Boston International College

(Affiliated to Pokhara University)

Bharatpur 10, Chitwan

A MAJOR PROJECT ON

"खाना Finder"

Submitted in Partial Fulfillment of the Requirements for
the degree of **Bachelor in Computer Information System**
under **Pokhara University,** Nepal

SUBMITTED BY :

Aarati Tamang: [19080151]
Pratina Poudel: [19080160]
Asija Thapa: [19080153]
Supriya Wasti: [19080164]

SUBMITTED TO:

Department of Computer Information System

Boston International College

Bharatpur 10,Chitwan

June, 2023

Dedication

We would like to express our heartfelt appreciation and gratitude to our beloved and highly esteemed parents for their unwavering belief in us from the very beginning and for their constant support. Similarly, we extend our dedication of this report to our supervisor and lecturers, whose consistent guidance and responsiveness have been instrumental in our journey.

Declaration

We, **Pratina Poudel, Aarati Tamang, Asija Thapa and Supriya Wasti,** hereby, declare that the work presented in this project entitled **"KHANA FINDER"** is the outcome of the investigation performed by us under the supervision of faculty, Professor at Boston International College affiliated to Pokhara University. We also declare that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma. We owe all the liabilities relating to the accuracy and authenticity of the data and any other information included hereunder.

…….

Er. Mini Madhav Khanal

Supervisor

Date: June, 2023

Recommendation

This is to certify that this project work entitled **"KHANA FINDER",** prepared and submitted by **Aarati Tamang, Asija Thapa, Pratina Poudel, Supriya Wasti** in partial fulfillment of the requirements of the degree of Bachelor in Computer Information System awarded by Pokhara University, has been completed under my/our supervision. We recommend the same for acceptance by Pokhara University.

Signature ……

Name of the Supervisor: Er. Mini Madhav Khanal

Organization: Boston International College

Date: June, 2023

Certificate

This project entitled "**KHANA FINDER"** prepared and submitted by **Aarati Tamang, Asija Thapa, Pratina Poudel, Supriya Wasti** has been examined by us and is accepted for the award of the degree of Bachelor in Computer Information System by Pokhara University.

Name of External: Signature Date

Designation:

Organization:

Name of Supervisor: Er. Mini Madhav Khanal Signature Date

Designation : Lecturer

Organization : Boston International College

Head of Department: Nabin Kumar Shrestha Signature Date

Organization : Boston International College

Acknowledgement

We would like to express our special thanks to our college Boston International College, Principal/Chairman of our college Prof. Anand Bahadur Chand, department head, beloved friends, supervisor **Mr Er.Mini Madhav Khanal** and all those individuals who were directly and indirectly involved during preparation of our project.

Name of the Students:
Aarati Tamang
Asija Thapa
Pratina Poudel
Supriya Wasti

Date: June, 2023

Abstract

As a firm, staying competitive in the market is never easy. It faces a lot of competition from each and every competitor. The firms have to always come up with a better strategy to satisfy their customers, incorporate latest technologies to provide better service to their customers. The mobile application for food delivery is a solution that supports collaboration in restaurants as the foundation for gaining competitive advantage and maintaining market share.This paper describes an ongoing effort in developing a food delivery application. There are some other ways to obtain the collaboration but this is a better solution. There are several technologies that are needed for the design and implementation of this mobile application, some of them include technologies like Android Studio, React Native, etc. There is a high level integration needed to bring this on to a single track and make this work. This mobile application has used the up to date technologies so this way the application can be the most sophisticated one on the market. This application uses technologies that are completely open source and involves a lot less capital than other tools out there in the world. This way the firm whoever uses this technology will be able to see a growth in productivity, higher profits and the most important thing would be to make the customers happier. With some minor changes to the application it shall be able to make it available to other companies as well. In order to figure out if this application was indeed capable of managing the supply chain management in the organization, this report specifics the technologies utilized and how they became acquainted. This section provides an overview of the application's design and the three demand forecasts. The findings indicated that the food delivery process had improved, and expenses had decreased.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

App- Application

VSCode- Visual Studio Code

ERD- Entity Relationship Diagram

GPS-Global Positioning System

API-Application programming interface

CHAPTER-1
Introduction

1.1 Background

In recent years, food delivery apps have gained significant popularity in Nepal.With the rise of technology and smartphones,people have become more reliant on digital platforms to order food from their favorite restaurants and cafes. Food transportation is a critical part of any company that prepares foods or offers services. Effective food delivery helps businesses improve customer happiness, cut costs, and assure on-time product delivery to customers. Since they enable real-time data analysis, order tracking, and straightforward communication, mobile applications now play a critical role in food delivery.

Programmers may create outstanding cross-platform apps for the iOS and Android operating systems using the well-known open-source ReactNative mobile application development framework. Applications created using ReactNative are recognized for their exceptional performance, gorgeous user interfaces, and seamless user interfaces.

1.2 Problem Statement

For placing any orders customers have to visit hotels or restaurants to know about food items and then place order and pay, for that process time and manual work is required. While placing an order over the phone, the customer lacks the physical copy of the menu item, and lacks visual confirmation that the order was placed correctly. Every restaurant needs certain employees to take the order over phone or in-person, to offer a rich dining experience and process the payment.

In today's market, labor rates are increasing day by day making it difficult to find employees when needed. Hence, to solve this issue, **"Khana Finder"**, is originally designed for small scale businesses like Cafe, Fast Food restaurants. The system greatly simplifies the ordering process for both the customer and the restaurant and also greatly lightens the load on the restaurants. The system also automatically calculates and displays

the final bill so the bills are ready to print without having any error because the information for that item will be already inserted.

1.3 Objectives

- To offer a convenient way for customers to order food from a variety of restaurants.
- To establish efficient logistics and delivery systems.
- Personalize user experience to acquire useful data that may be utilized to enhance the operation of the app.

1.4 Applications

Our online food delivery app **("Khana Finder")** is suitable for a wide spectrum of consumers, including:

- **Customers:** The primary users of our online food delivery app are customers who wish to purchase food from restaurants and have it delivered to their door. Customers may use the app to explore menus, place orders, make payments, and monitor their deliveries.

- **Restaurants and Food enterprises:** Our Online food delivery app gives an extra channel for restaurants and food enterprises to access a broader consumer base. They can collaborate with the app to take orders from clients, enhancing their visibility and sales.

- **Delivery Partners:** Individuals that work with the online food delivery service to pick up orders from restaurants and deliver them to clients are known as delivery partners. They utilize the app to obtain order details, route to the restaurant and customer locations, and update delivery status.

- **App Administrators and Support employees:** To manage the platform, assure smooth operations, respond to consumer inquiries, and handle any technical issues that may emerge, the online food delivery app requires administrators and support employees.

CHAPTER-2
Literature Review

Online food delivery app is the system where we can order the food item through the internet with just one click, which can make our daily life easier and faster. Currently, clients visit restaurants, choose their meals, and make payments on average once a day. Some restaurants allow patrons to pre-order items that will be available for pick-up or delivery by calling the restaurant in advance. Because the restaurants occasionally run out of particular goods or because there is no option to order specialty delicacies, some customers don't always get the variety they want. [1]

Previous works:

DoorDash is the most popular among all the apps that provide online food delivery services to the consumers. It is an American company based in San Francisco, California that operates an online food ordering and food delivery platform. With a 56% market share, DoorDash is the largest food delivery company in the United States. It also has a 60% market share in the convenience delivery category. As of December 31, 2020, the platform was used by 450,000 merchants, 20,000,000 consumers, and one million deliverers. While DoorDash has established itself as a dominant player in the food delivery market, there is still room for improvement. One limitation is the availability of restaurants on the platform. While DoorDash has a vast network of partner restaurants, there may still be gaps in coverage, particularly in certain geographical areas or cuisines. Additionally, DoorDash's fees and surcharges, along with long delivery times during peak hours, can sometimes impact the overall customer experience. **(Source: via.wikipedia)**

In the context of Nepal, a number of limitations have been brought to light by prior research on mobile applications for ordering meals in Nepal.This section reviews prior research on food delivery smartphone apps in Nepal, highlighting its flaws and describing how **Khana Finder** can address them.

Study 1: User Experience and Interface Design

Previous studies by Regmi et al. (2019) focused on analyzing the user interface and user experience of mobile apps for ordering food in Nepal. The study highlighted negative aspects such as challenging navigation and lengthy loading times. Khana Finder will avoid these limitations by prioritizing user-friendly user interfaces and improving loading time.

Study 2: Payment Systems and Challenges

Research by Gautam et al. (2020) investigated the challenges related to payment systems in food ordering mobile applications in Nepal. The study found some drawbacks, including the limited local payment choices. To overcome these restrictions, Khana Finder will incorporate widely used local payment options like eSewa, Khalti, or IME Pay. [3]

Study 3: Delivery Logistics and Infrastructure

A study conducted by Rai et al. (2021) examined the challenges and opportunities associated with delivery logistics and infrastructure in food ordering mobile applications in Nepal. The research highlighted limitations such as unreliable delivery services, inconsistent tracking mechanisms, and delays in delivery. To overcome these limitations, Khana Finder will focus on building a robust network of reliable delivery partners, implementing real-time tracking systems and ensuring timely and efficient service.

Study 4: Customer Support and Feedback

Customer support and feedback play a vital role in ensuring user satisfaction and resolving issues promptly. However, research by Bhattarai et al. (2020) identified limitations in terms of slow response times, lack of effective communication channels, and inadequate handling of user complaints. To overcome these limitations, Khana Finder will focus on establishing efficient customer support channels, providing timely responses, and implementing mechanisms to address user feedback and complaints.[5]

CHAPTER-3
METHODOLOGY

3.1 System overview/ Architectural Representation of the system.

In this project we have developed an online food delivery application . Where there is admin, who provides id and password to the restaurant and then the restaurant contacts the admin and receives the id and password and then after the restaurant logs into the system they can start doing their job ie: inserting the food items with actual price. Here all the customers can visit the page and place an order.

This system is a completely mobile based application and is developed using ReactNative. We have one home page where multiple options allow the user to choose different food items with the actual price. There is a system administrator who will have the right to add and manage user accounts, a manager who will be managing product and orders. Following are the system design of the system:

- Create an account.
- Manage their account.
- Log in to the system.
- Navigate the restaurant's menu.
- Select an item from the menu.
- Add an item to their current order.
- Review their current order.
- Place an order.
- Can track their orders
- Receive confirmation

3.2 Research design and method

In this project, we have outlined the research design and methods that we have used to ensure that the application meets the needs of its users and is efficient, user-friendly, and reliable. The research design we have used in this project is a mixed-methods approach that combines qualitative and quantitative research methods. This approach enables us to gather data from various sources.

The following quantitative research methods are employed:

- **Surveys:** We have conducted surveys to gather data from potential users of the food delivery application. The survey is distributed through social media, email, and other online platforms.
- **Data Analysis:** We have analyzed the data collected from the surveys to identify user preferences and needs. This information helped us to design the application to meet the needs of its users**.**

3.2.1 Entity Relationship Diagram

The entities and their relationships in the system to be created are established using an entity relationship diagram. ERD exemplifies the logical organization of databases. ER diagrams can offer a unified picture of data that is independent of any data model and are often easy to use and understand.

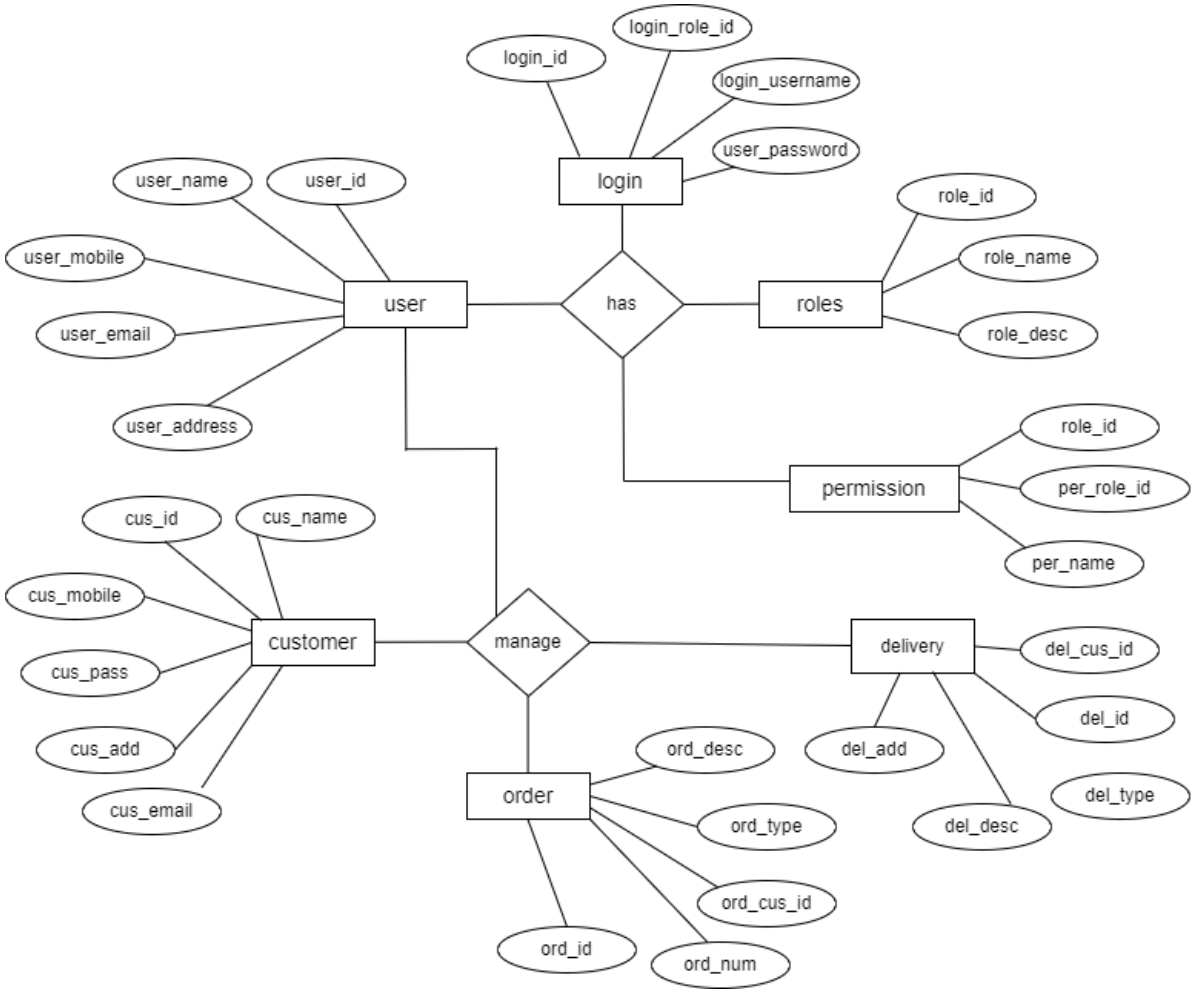


Figure 1 Entity Relationship Diagram of Khana Finder

(Source: Group study, 2023. Made on draw.io)

3.2.2 Use Case Diagram

The fundamental form of system/software requirements for a new, developing software program is displayed by IT as a use case diagram. Use cases describe the desired action (what), not the precise process by which it will be accomplished (how). Once defined, use cases can be represented textually or visually (using a use case diagram). Use case modeling's ability to assist in system design from the standpoint of the end user is a fundamental idea.

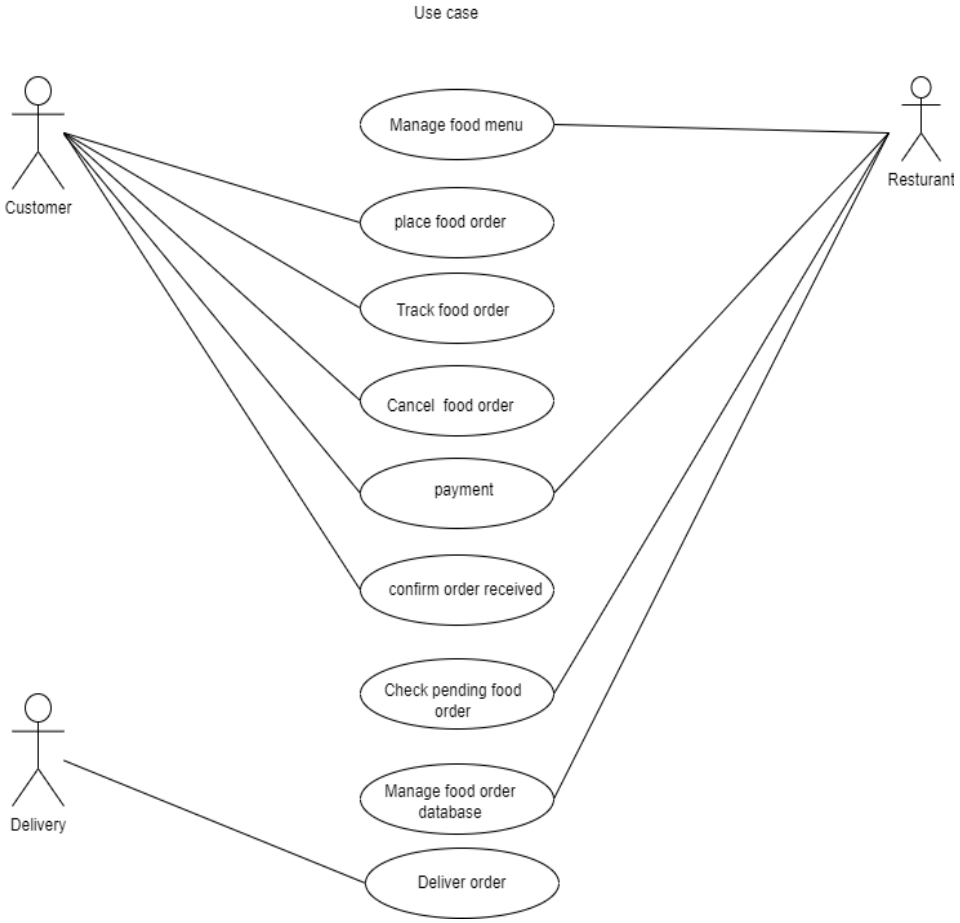


Figure 2: Use Case Diagram Of Khana Finder

(Source: Group study, 2023. Made on draw.io)

3.2.3 Flowchart

DFD depicts the information flow within a system. They show the information and how it moves across particular system activities. They offer one type of report documentation. These charts aid in illustrating the top-down, graphical flow of data through the system. They aid in providing a pictorial depiction of the system's parts, functions, and interfaces.

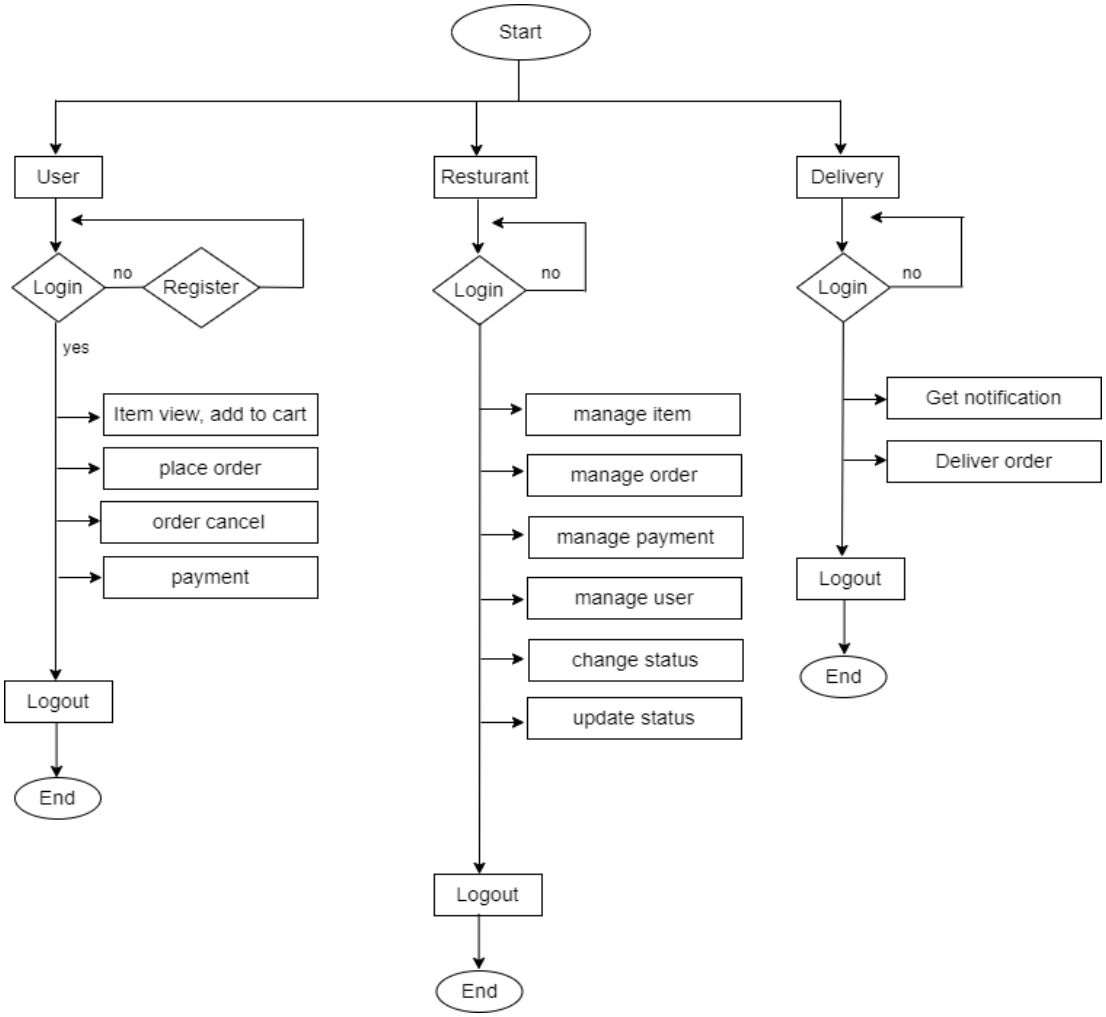


Figure 3: Flowchart of Khana Finder

(Source: Group study, 2023. Made on draw.io)

3.2.4 Software Development Life Cycle (SDLC)

As seen in Figure, the incremental model is a method of software development where requirements are divided into a number of independent software development cycle modules. Analysis, design, implementation, testing/verification, and maintenance are the stages of incremental development. As depicted in Figure, each iteration goes through the processes of requirements, design, coding, and testing. And when more capability is added to the system with each new iteration, it eventually reaches its full potential. When the initial increment is provided, the system is put into production. The fundamental functionality is frequently handled in the first increment, and then additional features are added in the following increments. Following the client's analysis of the core product, plans are developed for the following increment.

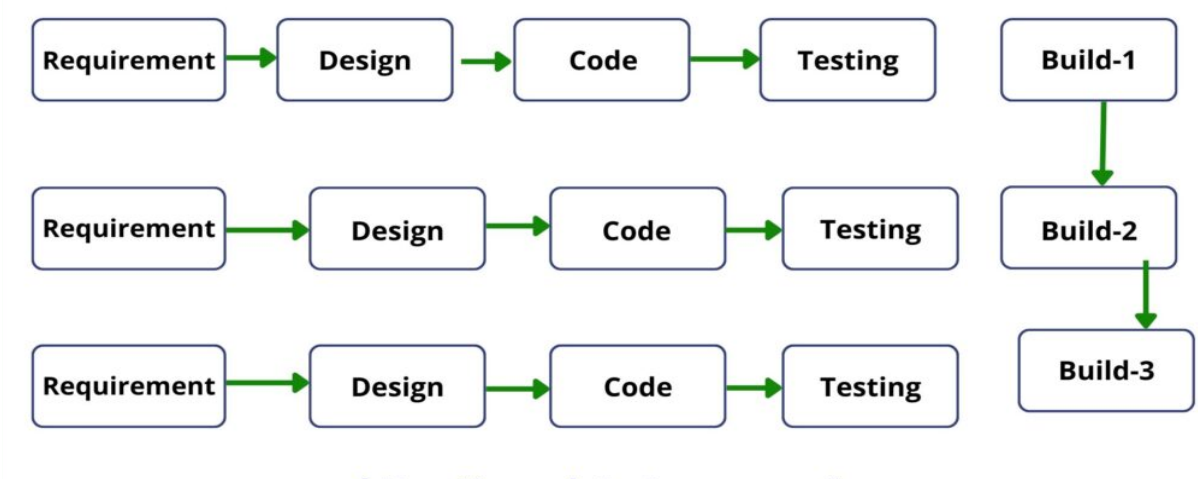


Figure : Incremental Model

Explanation of incremental phases are as follows:

- **Requirement analysis**: The requirements and specifications for the software are gathered during this first stage of the software development paradigm.
- **Design:** During this stage, various high-end functions are created after gathering all the necessary resources.
- **Code:** Following that, software is coded at this level.

- **Test:** The testing phase begins after the system has been deployed.

3.3 Tools and Technologies used

There are several technologies used for the development of **Khana Finder.** They are as follows:

3.3.1 Software

Khana Finder uses a range of tools and technologies, including:

- **ReactNative**

  An open-source framework called React Native allows developers to use React and JavaScript to create cross-platform mobile applications. By sharing a sizable percentage of the codebase between the iOS and Android platforms, it enables developers to construct native-like apps for both of these operating systems. React Native increases development productivity while offering near-native performance with rapid reloading and access to native components.

- **API**

  A concept known as "API" (application programming interface) is used in a variety of contexts, including enterprise code, microservices, and cloud-native architectures. An API is an interface that programmatically allows software developers to communicate with resources or software components that are not part of their own code. API are required to connect apps and carry out a predetermined function that is based on sharing data and running predefined operations. They serve as a go-between, enabling programmers to create fresh programmatic interactions across the many programs that consumers and companies use on a daily basis. Some advantages of API are:

  ➔ Coordination with system owners and other stakeholders is helpful when updating the agency's systems.

- ➔ Content can be created once and automatically published or made accessible to multiple channels with API access.
- ➔ In order to deliver specialized user experiences, APIs can be used to deliver services and information to new audiences and in certain settings.
- ➔ APIs enable computers to manage workloads that would otherwise need humans to do manual tasks.

- **VS CODE**

  A device that helps a programming platform quickly create software applications. The word either refers to software packages used by developers to author and bundle multimedia content that is delivered to end users or tools that enable the development of whole programs using compiled or interpreted code. Common names for the former include "integrated development environment" and "authoring tool" (JavaScript). Features of VS Code are:

  - ➔ It is a desktop-based source code editor for Windows that is lightweight but effective.
  - ➔ It includes Git integration, which enables you to push, pull, and commit code changes to an external Git repository.
  - ➔ To link the current file extension to an installed language, we can use the configure file association for command in Vs code.
  - ➔ With the use of the split view in Visual Studio Code, multiple files from the same project can usually be worked on at once.
  - ➔ If any code fragment is left unfinished, it can identify it.
  - ➔ Themes like Light and Dark, Git Repositories, Commit "Action Button," Three-Way Merge Editor, and others are available in Vs Code.

- **Visual Studio Code**

  Microsoft created Visual Studio Code (VS Code), a popular source code editor. It offers a highly adaptable and expandable development environment and is free

and open-source. There are versions of VS Code on Windows, macOS, and Linux.

These are some of Visual Studio Code's main attributes:

- ➔ Speedy and light: Fast startup and effective operation.
- ➔ Syntax highlighting, auto-completion, code formatting, and linting are all features of intelligent code editing.
- ➔ Extensions: A wide variety of extensions for various tools, frameworks, and languages.
- ➔ Debug code within the editor with the built-in debugger.
- ➔ Run scripts and command-line tools inside the editor using the integrated terminal.
- ➔ Task automation: Assistance with task runners and build systems for automating processes.

3.3.2 Hardware

The basic hardware requirements for using Khana Finder include:

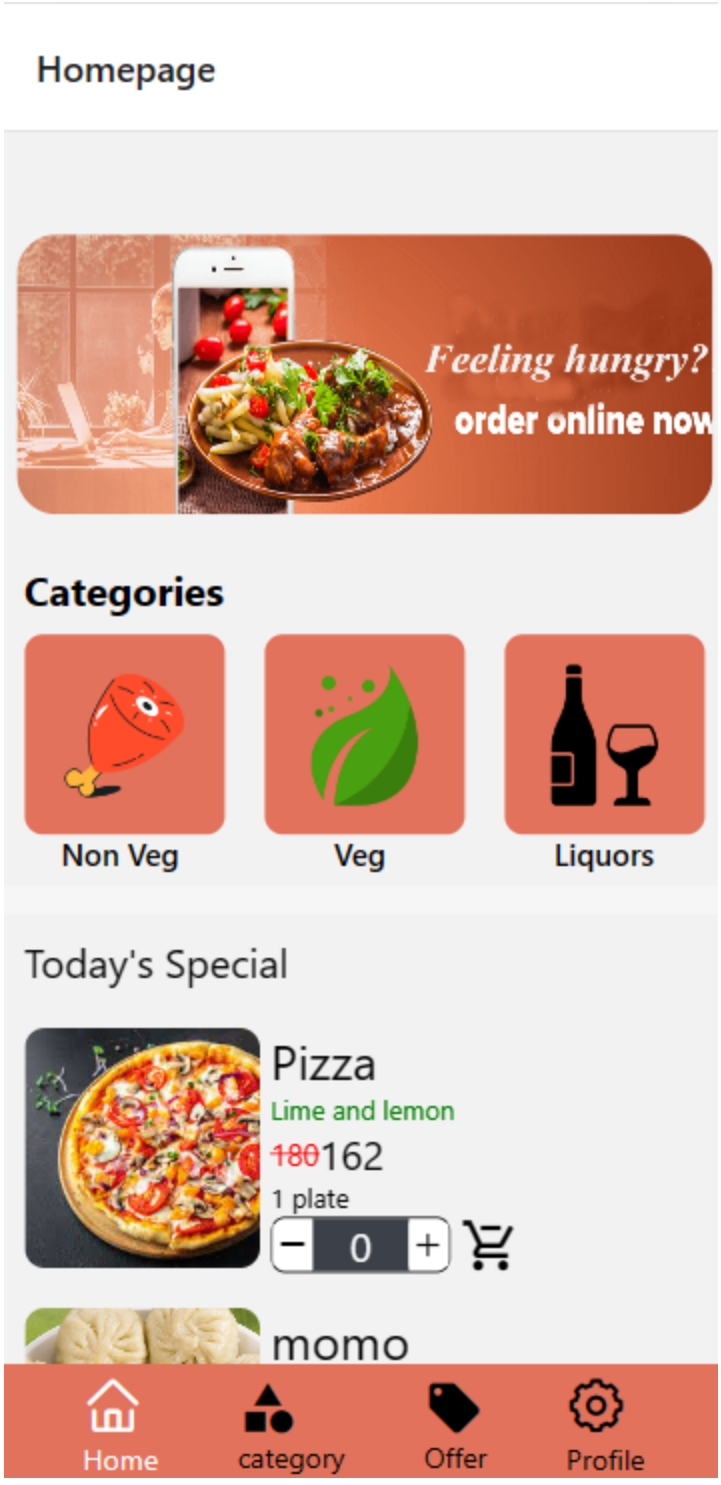
- Smartphone: Any modern smartphone running on either Android or iOS operating systems can be used to access and use the Khana Finder application.
- Internet Connectivity: To use Khana Finder, a stable and reliable internet connection is required. The application relies on the internet to display restaurant menus, place orders, and track deliveries.
- GPS: **Khana Finder** utilizes GPS technology to track the location of customers and delivery personnel, and to provide accurate location-based services.

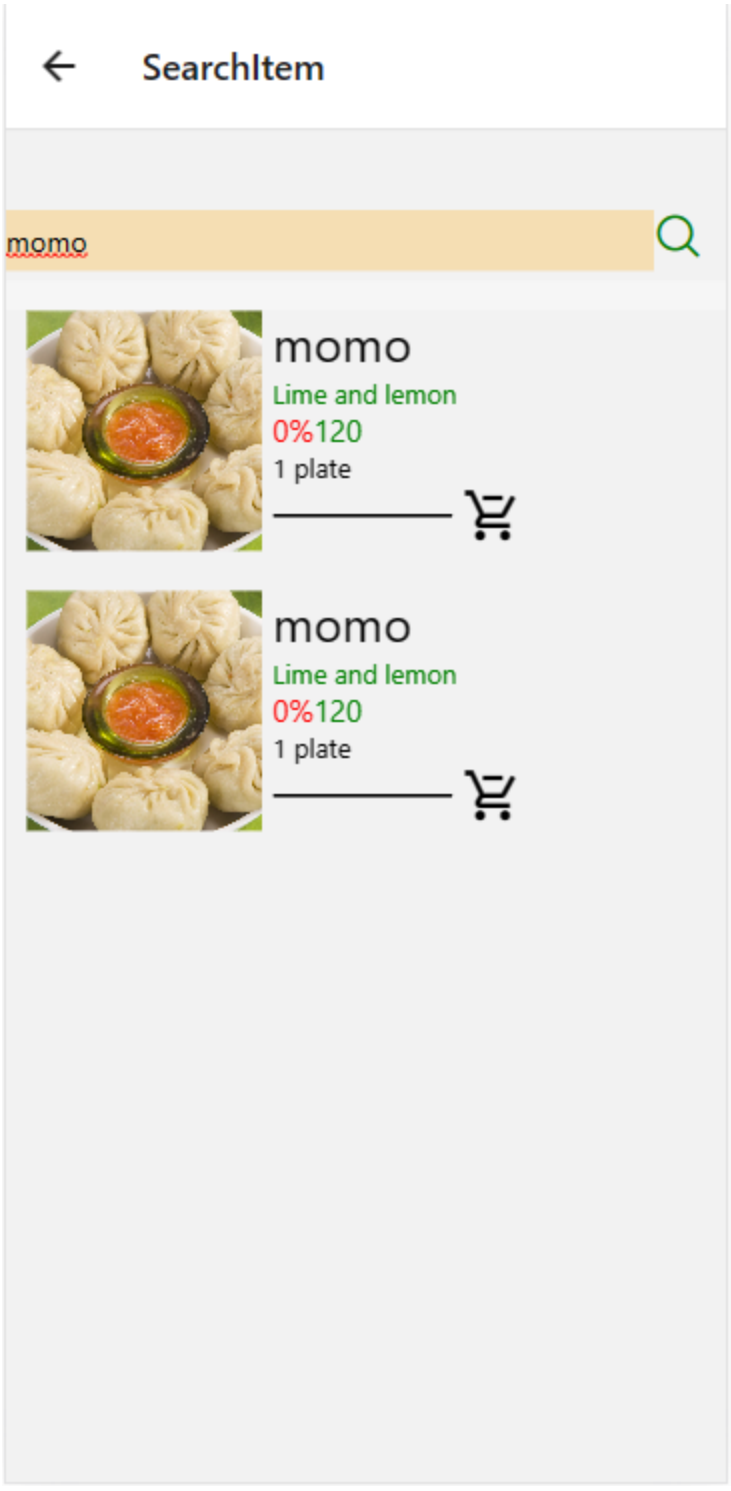
CHAPTER-4
EPILOGUE

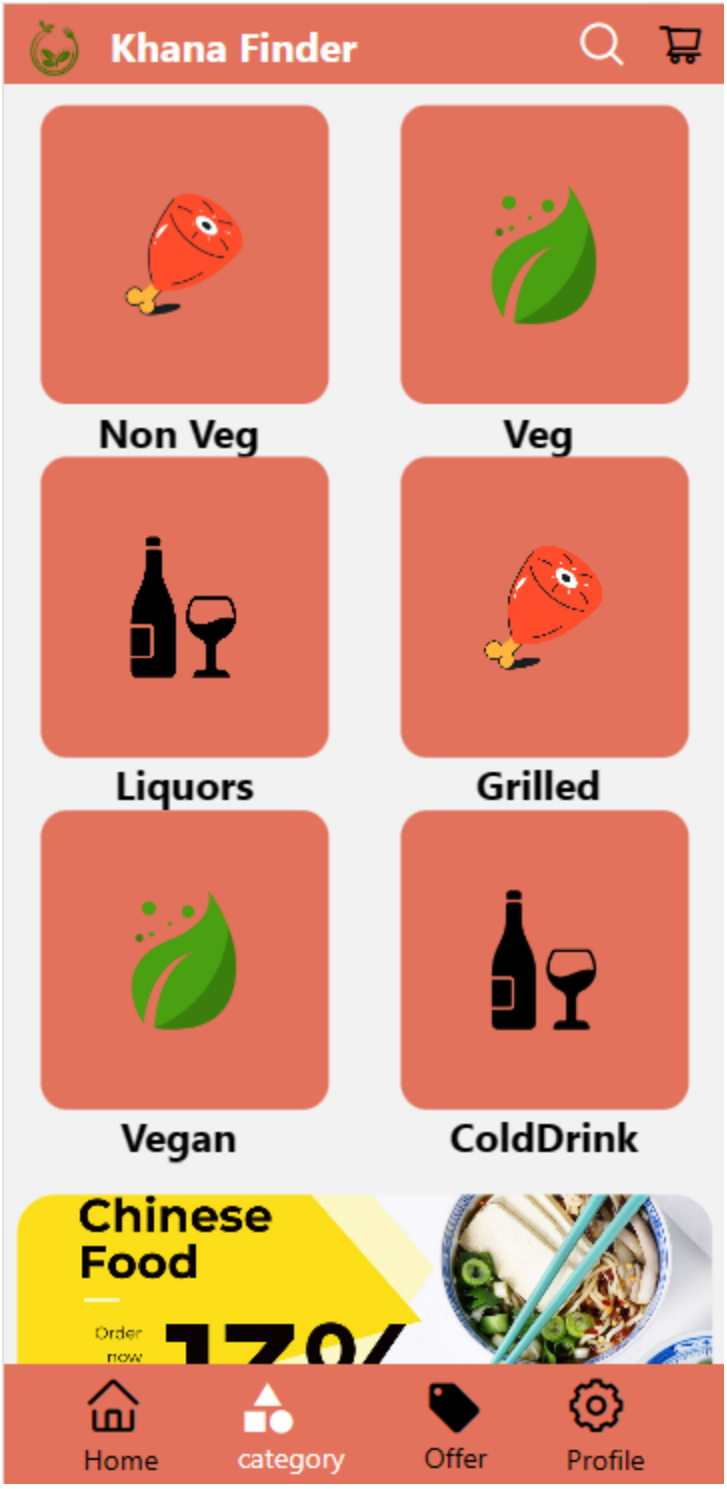
4.1 Expected Output

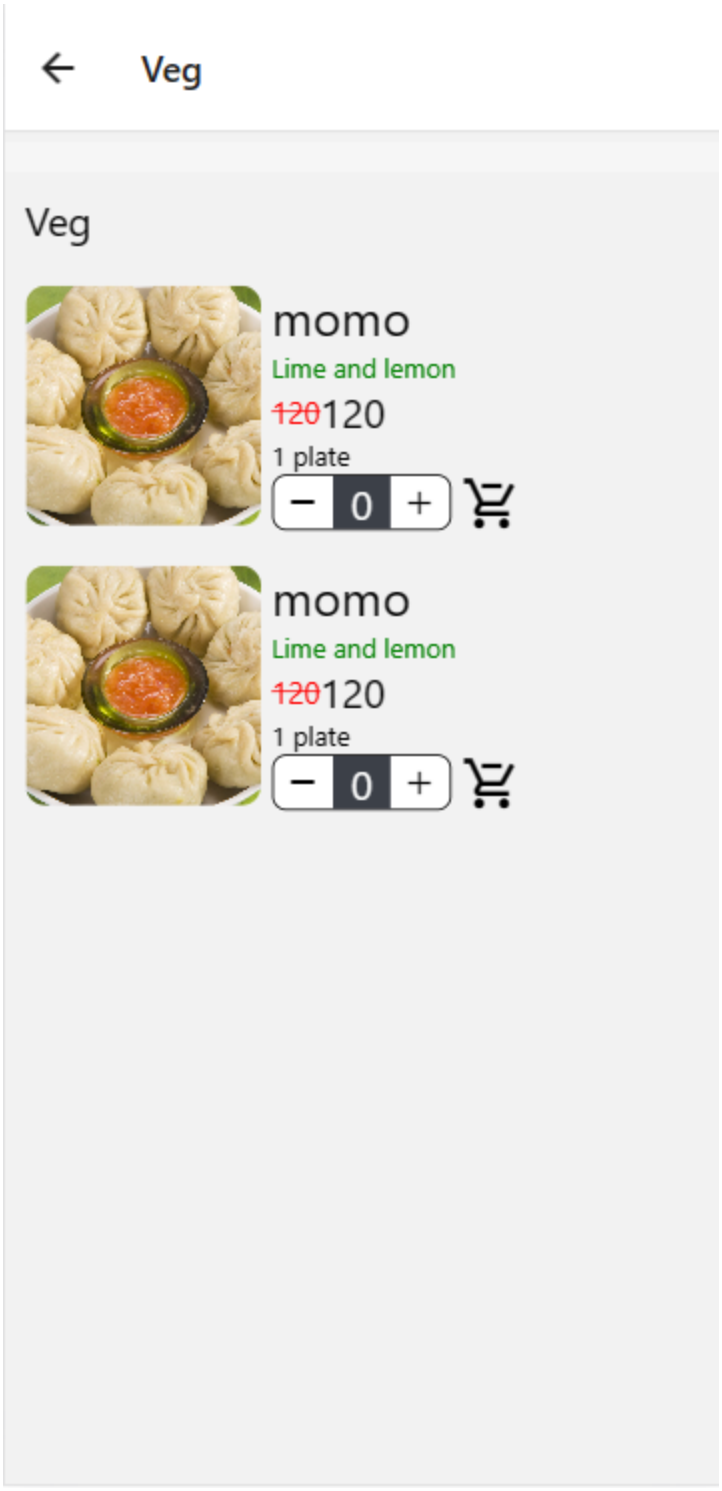
After the completion of Khana Finder, we got the following outputs which will minimize the problems as well as solve the existing problem.

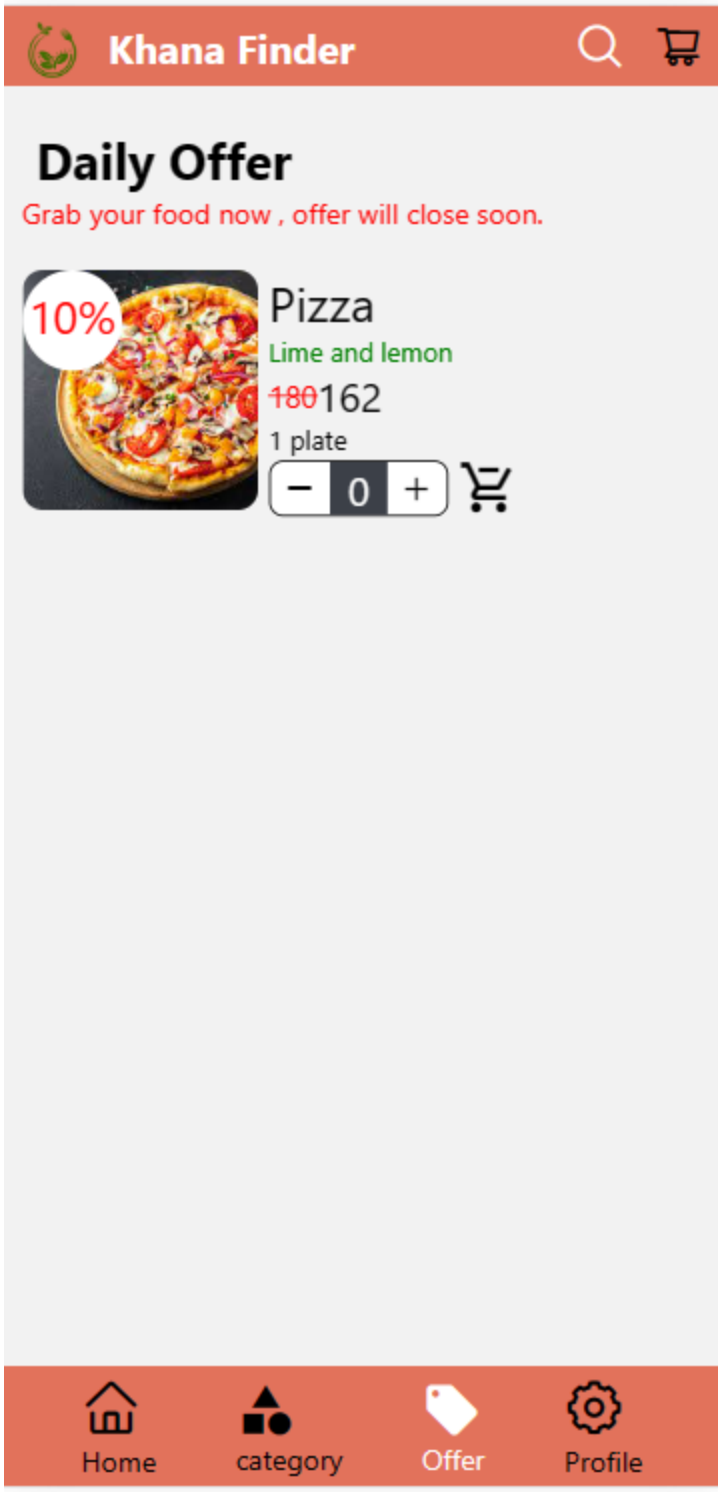
- Automatically compute the bill
- Lightens the load on the restaurants
- No more busy phones or the requirement for extra phone lines
- Accommodate huge amount of order at a time
- Improve the communication between the client and the server
- Efficient in term of time and money
- Provide better services

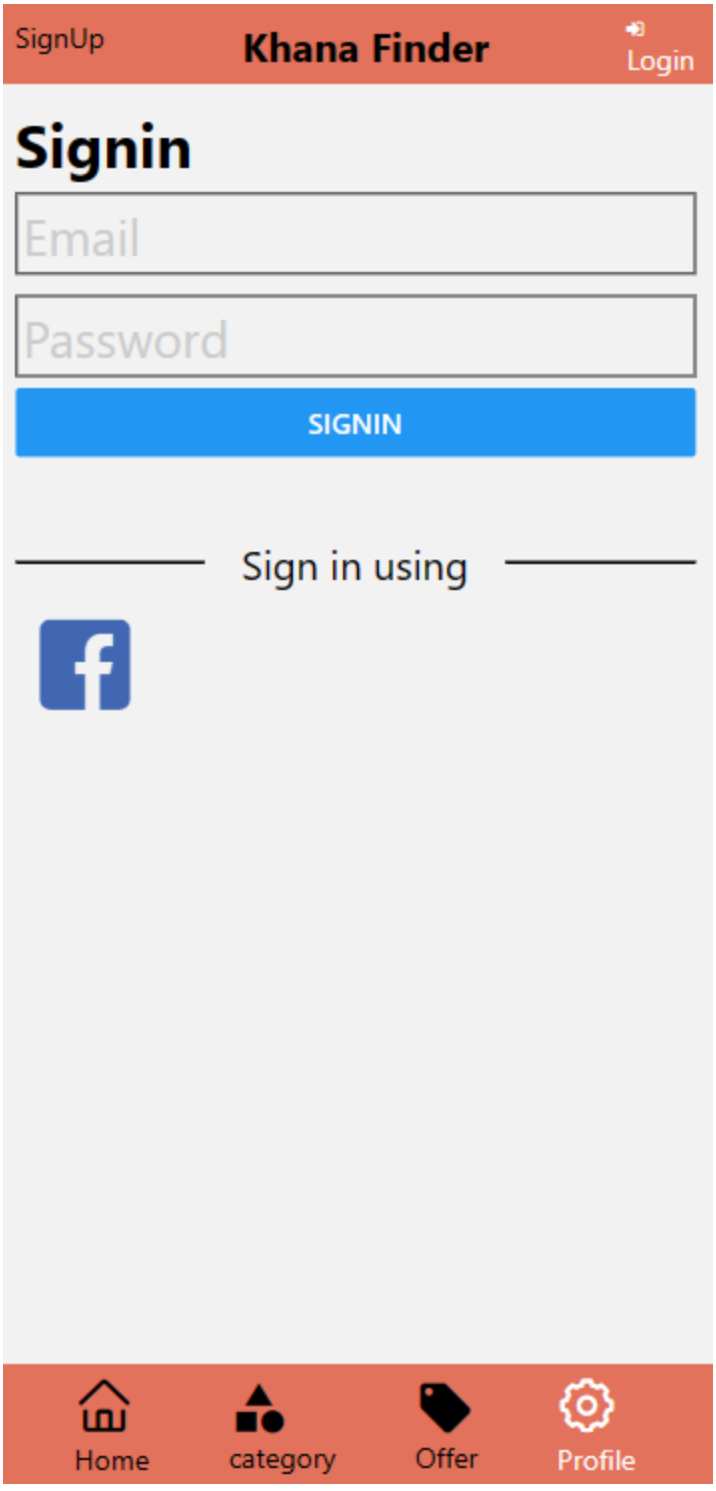

Homepage
Feeling hungry?
order online now
Categories
Non Veg
Veg
Liquors
Today's Special
Pizza
Lime and lemon
180162
1 plate
0
momo
Home
category
Offer
Profile

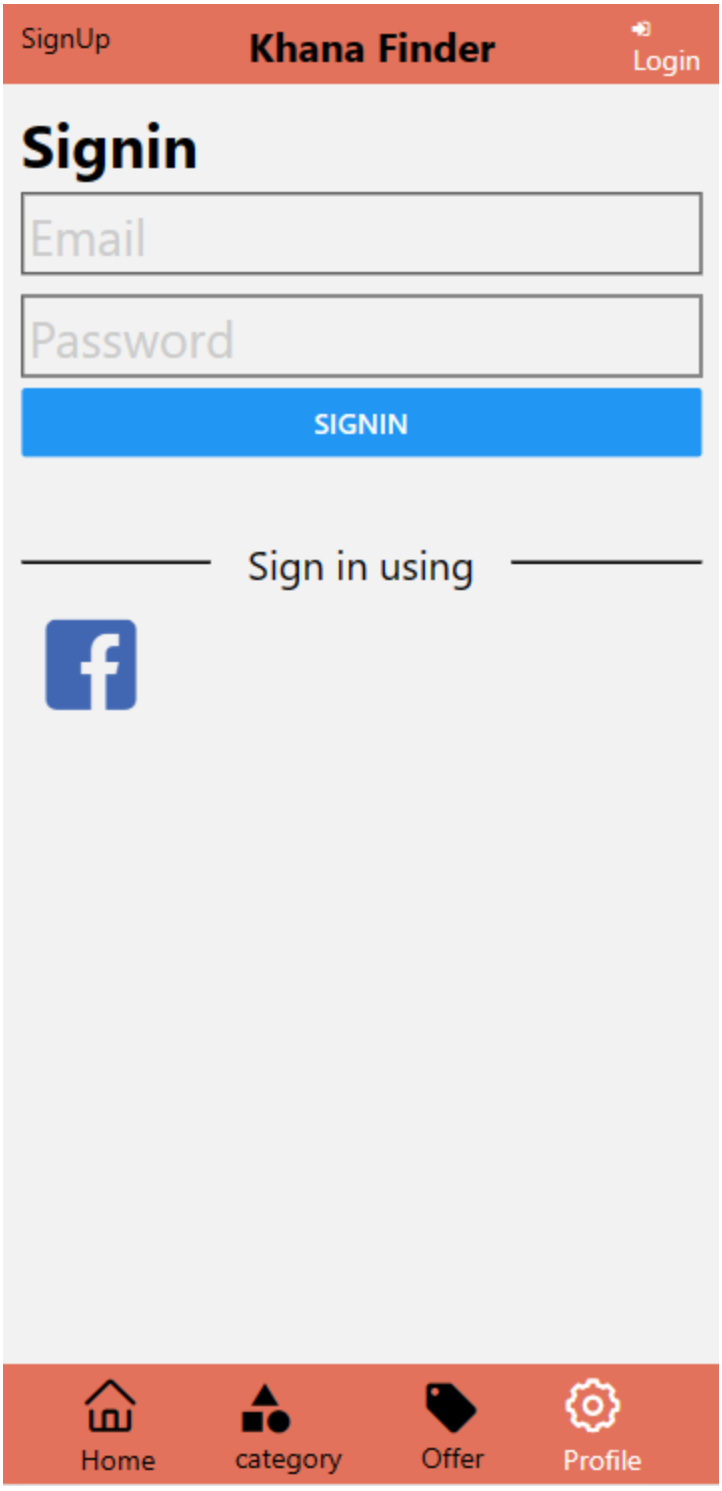

SearchItem
momo
momo
Lime and lemon
0%120
1 plate
momo
Lime and lemon
0%120
1 plate

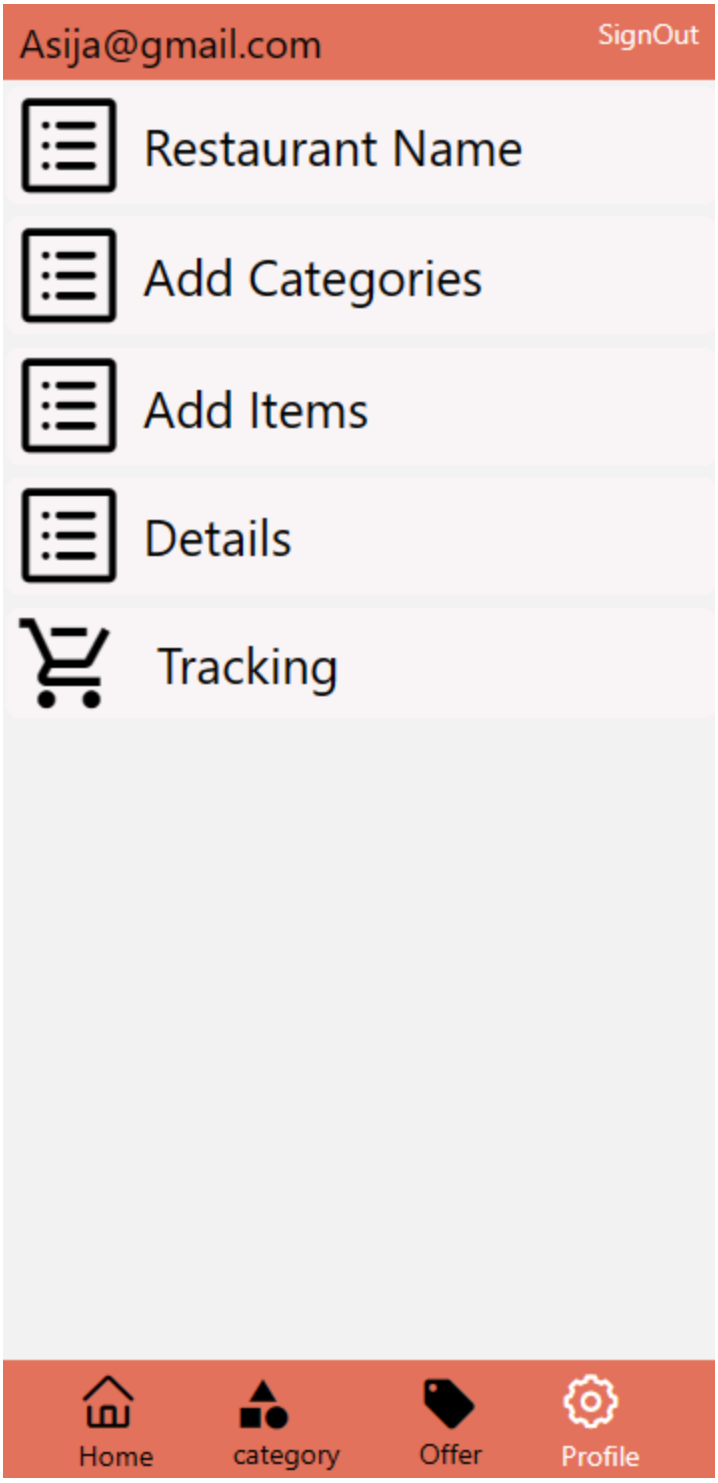

Khana Finder
Non Veg
Veg
Liquors
Grilled
Vegan
ColdDrink
Chinese Food
Order now
Home
category
Offer
Profile

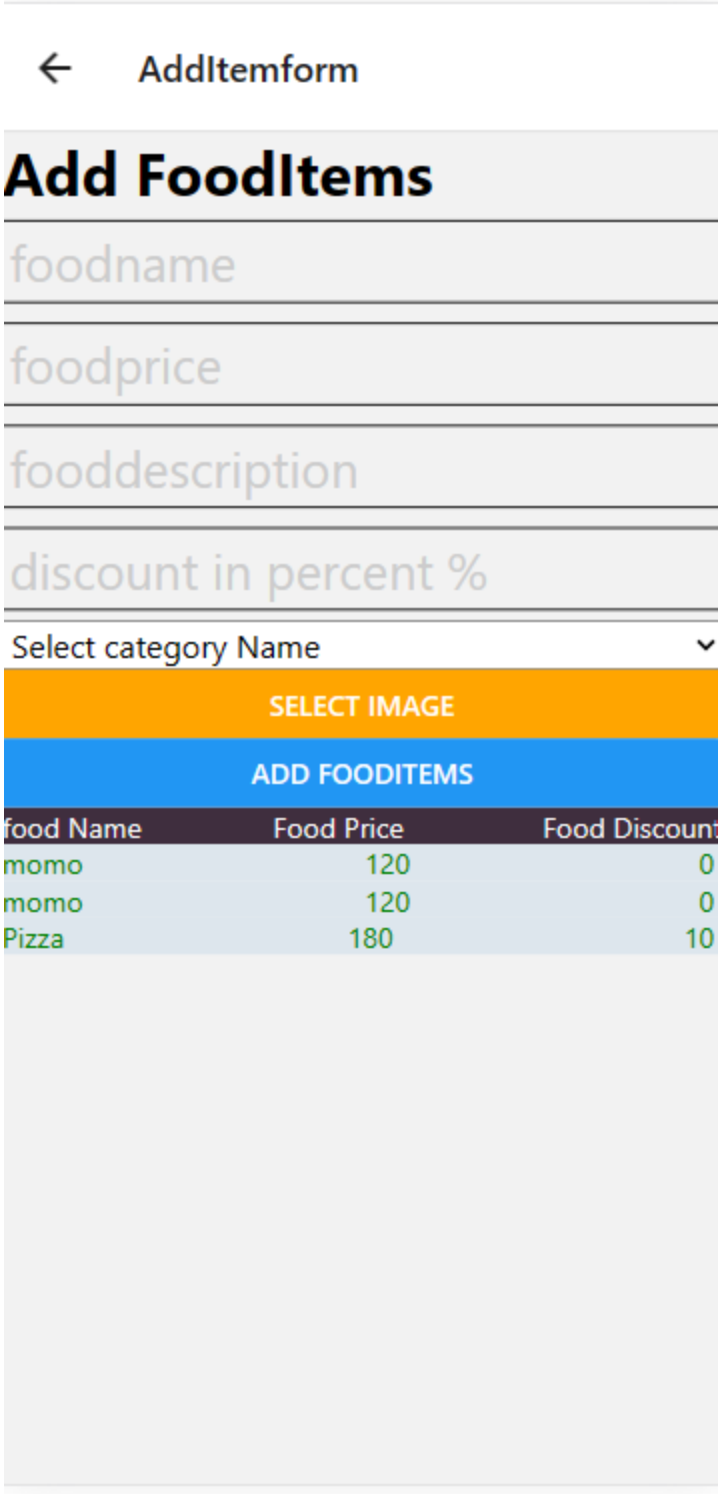

Veg
Veg
momo
Lime and lemon
120120
1 plate
0
momo
Lime and lemon
120120
1 plate
0

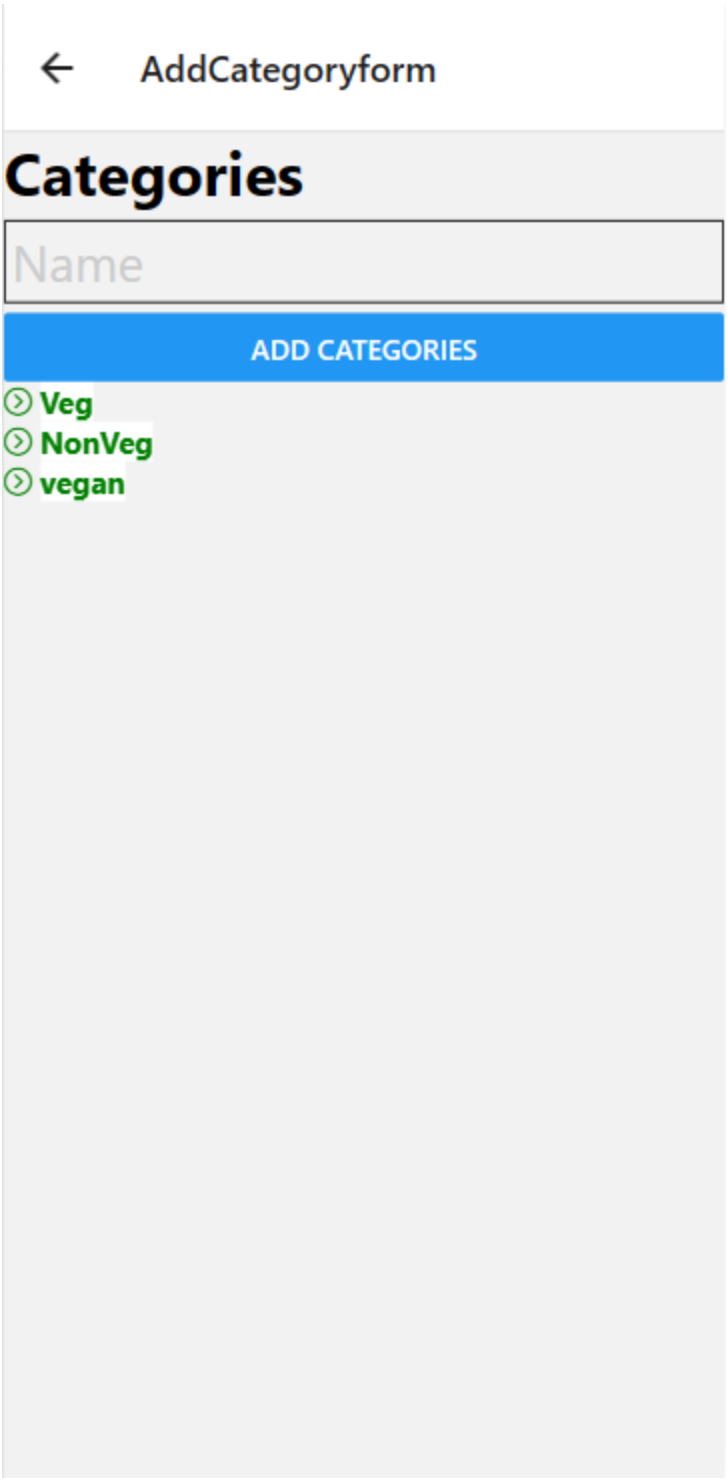

Khana Finder
Daily Offer
Grab your food now , offer will close soon.
10%
Pizza
Lime and lemon
180162
1 plate
0
Home
category
Offer
Profile

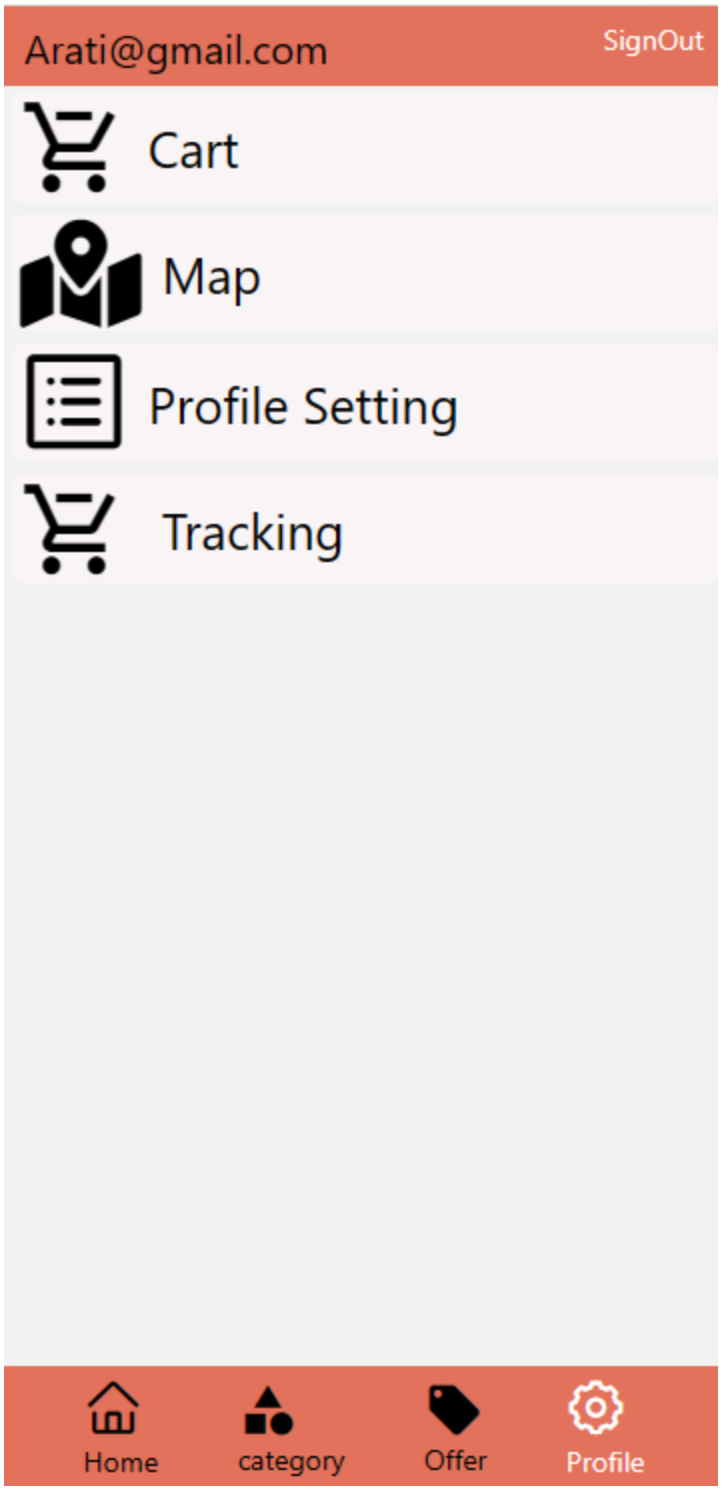

SignUp
Khana Finder
Login
Signin
Email
Password
SIGNIN
Sign in using
Home
category
Offer
Profile

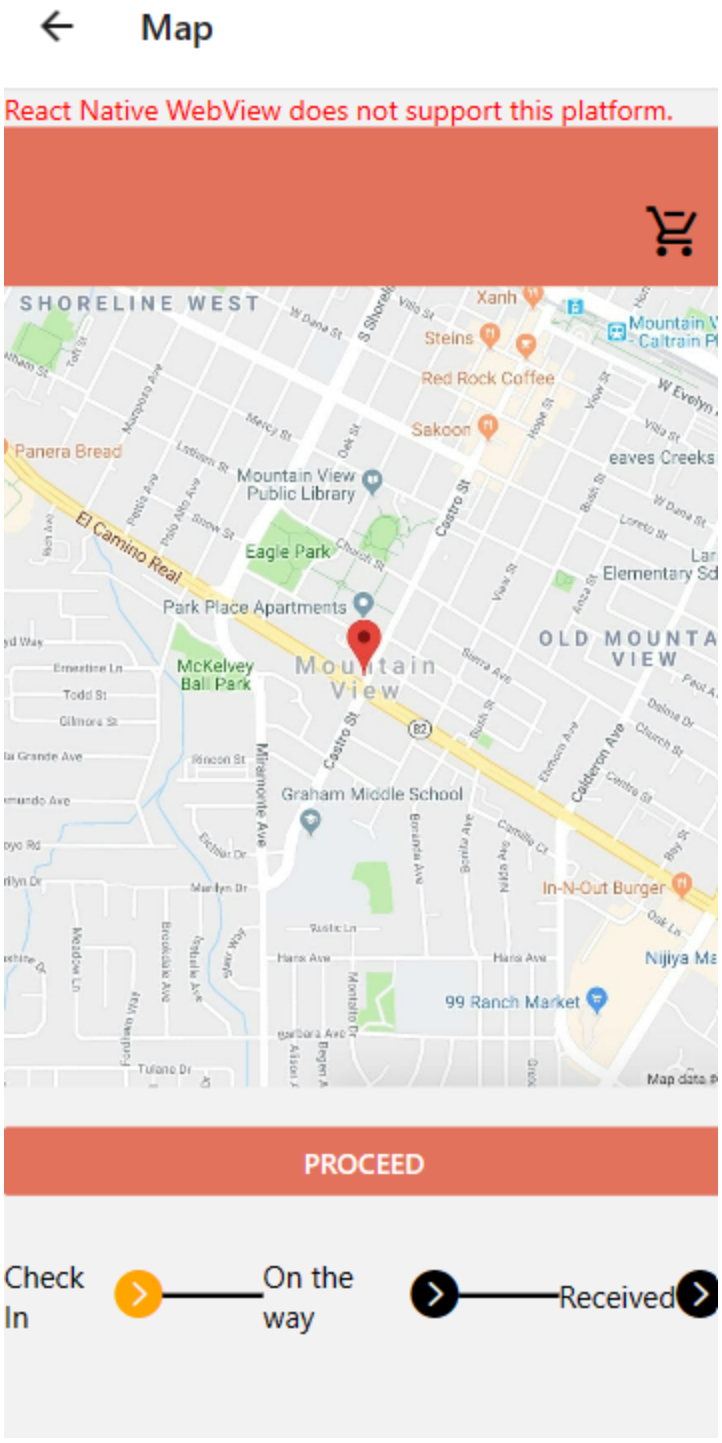

SignUp
Khana Finder
Login
Signin
Email
Password
SIGNIN
Sign in using
Home
category
Offer
Profile

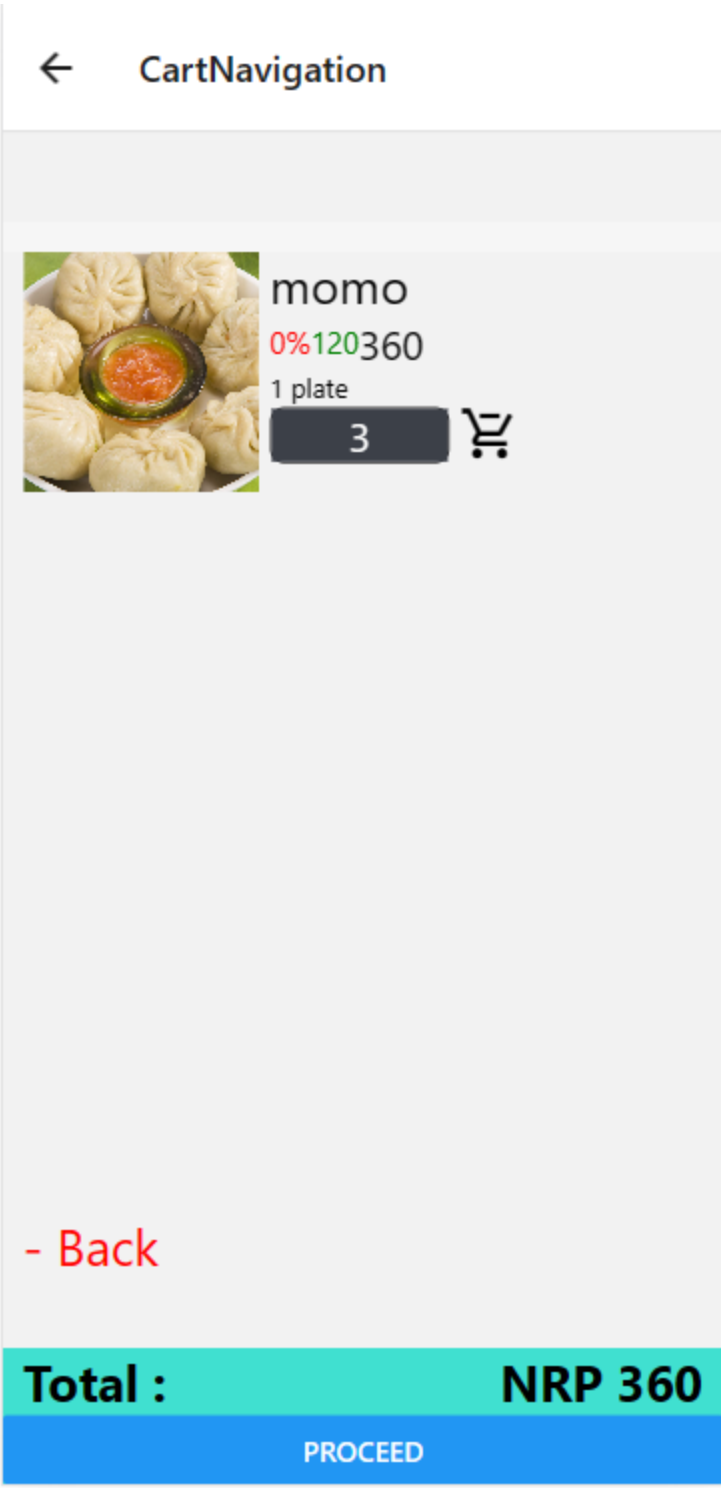

Asija@gmail.com
SignOut
Restaurant Name
Add Categories
Add Items
Details
Tracking
Home
category
Offer
Profile

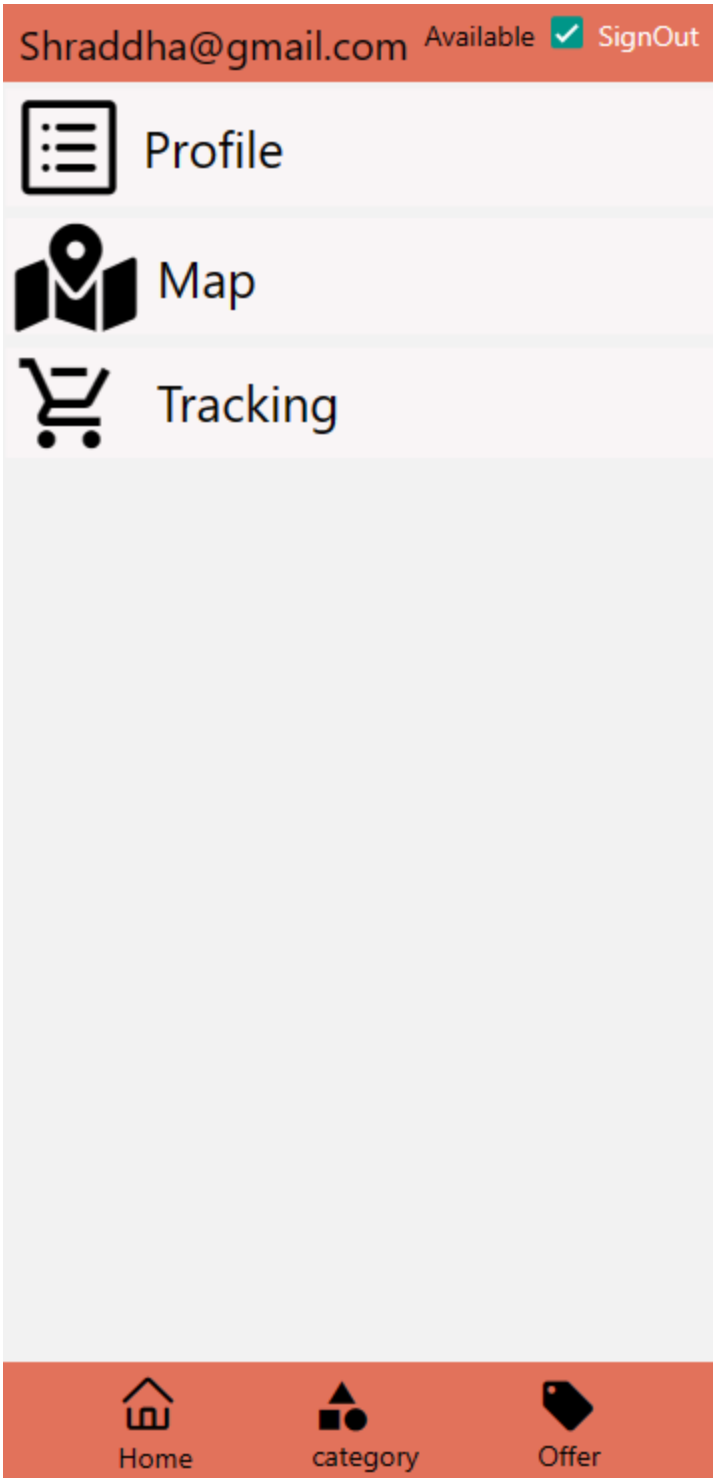

AddItemform
Add FoodItems
foodname
foodprice
fooddescription
discount in percent %
Select category Name
SELECT IMAGE
ADD FOODITEMS
food Name
Food Price
Food Discount
momo
120
0
momo
120
0
Pizza
180
10


AddCategoryform
Categories
Name
ADD CATEGORIES
Veg
NonVeg
vegan


Arati@gmail.com
SignOut
Cart
Map
Profile Setting
Tracking
Home
category
Offer
Profile


Map
React Native WebView does not support this platform.
SHORELINE WEST
Xanh
Steins
Red Rock Coffee
Sakoon
Panera Bread
Mountain View Public Library
Eagle Park
El Camino Real
Park Place Apartments
McKelvey Ball Park
Mountain View
OLD MOUNTAIN VIEW
Graham Middle School
In-N-Out Burger
99 Ranch Market
PROCEED
Check In
On the way
Received


CartNavigation
momo
0%120360
1 plate
3
- Back
Total :
NRP 360
PROCEED


Shraddha@gmail.com
Available
SignOut
Profile
Map
Tracking
Home
category
Offer

Code for Homepage

```
import { View,Image,StyleSheet,Text,Dimensions, ImageBackground, TouchableOpacity } from 'react-native'
import React, { useEffect, useState } from 'react'
import { Touchable } from 'react-native'
import Todayspecial from './Todayspecial'
import DiscountItem from './DiscountItem'
import { AntDesign , MaterialIcons } from '@expo/vector-icons';
import { ScrollView, TextInput } from 'react-native-web'
import NavbarTop from './NavbarTop'
import axios from 'react-native-axios';
const Homepage = ({navigation}) => {


   const screenWidth = Dimensions.get('window').width;

 return (
  <View>
     {/* Navbar Top Component (NavbarTop)*/}
     <NavbarTop navigation={navigation}/>

    {/* Navbar Bottom */}
    <View style={{
      height: 60,
      width: "100vw" ,
      backgroundColor:"#E2725B",
      position:'fixed',
      bottom:0,
      zIndex:10,


   }}>

   <View style={{
      flexDirection: "row",
      width:'100%',
      justifyContent:"space-evenly",
      alignItems:'center',paddingTop:5

   }}>
    <View>
                          <AntDesign name="home" size={30} color="white" /><Text style={{color:'white'}}>Home</Text></View>
```

```
            <TouchableOpacity     onPress={()=>{navigation.navigate('Categories')}}  >  <View>
<MaterialIcons                name="category"                size={30}                color="black"
/><Text>category</Text></View></TouchableOpacity>
      <TouchableOpacity  onPress={()=>{navigation.navigate('Offer')}} ><View> <MaterialIcons
name="local-offer"                                size={30}                                color="black"
/><Text>Offer</Text></View></TouchableOpacity>
       <TouchableOpacity    onPress={()=>{navigation.navigate('ProfileNavigator')}}  >  <View>
<AntDesign                name="setting"                size={30}                color="black"
/><Text>Profile</Text></View></TouchableOpacity>

   </View>

   </View>
   {/* Offer Image section */}
   <View style={css.offerimage1}>

      <Image style={css.offerbackground} source={require('../Public/order1.png')}>

      </Image>

   </View>

{/* Categories */}
 <View><Text style={{
   fontWeight: 'bold',
   fontSize: 20,
   marginTop:10,
   marginBottom:10,
   marginLeft:10

  }}>Categories</Text></View>


   <View style={{display:'flex' , flexDirection:'row' , alignItems:'center'}} >

    <View style={css.flex1} >
    <TouchableOpacity onPress={()=>{navigation.navigate('NonVeg')}}>
                                    <View     style={css.itemback}>     <Image     style={css.items}
source={require('../Public/ham.png')}></Image> </View>
          <Text style={css.itemtext}>Non Veg </Text></TouchableOpacity>
      </View>
      <View style={css.flex1}><TouchableOpacity onPress={()=>{navigation.navigate('Veg')}}>
                                  <View     style={css.itemback}>          <Image     style={css.items}
source={require('../Public/leaf.png')}></Image></View>
```

```
          <Text style={css.itemtext}>Veg </Text></TouchableOpacity>
        </View >
        <View style={css.flex1}><TouchableOpacity>
                                    <View     style={css.itemback}>              <Image     style={css.items}
source={require('../Public/wine.png')}></Image></View>
          <Text style={css.itemtext}>Liquors</Text></TouchableOpacity>
        </View >



    </View>


{/* Catlists */}
<Todayspecial/>

 {/* Offer Image section */}
 <View style={css.offerimage}>

<Image style={css.offerbackground} source={require('../Public/Discountpic.jpg')}>

</Image>

</View>

{/* sdff */}

<DiscountItem/>
<View style={{  marginBottom:45,}}></View>


    </View>

  )
}
const css = StyleSheet.create({

user:{
   width: 30,
height: 30,
},


txt:{
```

```
    fontSize: 20,
    fontWeight: "bold",
    color: "white"
},
serchbar:{
    width:110,
    border:'0.5px solid white',
    borderRadius:7,
    paddingLeft:3,
    fontSize:15

},

itemtext:{
    fontWeight: 500,
    fontSize: 15,
    textAlign:'center'
},
flex1:{
    flex:1,
    flexDirection:'column',
    alignItems:'center',

},
itemback:{
    width: 100,
    height: 100,
    backgroundColor:'#E2725B',
    justifyContent:'center',
    alignItems:'center',
    alignContent:'center',
    borderRadius:'9%'


},
offerimage1:{

    height:160,
    marginTop:45,
    borderRadius:20,
    backgroundImage:''

},
offerimage:{
```

```
    height:160,
    marginTop:10,
    borderRadius:20,

},
offerbackground:{

    height:140,
    resizeMode:'cover',
    margin:6,
    borderRadius:20,
    justifyContent:'center',
},


items:{
    width: 70,
    height: 70,



    },



})
export default Homepage
```

Code for Search Bar

```
import React, { useState } from 'react';
import { View, Text, ScrollView, StyleSheet, TouchableOpacity, TextInput, Image } from 'react-native';
import { AntDesign, MaterialCommunityIcons } from '@expo/vector-icons';
import axios from 'axios';

const SearchItem = () => {
  const [restName, setRestName] = useState();
  const [search, setSearch] = useState([]);


    axios
```

```
    .get('https://localhost:7222/api/Fetchdata/SearchRestaurant', {
      params: {
        Rest: restName,
      },
      headers: {
        'Content-Type': 'application/json',
      },
    })
    .then(response => {
      console.log('Response of Search:', response.data);
      setSearch(response.data);
    })
    .catch(error => {
      console.error('Error submitting order:', error);
    });


return (
  <View>
    {/* margin Top */}
    <View style={{ marginBottom: 40 }}></View>
    {/* Navbar */}
    {/* ... */}

    <View style={{ display: 'flex', flexDirection: 'row'  }}>
     <TextInput
        value={restName}
        onChangeText={setRestName}
        style={css.input}
        placeholderTextColor="black"
        placeholder='Search Restaurant'
      />

        <AntDesign name="search1" size={24} color="green" />

    </View>

    <ScrollView showsHorizontalScrollIndicator={false}>
      {/* line to separate */}
      <View style={css.line}></View>

      {/* Rendering search results */}
      {search.map((item, index) => (
        <View style={styles.container} key={item.fid}>
```

```
          <View style={{ flex: 1 }}>
            <Image style={styles.img} source={{ uri: item.fimage }} />
          </View>
          <View style={{ flex: 2, marginLeft: 5 }}>
            <Text style={{ fontSize: 24, fontWeight: '400' }}>{item.fname}</Text>
            <Text style={{ fontSize: 13, color: 'green' }}>{item.rname}</Text>
            <View style={{ flexDirection: 'row', alignItems: 'center' }}>
              <Text style={{ fontSize: 15, color: 'red' }}>{item.fdiscount}%</Text>
              <Text style={{ fontSize: 15, color: 'green' }}>{item.fprice}</Text>
              <Text style={{ fontSize: 20, color: 'black' }}>{item.ftotalprice}</Text>
            </View>
            <Text style={{ fontSize: 13 }}>1 plate</Text>
            <View style={{ flexDirection: 'row', alignItems: 'center' }}>
              <View style={styles.cart}>
                        <View style={{ backgroundColor: '#3C4048', flex: 1, alignItems: 'center',
justifyContent: 'center' }}>
                  <Text style={{ color: 'white', fontSize: 20 }}>{item.fquantity}</Text>
                </View>
              </View>
              <TouchableOpacity key={index}>
                 <MaterialCommunityIcons style={{ marginLeft: 5 }} name="cart-minus" size={30}
color="black" />
              </TouchableOpacity>
            </View>
          </View>
        </View>
      ))}

      {/* ... */}
    </ScrollView>
  </View>
 );
};

const css = StyleSheet.create({
 input: {
  height: 30,
  elevation: 10,
  borderBottomColor: 'green',
  backgroundColor: 'wheat',
  width:'90vw'
 },

 line: {
```

```
    marginTop: 5,
    width: '100%',
    height: 15,
    backgroundColor: "#f6f6f6"
  },
  img: {
    width: '100%',
    height: '100%',
    borderRadius: 10,
    resizeMode: 'cover'
  },
  container: {
    display: 'flex',
    flexDirection: 'row',
    width: '100%',
    height: 120,
    marginLeft: 10,
    marginBottom: 20
  },
  cart: {
    display: 'flex',
    flexDirection: 'row',
    width: 90,
    alignItems: 'center',
    justifyContent: 'space-evenly',
    borderWidth: 0.2,
    borderRadius: 7,
    backgroundColor: 'white'
  }
});


const styles = StyleSheet.create({
  line: {
    marginTop: 5,
    height: 15,
    backgroundColor: "#f6f6f6"
  },
  img: {
    width: '100%',
    height: '100%',
    elevation: 10,
    resizeMode: 'cover'
  },
```

```
  container: {
    flexDirection: 'row',
    width: '100%',
    height: 120,
    marginLeft: 10,
    marginBottom: 20
  },
  cart: {
    flexDirection: 'row',
    width: 90,
    alignItems: 'center',
    justifyContent: 'center',
    borderWidth: 0.2,
    borderRadius: 7,
    backgroundColor: 'white'
  }
});


export default SearchItem;
```

4.2 Budget Analysis

Item Description	**Estimated Cost (**In NRs.**)**
Flutter Framework	Free
XAMPP Software	Free
Internet	1500/month
Data Collection	15,000
Hosting charge	2000/month
Laptop	1,00,000

Table 1: Budget Analysis

CHAPTER 5
RESULTS AND TESTING

5.1 Introduction

Testing is the process of evaluating a system or component to ensure that it functions correctly and meets the intended requirements. It involves executing test cases, analyzing the results, and comparing them with expected outcomes to identify any discrepancies or defects.Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to find errors and weaknesses in a system. It is crucial to conduct thorough testing because a system that is only partially tested or not tested at all can lead to significant problems. Neglecting proper testing can result in high costs and risks for the system.

5.2 Testing Level and Test Plan

A Test Plan serves as a roadmap for the testing process and helps ensure that testing activities are organized, thorough, and aligned with project goals. In the test plan, we are concerned with two types of testing which are:

5.2.1 Functional Testing

This type of testing is tested within the software to ensure that it confirms with all the requirements. It's a way of checking if the system has been developed according to the business requirements.

5.2.2 Non-Functionality Tests

It focuses on the non-functional requirements of the software like performances, endurance, volume, scalability etc. Generally, there are four levels of software testing in Software Engineering, they are:

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

Due to the large number of units in our module, it was not practical to include the results of all individual unit tests in this report. However, rest assured that comprehensive testing has been conducted on every feature of all the modules. Upon those testing, most of the unit tests were passed, however upon further testing, some of the unit tests failed indicating that the project needs further optimization and bug fixes. Some of the unit tests are given in Table 2- Table 6.

Table 2:Unit Testing For Register Module

Test Case Description	Test Steps	Preconditions	Expected Result	Actual Result	Test Status
User successfully registers an account with valid credentials.	1. Navigate to the registration page. 2. Enter valid email address in the email field. 3. Enter a password in the password field. 4. Confirm the password in the confirm password field. 5. Fill in the remaining required fields. 6. Click on the "Register" button.	Users should already have installed the Khana Finder app.	1.Registration page is displayed. 2.Email field accepts the input. 3.Password field accepts the input. 4.Confirm password field accepts the input. 5.Required fields accept the input.	As expected	**Pass**
User provides invalid email format	1. Navigate to the registration page 2. Enter an invalid email format in the email field 3. Complete the remaining fields 4. Click on the "Register" button	Users should already have installed Khana Finder application	1.Registration page is displayed 2.error message pop up indicating invalid format 3.Required fields accept the input 4.User receives an error message indicating invalid email format	As expected	**Pass**

(Source: Group study, 2023)

Table 3:Unit Testing for Login Module

Test Case Description	Test Steps	Preconditions	Expected Result	Actual Result	Test Status
Check whether the login button is clickable or not	1. Goto Khana Finder login page. 2. Enter the credentials and click the login button.	Users should already have registered in the Khana Finder account.	Login Button should get clicked.	As expected	**Pass**
Check whether user can login with invalid credentials	1. Goto Khana Finder login page. 2. Enter the wrong credentials and click the login button.	Users should already have registered in the Khana Finder account.	Users should not get logged in.	As expected	**Pass**
Check whether user can login with valid credentials	1. Goto Khana Finder login page. 2. Enter the valid credentials and click the login button.	Users should already have registered in the Khana Finder account.	Users should get logged in.	As expected	**Pass**

(Source: Group study, 2023)

Table 4:Unit Testing for Add to Cart Module

Test Case Description	Test Steps	Preconditions	Expected Result	Actual Result	Test Status
User adds a food item to the cart and verifies the cart contents	1. Login to the khana finder application 2.Browse available menu or search for a specific food item 3. Select food from menu and click on the "Add to Cart" button 4. click on the cart icon to view the cart contents 5. Verify that the added food item is correctly displayed	1.User is logged into the khana finder app. 2.User has a valid session. 3.The cart feature is accessible & operational.	1.User is logged in successfully. 2.Food menu is displayed with relevant search results 3.Food item is added to the cart 4.Cart page is displayed with the added food item listed 5.Food items are correctly displayed	As expected	**Pass**

(Source: Group study, 2023)

Table 5:Unit testing for Payment Module

Test Case Description	Test Steps	Preconditions	Expected Result	Actual Result	Test Status
Successful payment with valid details	1. Navigate to the checkout page 2. Enter valid payment details 3. Click on the "Proceed Payment" button 4. Verify the order confirmation message	Add to cart should have at least 1 item to be purchased.	1.The checkout page is displayed 2.The payment details are accepted and validated 3.The payment is successfully processed 4.An order confirmation message is displayed	As expected	**Pass**
Payment with invalid credit card details	1. Navigate to the checkout page 2. Enter invalid credit card details 3. Click on the "Proceed Payment" button	Add to cart should have at least 1 item to be purchased.	1.The checkout page is displayed 2.an error message is displayed 3.The payment is not processed and an error message is displayed	As expected	**Pass**

(Source: Group study, 2023)

Table 6:Unit Testing for Checkout Module

Test Case Description	Test Steps	Preconditions	Expected Result	Actual Result	Test Status
Check if payment options applicable to order is displayed at checkout	1.visit the Ecommerce site 2.Select the product and add to cart 3.Click on checkout button	User must have items at add to cart	Payment options should be displayed at checkout	As expected	**Pass**
Check if the user is redirected to the homepage after check out.	1.visit the Ecommerce site 2.Select the product and add to cart 3.Click on checkout button	User must have items at add to cart	After checkout user is redirected to home page	As expected	**Pass**

(Source: Group study, 2023)

CHAPTER-6
CONCLUSION

In conclusion, **‘Khana Finder’**, is a flutter-based food delivery application that provides an efficient and user-friendly platform for customers to order food from their favorite restaurants. Hence, this project can be beneficial in daily life of people andwill save a lot of time by showing the list of food menus with the accurate price of different restaurants with daily updates and also help us to avoid long queues at the counter and can track their orders in real time.

We are confident that **‘Khana Finder’** not only meets the needs of customers but also provides a platform for restaurants to expand their customer base and grow their businesses. With the increasing demand for food delivery services, **‘Khana Finder’** is poised to become a leading food delivery application in the market.

CHAPTER:7
LIMITATIONS AND FUTURE ENHANCEMENTS

Limitation

- Required Internet Connection

  This system can only be used and performed under internet connection. Customers have to connect to wireless internet connection or use their phone data in order to place their order.

- Limited Restaurant options

  While food delivery apps aim to provide a wide variety of restaurant choices, there may still be limitations in terms of available options. Some popular or niche restaurants may not be partnered with the app, restricting users choices and limiting their access to specific cuisines or establishments.

- Environment impact

  The increasing use of food delivery apps has raised concerns about the environmental impact of increased packaging waste and emissions from delivery vehiclesSingle-use packaging, excessive use of plastic, and increased traffic congestion due to delivery services contribute to environmental challenges.

REFERENCE

[1]. Kirti Bhandge, Tejas Shinde, Dheeraj Ingale, Neeraj Solanki, Reshma Totare,"A Proposed System for Touchpad Based Food Ordering System Using Android Application", International Journal of Advanced Research in Computer Science Technology (IJARCST 2015).

[2] Bonnie Eisenman. Learning React Native: Building Native Mobile Apps with JavaScript; Chapter 1; 2015.

[3] H. Orpilla , "3 advantages of online food ordering for customers,"Star Micronics POS Blog, 22-Dec-2020. [Online]. Available: http://www.starmicronics.com/blog/advantages-of-online -food-ordering-for-customers/. [Accessed: 26-Dec-2021].

[4]. Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke, S. R Lahane,"Digital Ordering System for Restaurant Using Android", International Journal of Scientific and Research Publications 2013.

[5]. Regmi, D. Online food order system (master's thesis). Nepal Engineering College, Nepal.

[6]. Amit Shankar, Charles Jebarajakirthy, Preeti Nayal, Haroon Iqbal Maseeh, Aman Kumar, Achchuthan Sivapalan, Online food delivery: A systematic synthesis of literature and a framework development, International Journal of Hospitality Management, Volume 104, 2022, 103240.