderWidth: 0.8,

                marginVertical: 10,

                color: "#088F8F",

                textAlign: "center",

                padding: 5,

                fontSize: 17,

                fontWeight: "bold",

              }}

            >

              Add

            </Text>

          </Pressable>

        )}

      </Pressable>

    </View>

  );

};

export default DressItem;

const styles = StyleSheet.create({});

## 9.10 carousel.js

import { StyleSheet, Text, View } from 'react-native'

import React from 'react'

import { SliderBox } from "react-native-image-slider-box";

const Carousel = () => {

    const images = [

        "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/druwjzmfmz7qvepq3bkr",

        "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/lv6jl9qdscekjmwkxm9l",

      ];

      return (

        <View>

          <SliderBox

            images={images}

            autoPlay

            circleLoop

            dotColor={"#13274F"}

            inactiveDotColor="#90A4AE"

            ImageComponentStyle={{

              borderRadius: 6,

              width: "94%",

            }}

          />

          </View>

  )

}

export default Carousel

const styles = StyleSheet.create({})

## 9.11 Services.js

import { StyleSheet, Text, View ,ScrollView,Pressable,Image} from 'react-native'

import React from 'react'

const Services = () => {

 const services = [

    {

      id: "0",

      image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/lv6jl9qdscekjmwkxm9l",

      name: "Egg Rice",

    },

    {

      id: "11",

      image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/akmx533z73jjbq8avy6v",

      name: "Chicken Rice"

    },

    {

      id: "12",

      image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/byonwwb8mzxyqluxlqpq",

      name: "Chilly Psneer",

    },

    {

      id: "13",

      image: "https://res.cloudinary.com/swiggy/image/upload/fl\_lossy,f\_auto,q\_auto,w\_208,h\_208,c\_fit/fsitbray4gq1kxcndqqx",

      name: "Spl Veg Biryani",

    },

  ];

  return (

    <View style={{padding:10}}>

        <Text style={{fontSize:16,fontWeight:"500",marginBottom:7}}>Services Available</Text>

        <ScrollView horizontal showsHorizontalScrollIndicator={false}>

            {services.map((service,index) => (

                <Pressable style={{margin:10,backgroundColor:"white",padding:20,borderRadius:7}} key={index}>

                    <Image source={{uri:service.image}} style={{width:70,height:70}}/>

                    <Text style={{textAlign:"center",marginTop:10}}>{service.name}</Text>

                </Pressable>

            ))}

        </ScrollView>

    </View>

  )

}

export default Services

const styles = StyleSheet.create({})

# 9.12 Cart Reducer.js

import { createSlice } from "@reduxjs/toolkit";

export const CartSlice = createSlice({

    name:"cart",

    initialState:{

        cart:[],

    },

    reducers:{

        addToCart:(state,action) => {

            const itemPresent = state.cart.find((item) => item.id === action.payload.id);

            if(itemPresent){

                itemPresent.quantity++;

            }else{

                state.cart.push({...action.payload,quantity:1})

            }

        },

        removeFromCart:(state,action) => {

            const removeItem = state.cart.filter((item) => item.id !== action.payload.id);

            state.cart = removeItem;

        },

        incrementQuantity:(state,action) => {

            const itemPresent = state.cart.find((item) => item.id === action.payload.id);

            itemPresent.quantity++;

        },

        decrementQuantity:(state,action) => {

            const itemPresent = state.cart.find((item) => item.id === action.payload.id);

            if(itemPresent.quantity == 1){

                itemPresent.quantity = 0;

                const removeItem = state.cart.filter((item) => item.id !== action.payload.id);

                state.cart = removeItem;

            }else{

                itemPresent.quantity--;

            }

        },

        cleanCart:(state) => {

            state.cart = [];

        }

    }

});

export const {addToCart,removeFromCart,incrementQuantity,decrementQuantity,cleanCart} = CartSlice.actions;

export default CartSlice.reducer;

# 9.13 Product reducer.js

import { createSlice } from "@reduxjs/toolkit";

export const productSlice = createSlice({

    name:"product",

    initialState:{

        product:[],

    },

    reducers:{

        getProducts:(state,action) => {

            state.product.push({...action.payload});

        },

        incrementQty: (state, action) => {

            const itemPresent = state.product.find((item) => item.id === action.payload.id);

            if (itemPresent) {

                itemPresent.quantity++;

            }

        },

        decrementQty: (state, action) => {

            const itemPresent = state.product.find((item) => item.id === action.payload.id);

            if (itemPresent.quantity === 1) { // Check if quantity is 1 before decrementing

                itemPresent.quantity = 0;

                const removeItem = state.product.filter((item) => item.id !== action.payload.id);

                state.product = removeItem; // Use 'product' instead of 'cart'

            } else {

                itemPresent.quantity--;

            }

        },

    }

});

export const {getProducts,incrementQty,decrementQty} = productSlice.actions;

export default productSlice.reducer;

# 9.14 Stack Navigator

import { StyleSheet, Text, View } from 'react-native';

import React from 'react';

import { NavigationContainer } from '@react-navigation/native';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import HomeScreen from './screens/HomeScreen';

import PickUpScreen from './screens/PickUpScreen';

import CartScreen from './screens/CartScreen';

import LoginScreen from './screens/LoginScreen';

import RegisterScreen from './screens/RegisterScreen';

import ProfileScreen from './screens/ProfileScreen';

import OrderScreen from './screens/OrderScreen';

const StackNavigator = () => {

    const Stack = createNativeStackNavigator();

  return (

    <NavigationContainer>

      <Stack.Navigator>

      <Stack.Screen name="Login" component={LoginScreen} options={{headerShown:false}} />

        <Stack.Screen name="Home" component={HomeScreen} options={{headerShown:false}} />

        <Stack.Screen name="PickUpScreen" component={PickUpScreen} options={{headerShown:false}} />

        <Stack.Screen name ="CartScreen" component={CartScreen} options={{headerShown:false}}/>

        <Stack.Screen name="Register" component={RegisterScreen} options={{headerShown:false}}/>

        <Stack.Screen name ="Profile" component={ProfileScreen} options={{headerShown:false}}/>

        <Stack.Screen name ="Order" component={OrderScreen} options={{headerShown:false}}/>

      </Stack.Navigator>

    </NavigationContainer>

  )

}

export default StackNavigator

const styles = StyleSheet.create({})