CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

FOOD WASTE MANAGEMENT

By

19BAI1045	P. SAI PAVAN GURU JAYANTH
19BCE1289	B. LAXMI NITIN SINGH
19BCE1843	V. KESHAVA SAI KUMAR
19BPS1040	P. KARTHIK
19BPS1079	Y.M JASHWANTH KUMAR
19BPS1087	T. PUNEETH

B. Tech Computer Science and Engineering

Submitted to

DR. HARINI S

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

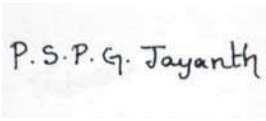
April 2022

DECLARATION

I hereby declare that the report titled "**Food Waste Management**" submitted by us to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Dr. Harini S**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

B. Laxmi Nitin Singh 19BCE1289

V. Keshava Sai Kumar 19BCE1843



Potti Sai Pavan Guru Jayanth 19BAI1045

Puneeth Telugu 19BPS1087

Y M Jashwanth Kumar 19BPS1079

P. Karthik 19BPS1040

CERTIFICATE

Certified that this project report entitled "**Food Waste Management**" is a bonafide work of **B. Laxmi Nitin Singh (19BCE1289), V. Keshava Sai Kumar (19BCE1843) , Potti Sai Pavan Guru Jayanth (19BAI1045), Puneeth Telugu (19BPS1087), Y M Jashwanth Kumar (19BPS1079), P. Karthik (19BPS1040)** and they carried out the Project work under my supervision and guidance for CSE1901 - Technical Answers to Real World Problems (TARP).

Dr. Harini S

SCOPE, VIT Chennai

ACKNOWLEDGEMENT

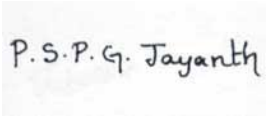
In preparation of our project for CSE1901 Technical Answers for Real World Problems J-Component, we had to take the help and guidance of our professor Dr. Harini S, who deserves our deepest gratitude for giving us good guidelines for assignment throughout numerous consultations. We would also like to expand our gratitude to all those who have directly and indirectly guided us in writing this paper.

We also thank Vellore Institute of Technology for giving us the required resources and the opportunity to make this project, and the consent to include copyrighted pictures as a part of our paper.

Many people, especially our classmates have made valuable comment suggestions on our paper which gave us an inspiration to improve the quality of the assignment.

B. Laxmi Nitin Singh 19BCE1289

V. Keshava Sai Kumar 19BCE1843



Potti Sai Pavan Guru Jayanth 19BAI1045

Puneeth Telugu 19BPS1087

Y M Jashwanth Kumar 19BPS1079

P. Karthik 19BPS1040

ABSTRACT

A drastic increase can be seen in food waste. As per data given by Food and Agriculture Organization (http://www.fao.org/food-loss-and-food-waste/flw-data), 1/3rd of food produced for human consumption is wasted globally, which accounts for almost 1.3 billion tons per year. On the other hand, also as per WHO 20% of the population face extreme food shortages. Hence there is a need to come up with a solution that can avoid food waste & can help feed the needy.

To Overcome this issue of food wastage we would like to make a mobile application to provide the leftover food to the people who are needy. By the mobile application we would like to bridge the gap between people /restaurants/ cafes through NGOs to the people who are needy. Along with the application we would like to addon a novelty to the project by providing a "**food quality tester**" using Arduino and sensors so that when the delivery agent who picks up the food from house/restaurants/cafes tests the quality of the food so that a healthy and a hygienic food is provided to the needy.

CONTENTS

	Declaration	i
	Certificate	ii
	Acknowledgement	iii
	Abstract	iv
1	Introduction	7
	1.1 Objective and goal of the project . . . . …….	7
	1.2 Problem Statement. ….	7
	1.3 Motivation . . . . ……...	7
	1.4 Challenges . . . . …….....	8
2	Literature Survey	8
3	Requirements Specification	12
	3.1 Hardware Requirements . . . . …	12
	3.2 Software Requirements	12
4	System Design	12
5	Implementation of System	13
	5.1 Hardware Approach	13
	5.2 Measurement Setups	13
	5.3 Process Steps	13
	5.4 Quality Tester Block Diagram	14
	5.5 Software Modules	14
	5.6 Flowchart	16
6	Results & Discussion	17
7	Conclusion and Future Work	20
8	References	20
Appendix		22

1. Introduction

1.1 Objective and goal of the project

The main objective of the project is to overcome the issue of the food waste by developing a mobile application to bridge the gap between people /restaurants/ cafes through NGOs to the people who are needy and designing a hardware "**food quality tester**" to ensure the quality of food.

1.2 Problem Statement

Food waste management in India is becoming a critical problem due to the continuous increase of the Indian population. Indians waste the maximum amount of food as much as the whole of the UK consumes – a data point which will not be most indicative of our love of surfeit, because it is of our population. Most of the food is wasted in weddings, canteens, hotels, social and family functions, and households. Still, food wastage is a horrendous issue, so is food waste management in India. Our streets, garbage bins, and landfills are spoiling our environment and have sufficient evidence to prove it.

1.3 Motivation

On Average, 40% of food produced were being disposed of. This means that 7.5 tons of food are discarded daily. Some 84.7% of the whole waste material recorded was thrown within the bin, whereas the remainder was either fed to the poor or some animals. A big portion of the waste material binned was still in edible condition. If the edible waste material generated is used, we tend to estimate that it might feed a minimum of 2000 individuals daily.

Solely a pair of the ten shops surveyed were part waste aware, i.e., they separated the edible from the inedible, and ensured that food in condition reached empty stomachs. One among them disposed of their food at twelve noon, in order that it might be fed to the cows within the space. The other claimed to administer away all edible waste material to the native laborers and employees, for free, at the tip of the day.

1.3 Challenges

Google Maps API billing was unable to happen to get directions in the delivery module also firebase real-time database had to be done in testing mode, but it had 1 month for testing mode.

For testing the Android Studio was used but it was very slow and took a large space and sometimes was not responding correctly.

2. Literature Survey

[1] The next step in sustainable dining: the restaurant food waste map for the management of food waste Ludovica Principato, et al: This paper has studied the Identification of phases in which food waste occurs in restaurants and to identify mitigation activities of food waste generation processes.
Scope for improvement: Could explore and analyze out-of-home waste from both the client and business perspective according to the proposed RFWM, to target specific attitudes and behavioural changes, and to quantify the impacts of the suggested changes.

[2] Food waste matters - A systematic review of household food waste practices and their policy implications Karin Schanes, et al: Why food waste occurs in the private household and food supply chain. Also identifying the contribution of different social ontologies.
Scope for improvement: Requires the tests and assesses the effectiveness and impact of different policy measures and other interventions on food waste practices.

[3] Sustainable value co creation at the Bottom of the Pyramid: using mobile applications to reduce food waste and improve food security Chrysostomos Apostolidis, et al: How food waste mobile apps can support sustainable value co-creation at the Bop.
Scope for improvement: There is scope for exploration of the role of not-for-profit organizations and charities currently supporting food insecure households, and their potential to adopt or collaborate with technologies such as FWMAs to make the process easier and more trustworthy, and potentially avoid stigma and opportunistic behaviors by businesses and
customers.

[4] Generation and prevention of food waste in the German food service sector in the COVID-19 pandemic – Digital approaches to encounter the pandemic related crisis Christina Strotmann, et al: How far has the pandemic affected the generation of food waste and the implementation of prevention measures? How does the use of digital technology, which has been increasingly accessed since the beginning of the pandemic, offer opportunities to further reduce food waste in the sector?

Scope for improvement: To develop more tailored digital solutions for the prevention of food waste, the interdependency between all actors of the food supply chain also needs to be investigated to expose the diverse needs.

[5] Food sharing, redistribution, and waste reduction via mobile the methods: A social network analysis John Harvey et al: How a popular food sharing mobile application named OLIO has resulted in a tremendous reduction in food waste in the UK by reconfiguring traditional supply chain roles of consumers.
Scope for improvement: We should also examine the socio enlarge amounts of technology assisted food sharing donors, volunteers, and recipients.

[6] FOOD QUALITY MONITORING SYSTEM BY USING ARDUINO B. Ravi Chander et al: The Quality of the food needs to be monitored and it must be prevented from rotting and decaying by the atmospheric factors like temperature, humidity and dark.
Scope for improvement: No proper implementation is done; more sensors could be added to detect the spoilage of the food early and correctly. More precise threshold values could be used to get accurate result the7] Food waste management in ethnic food restaurants Viachaslau Filimonau et al: This paper contributed to knowledge with an exploratory study of food waste and its management in Chinese cuisine restaurants operating in the UK.
Scope for improvement: This study is empirical in nature and based on of peopleervations and measurements of the phenomenon of food waste in ethnic food restaurants. s should examine the phenomenon of food waste and its management in restaurants specializing in other ethnic cuisines.

[8] Review on Efficient Food Waste Management System Using Internet of Things Bharath Kumar et al: This article provides an extensive survey of food waste management techniques/methods.
Scope for improvement: This article focused on just highlighting the methods/ways instead of elaborating the methods which could be easily understandable.

[9] Proposal of a LoRaWAN - based IoT System for Food Waste Management Evjola Spaho et al: This paper proposed a low-cost food waste management system, capable of converting food wastes to compost. To make the management system safe, efficient, and less time consuming.
Scope for improvement: This application targets only the large amount of food waste, which comes from restaurants, etc., this can address the less amount of food waste produced in the households.

[10] Food waste management in hospitality operations: A critical review Viachaslau Filimonau et al: This paper provides a critical, analytical account of the literature on hospitality food waste made from the viewpoint of hospitality managers
Scope for improvement: Further they can harmonize the different methodologies for hospitality food waste quantification and characterization.

[11] A Methodology for Sustainable Management of Food Waste Guillermo Garcia-Garcia: This paper takes a holistic approach with the aim of achieving a better understanding of the different types of food waste, and using this knowledge to support informed decisions for more sustainable management of food waste.
Scope for improvement: Evaluating the recritical, analyticale management alternatives is a complex task. To meet this LoRaWANey should develop an analysis method and associated figures of merit to allow quantitative comparison of waste management alternatives, with a focus on the environment.

[12] A Novel Approach of Online Food Management System for Needy People During Covid 19. Aishwarya Sadhu et al: The aim of this paper is to develop an online food waste management application.
Scope for improvement: This web application has targeted only the restaurants, this application can be extended for the households and different places like cafes, etc.

[13] Food waste management innovations in the foodservice industry Carlos Martin Rios, et.al: Determine innovative practices for food waste management in the food service sector, as there is limited empirical studies as to how food service firms address innovative management approaches to food waste.
Scope for improvement: They may address such tools and concepts, as well as different types of innovations and sources of co-operation between collaborative firms and traditional food service organizations.

[14] Food waste management in the catering industry: Enablers and interrelationships Zepeng Wu, et.al: To analyze the degrees of importance and interrelationships of food waste enablers using holistic cause and effect diagrams and priority ranking.
Scope for improvement: Can include different actors in the entire supply chain and can also explore the behavior of customers and managers in different cultures to allow further trans variability of the findings presented.

[15] Household food waste management: socio ecological dimensions Vasiliki Aitsidou, et.al: This paper is to estimate respondents' socio ecological consciousness (SEC) (Knowledge, perceptions, behavior, habits and feelings) concerning household food waste (HFW) and to identify factors that influence attitude concerning HFW.
Scope for improvement: The study is that it is a subjective self-reported survey, in which respondents tend to overestimate their attitude and another one is to regard the small but particularly important (because of PPC operation) area in Greece, so the results cannot be generalized across the country. However, there is a necessity for more data, both qualitative and quantitative, regarding in Greece.

[16] Food Wastage Reduction Mobile Application M.S.Elavarasan, et.al: This project is used to manage wastage foods in a useful way. Every day the solutions inserting lots of food. So, we must reduce that food wastage problem online.

Scope for improvement: In future work, there was no standard food information system on food packages that gives the user the information of both the name of the food, as well as its expiry date. The viable improvement would be getting the food name from the product barcode and reading the expiry date using OCR tools.

[17] Environmental Impact Assessment of Food Waste Management Using Two Composting Techniques Aisha Al Rumaihi, et.al: This paper addressed the Life cycle assessment of food waste.
Scope for improvement: The model will involve expansion of the model to include the impacts of wastewater return and sludge nutrient benefits. Furthermore, it is recommended to conduct further LCA comparisons with other food waste treatment scenarios, such as pyrolysis or other composting techniques including vermicomposting.

[18] How to manage and minimize food waste in the hotel industry: an exploratory research VeraAmicarelli, et.al: This paper aims to provide a better understanding of food waste management trends in the hotel industry as well as to highlight hotel kitchens and hotel food services weaknesses and opportunities to minimize food waste.
Scope for improvement: Further, formal agreements should be underwritten between hotels and food banks, with the aim of generating a food network that holds together companies, retailers, and charities. In terms of solutions inside-the-unit, several efforts are needed to improve processes at administrative, kitchen and service level. Indeed, considering the importance of involving technology in the hotel sector, units should have more support from national authorities both in terms of funding and consultancy.

[19] Testing the Effects of Food Quality, Price Fairness, And Physical Environment on Customer Satisfaction in Fast Food Restaurant Industry (Jalal Hanaysha): Factors affecting Customer Satisfaction were briefly discussed and analyzed toward international fast food restaurant brands in the east coast region of Malaysia.
Scope for improvement: The research conducted was on a limited number of people and restricted the location of research.

[20] System to Reduce and Manage Waste Food (Rucha Jadhav, Priyal Kulkarni, Neha Kumari, Komal Pagere): The growing trouble by constructing a sustenance waste diminishment chain of importance with source lessening and nourishment recovery as the actual manifestations of interest
Scope for improvement: The proposed system can be made available to the general public bringing a bridge between the NGOs and farmers nationwide. This could actually bring a change and feed the starving nation.

3. Requirements Specification

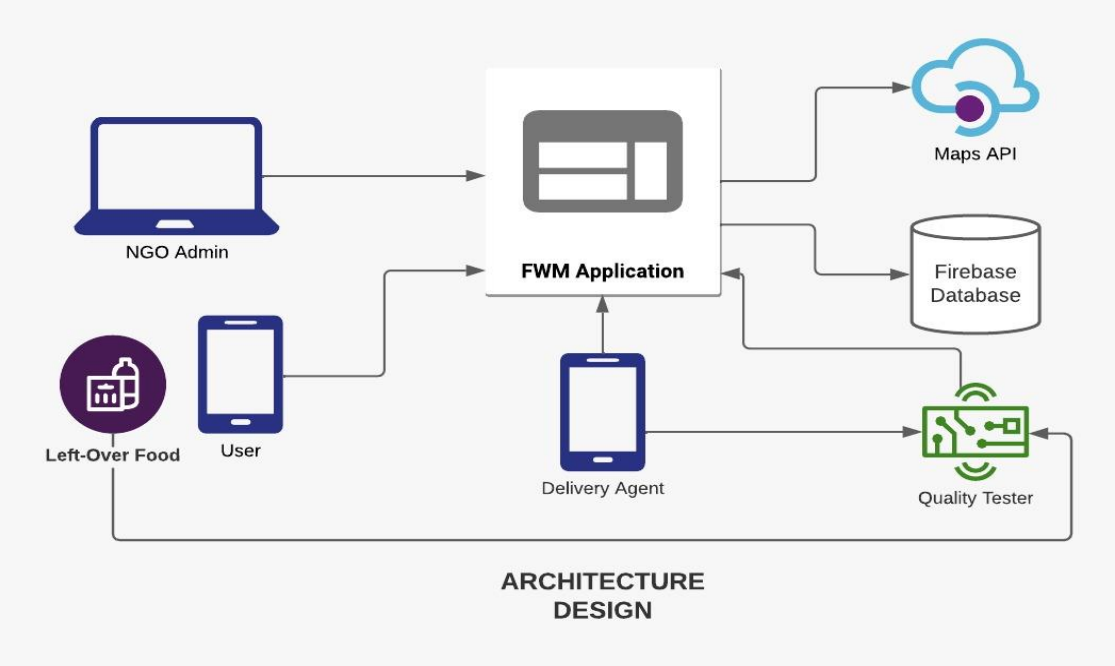
3.1 Hardware Requirements

- Arduino UNO
- MQ3 Sensor
- DHT 11 Sensor
- Jumper wires

3.2 Software Requirements

- React JS
- React Native
- Firebase DB (back-end)
- Visual Studio Code
- Windows 7/8/9/10/11

4. System Design



5. Implementation of System

5.1 Hardware Approach

- All the required components mentioned in the hardware requirements section were gathered.
- All the components were connected to the Arduino board with the help of jumper wires and were
- The Arduino was connected to PC using the USB cable
- The coding was done in the Arduino IDE and the script was uploaded in IDE.
- The output was verified in the Serial Monitor.

5.2 Measurement Setups

- **MQ3 Gas Sensor:** MQ3 is a gas sensing module, which is used to measure methane gas in the atmosphere. It contains Gas sensing layer, which is made up of SnO2. SnO2 is sensitive to gases like LPG, CH4, H2, CO, Alcohol, and smoke. As the decaying food emits methane gas (CH4), the MQ4 sensor can be used to measure this gas to monitor food quality.

- **DHT11 Sensor:** DH11 sensor to know the temp and humidity and based on the threshold values we will be categorizing them.

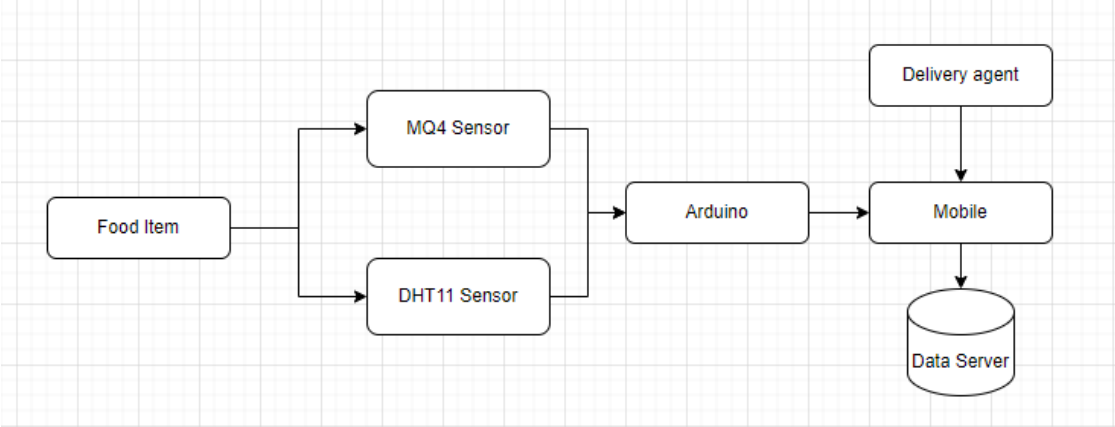
5.3 Process Steps

- Initially when the script is uploaded in the Arduino IDE the DHT11 and the MQ3 gas sensor started sensing the temperature, humidity, and gas values respectively under the normal environment conditions.

- If temperature, humidity, and the gas values sensed by the sensors are within the threshold limit the food is categorized as fit to consume and the output is displayed on the Serial Monitor.

- Else if any value of temperature, humidity or gas are not within the threshold range the food is categorized as not fit to consume and the output is displayed on Serial Monitor.

5.4 **Quality Tester Block Diagram**



5.5 **Software Modules**

Customer Module

Customer Module is developed using React Native and firebase real-time database was used to store the data. The customer Module contains the following pages

- **Signup page:** A new customer can register using the signup page to start using the FWM app.
- **Login page:** Once the customer registers he/she could login to the app using the credentials.
- **Profile page:** For the first-time login of the customer the customer will be asked for his name and mobile number.
- **Home page:** This is the front page of our app where the customer can add the food item which is leftover by giving the details of the leftover food and click on the continue to continue to the Address page.
- **Address page:** Once the user gives permission to access the location current location of the user will be pointed on the map and the location details will be shown and the user will be asked for a detailed address to help delivery agent reach the customer.
- **Confirm page:** Once the customer saves the address, he will be redirected to confirm page where the details of the delivery agent will be displayed once assigned by the NGO administrator.

NGO Module

NGO Module is developed using ReactJS and firebase real-time database was used to store the data. The NGO Module contains the following pages

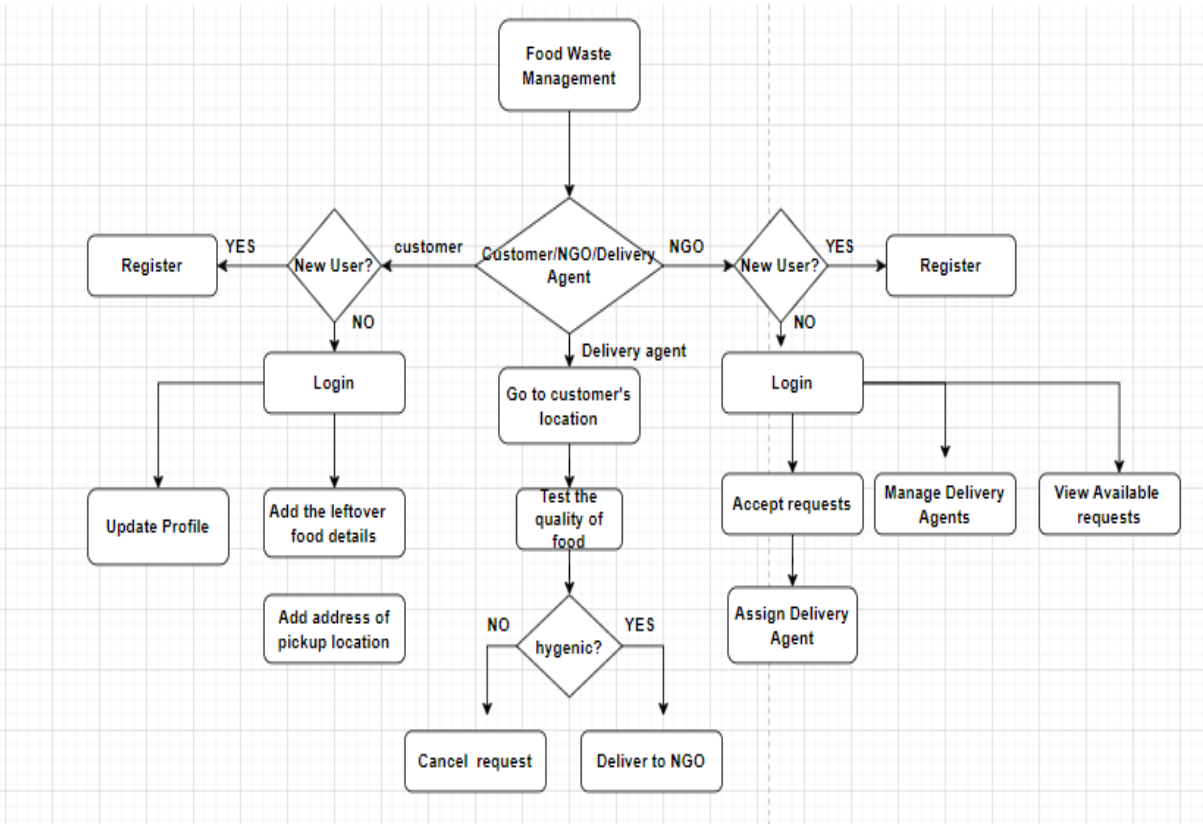
- **Signup page:** A new admin can register using the signup page to start using the FWM app.
- **Login page:** Once the admin registers, he/she could login to the app using the credentials.
- **Home page:** This is the front page of our website where the admin will show with the available request's admin can accept the request and will be redirected to Assign page.
- **Assign page:** Here the admin will be shown with the list of available delivery agents, and he could assign the delivery agent to the customer.

Delivery Module

Delivery Module is developed using React Native and firebase real-time database was used to store the data. The Delivery Module contains the following pages

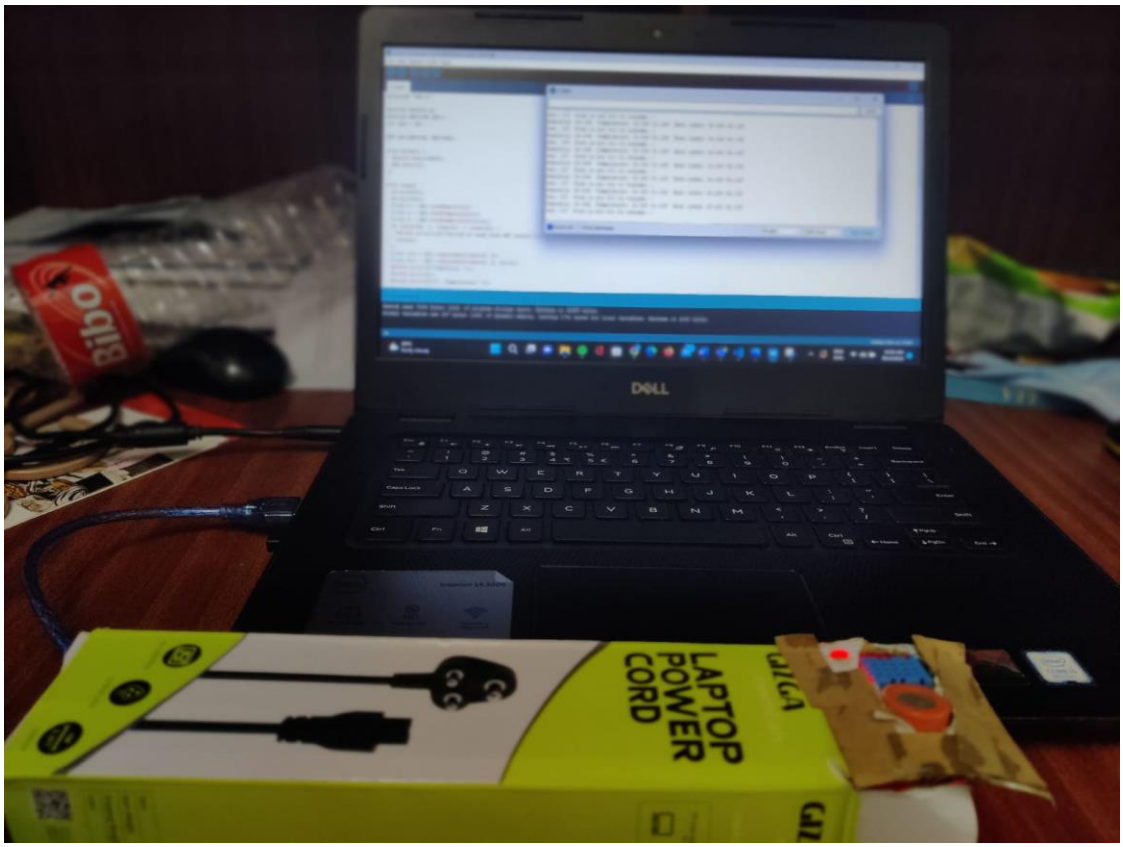
- **Signup page:** A new delivery agent can register using the signup page to start using the FWM app.
- **Login page:** Once the delivery agent registers, he/she could login to the app using the credentials.
- **Profile page:** For the first-time login of the delivery agent the agent will be asked for his name and mobile number.
- **Home page:** Here the delivery agent will be show with the assigned customer location pointed on the map and the details of the assigned customer will be and once the agent clicks on the pin, he can choose the google directions to reach the customer location.

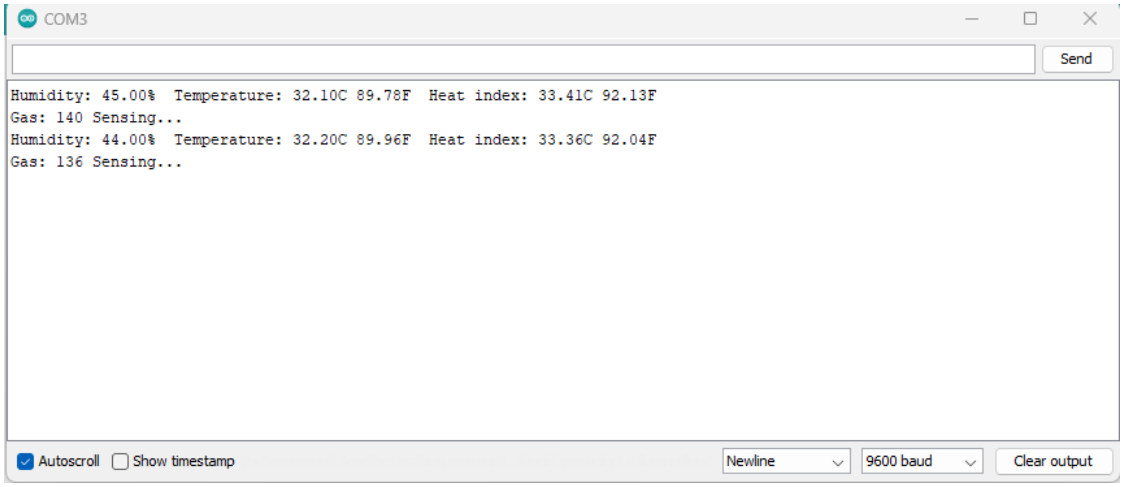
5.6 **Flowchart**



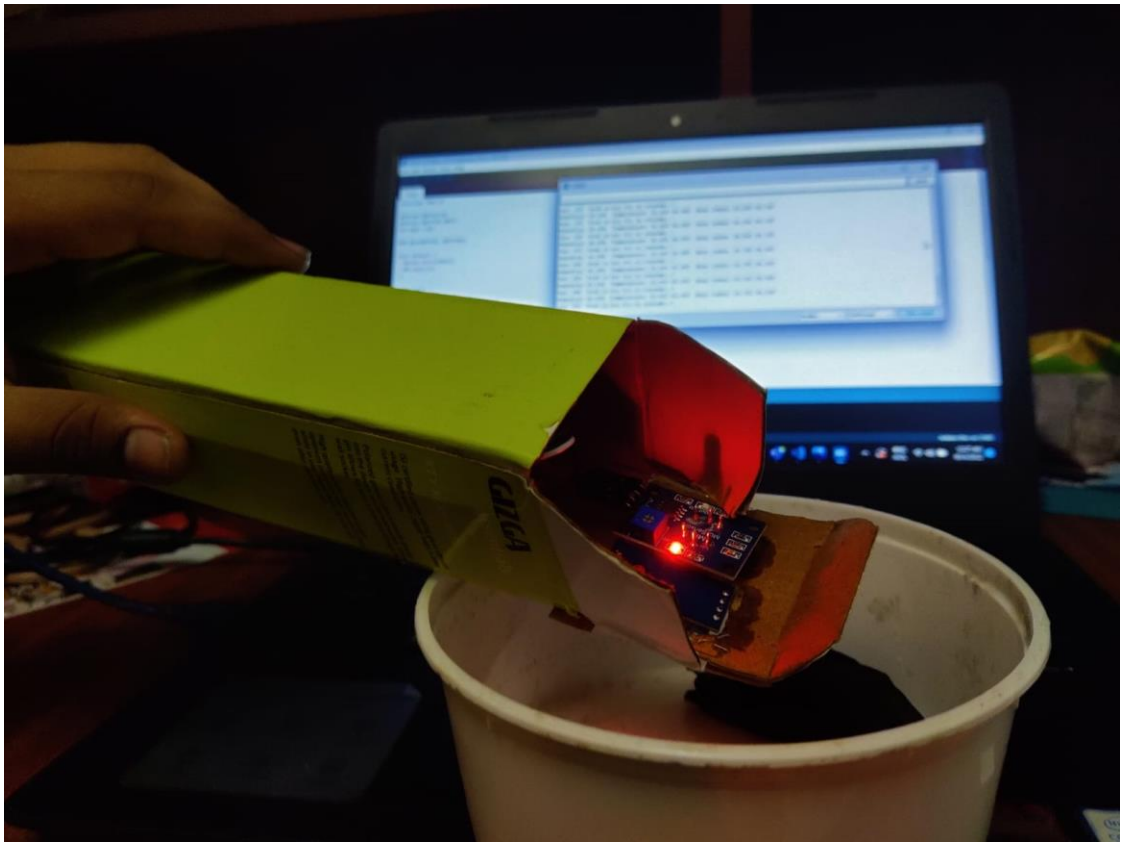
6. Results and Discussion

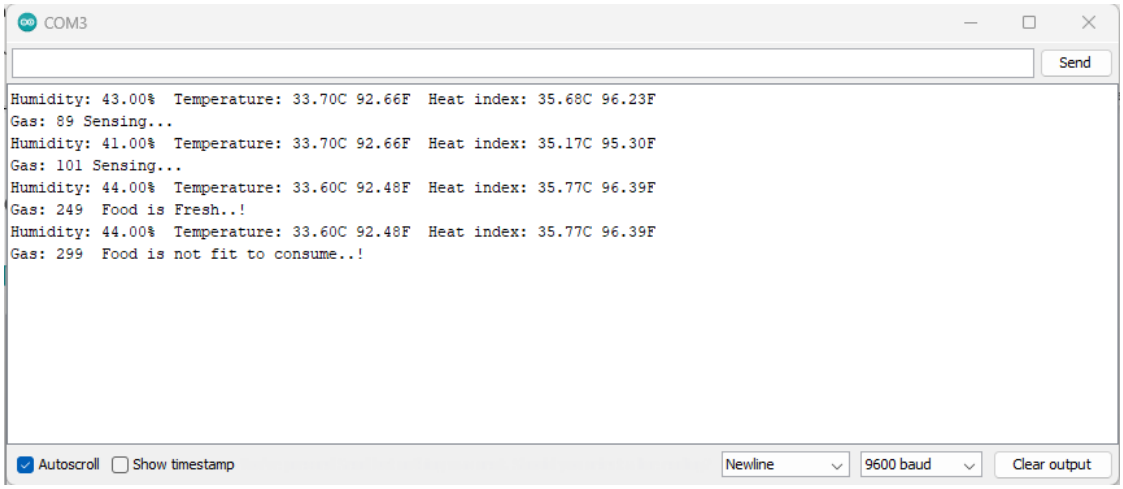
Sensing





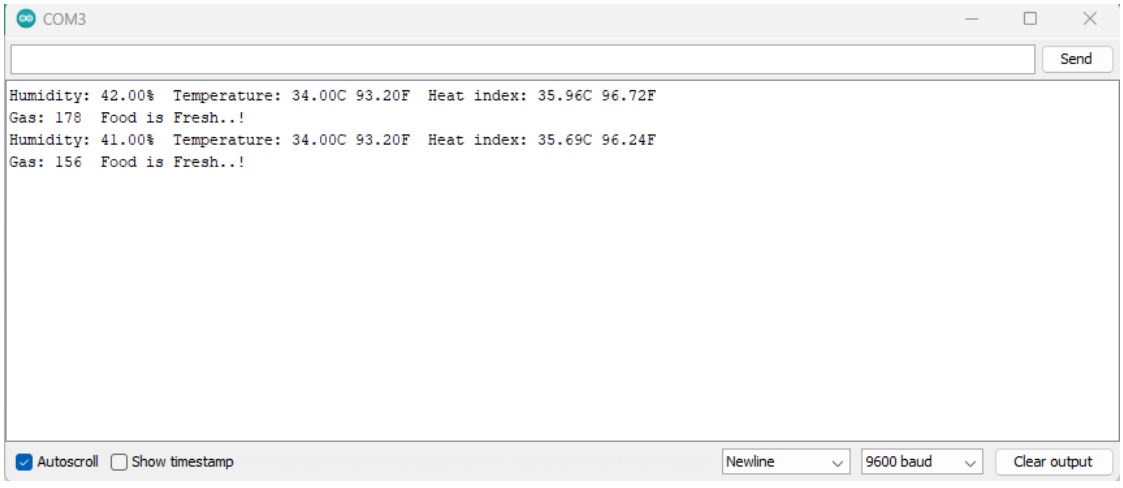
Spoiled Food





Fresh Food


GIGA
Lay's
Classic Salted


COM3
Send
Humidity: 42.00% Temperature: 34.00C 93.20F Heat index: 35.96C 96.72F
Gas: 178 Food is Fresh..!
Humidity: 41.00% Temperature: 34.00C 93.20F Heat index: 35.69C 96.24F
Gas: 156 Food is Fresh..!
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

7. Conclusion and Future Work

Wastage of food is managed by our application which will allow people to give their leftover food at home, restaurants, cafes to the people who are needy through NGO's and the quality of the food would be checked while getting the food from the people so that healthy and hygienic food could be provided to the needy.

This project will help to reduce food wastage and a great help to the people who are needy and starving without food.

8. REFERENCES

[1] https://link.springer.com/article/10.1007/s43039-021-00032-x

[2] https://www.sciencedirect.com/science/article/pii/S09596526183033669

[3] https://www.tandfonline.com/doi/abs/10.1080/0267257X.2020.1863448?journalCode=rjmm20

[4] https://www.sciencedirect.com/science/article/pii/S0038012121000963

[5] https://www.sciencedirect.com/science/article/abs/pii/S0019850118302591

[6] https://jespublication.com/upload/2020-110485.pdf

[7] https://www.sciencedirect.com/science/article/abs/pii/S0278431920302838

[8] https://www.researchgate.net/publication/350197823_Review_on_Efficient_Food_Waste_Management_System_Using_Internet_of_Things

[9] https://ijitis.org/index.php/ijitis/article/view/78/54

[10] https://www.sciencedirect.com/science/article/abs/pii/S0261517718302449

[11] https://link.springer.com/article/10.1007/s12649-016-9720-0

[12] https://www.ijert.org/a-novel-approach-of-online-food-management-system-for-needy-people-during-covid-19

[13] https://www.sciencedirect.com/science/article/abs/pii/S0956053X18304562

[14] https://www.sciencedirect.com/science/article/abs/pii/S0019850121000274

[15] https://www.emerald.com/insight/content/doi/10.1108/BFJ-02-2019-0111/full/html

[16] https://ieeexplore.ieee.org/document/8488190

[17] https://www.mdpi.com/2071-1050/12/4/1595

[18] https://www.emerald.com/insight/content/doi/10.1108/IJCTHR-01-2021-0019/full/html
[19] https://www.semanticscholar.org/paper/Testing-the-effects-of-food-quality%2C-price-and-on-Hanaysha/6734a6b6dd7d6e4a34073131e820dc1ba8cfe278
[20] https://www.irjet.net/archives/V6/i12/IRJET-V6I1296.pdf

APPENDIX

REGISTER PAGE

```
import {
  View,
  Text,
  SafeAreaView,
  TextInput,
  Image,
  ImageBackground,
  StyleSheet,
  Button,
} from "react-native";
import React from "react";
import { useState, useRef } from "react";
import { useAuth } from "../Context/AuthContext";
import ErrorMessage from "../Components/ErrMessage";

function Register({ navigation }) {
  const [email, setEmail] = useState();
  const [password, setPassword] = useState();
  const [Confirmpassword, setConfirmPassword] = useState();
  const { signup } = useAuth();
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);

  const onHandleSignup = async () => {
    if (password !== Confirmpassword) {
      return setError("Passwords do not match");
    }

    try {
      setError("");
      setLoading(true);
      await signup(email, password);
      navigation.navigate("Login");
    } catch {
      setError("Failed to create an account");
    }

    setLoading(false);
  };

  return (
    <SafeAreaView style={styles.container}>
```

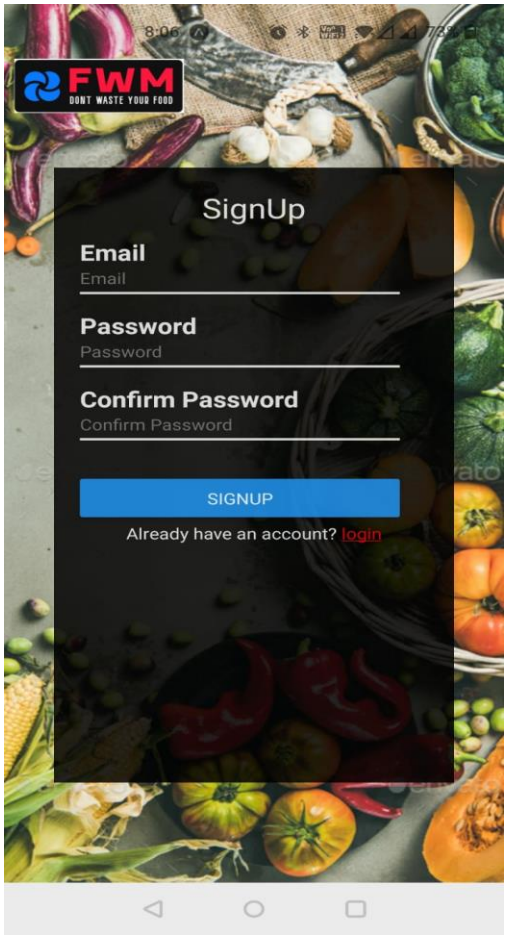
```
      <ImageBackground source={require("./back.png")} style={{ flex: 1
}}>
        <View
          style={{
            top: 50,
            height: 100,
          }}
        >
          <Image
            source={require("./logo.png")}
            style={{ resizeMode: "contain", height: "50%", right: 100 }}
          ></Image>
        </View>
        <View
          style={{
            width: "80%",
            backgroundColor: "black",
            top: 50,
            alignSelf: "center",
            opacity: 0.85,
            height: "70%",
          }}
        >
          <Text
            style={{
              color: "white",
              alignSelf: "center",
              fontSize: 25,
              top: 20,
            }}
          >
            SignUp
          </Text>
          {error ? <ErrorMessage error={error} visible={true} /> : null}
          <Text
            style={{
              color: "white",
              left: 20,
              fontSize: 20,
              fontWeight: "bold",
              top: 30,
            }}
          >
            Email
          </Text>
```

```
        <TextInput
          placeholder="Email"
          placeholderTextColor={"grey"}
          style={{
            width: "80%",
            borderBottomWidth: 2,
            left: 20,
            borderBottomColor: "white",
            top: 25,
            color: "white",
          }}
          value={email}
          onChangeText={(text) => setEmail(text)}
        ></TextInput>
        <Text
          style={{
            color: "white",
            left: 20,
            fontSize: 20,
            fontWeight: "bold",
            top: 40,
          }}
        >
          Password
        </Text>
        <TextInput
          placeholder="Password"
          secureTextEntry={true}
          placeholderTextColor={"grey"}
          style={{
            width: "80%",
            borderBottomWidth: 2,
            left: 20,
            borderBottomColor: "white",
            top: 35,
            color: "white",
          }}
          value={password}
          onChangeText={(text) => setPassword(text)}
        ></TextInput>
        <Text
          style={{
            color: "white",
            left: 20,
            fontSize: 20,
```

```
            fontWeight: "bold",
            top: 50,
          }}
        >
          Confirm Password
        </Text>
        <TextInput
          placeholder="Confirm Password"
          secureTextEntry={true}
          placeholderTextColor={"grey"}
          style={{
            width: "80%",
            borderBottomWidth: 2,
            left: 20,
            borderBottomColor: "white",
            top: 45,
            color: "white",
          }}
          value={Confirmpassword}
          onChangeText={(text) => setConfirmPassword(text)}
        ></TextInput>
        <View
          style={{
            width: "80%",
            left: 20,
            top: 80,
            fontSize: 20,
            fontWeight: "bold",
          }}
        >
          <Button
            title="SignUp"
            onPress={onHandleSignup}
            disabled={loading}
          ></Button>
        </View>
        <Text style={{ color: "white", top: 85, alignSelf: "center"
}}>
          Already have an account?{" "}
          <Text
            style={{ color: "red", textDecorationLine: "underline" }}
            onPress={() => navigation.navigate("Login")}
          >
            login
          </Text>
```

```
          </Text>
        </View>
      </ImageBackground>
    </SafeAreaView>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
  },
});

export default Register;
```

Sample Output



LOGIN PAGE

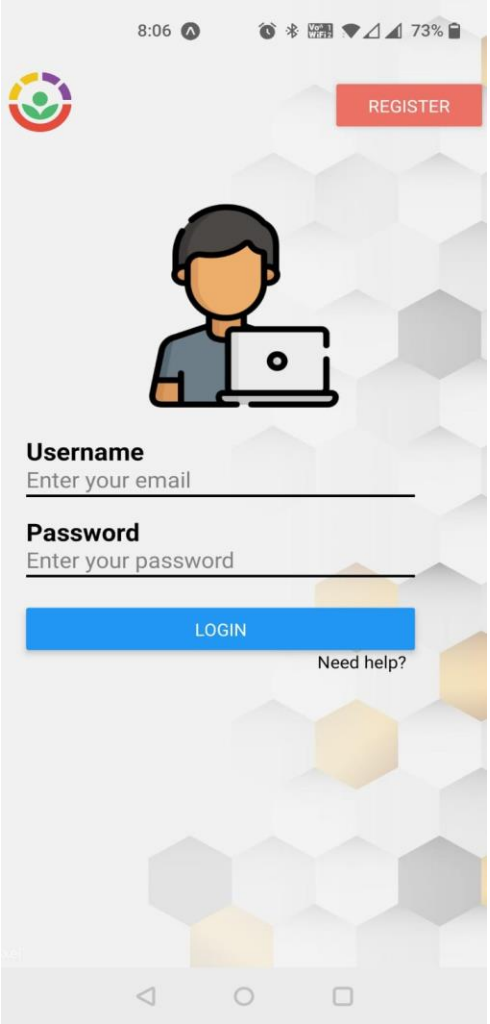
```
import {
  StyleSheet,
  Text,
  View,
  SafeAreaView,
  Button,
  Image,
  TextInput,
  ImageBackground,
} from "react-native";
import { auth } from "../Database/firebase";
import { useState, useRef } from "react";
import { useAuth } from "../Context/AuthContext";
import ErrorMessage from "../Components/ErrMessage";
import React from "react";

function Login({ navigation }) {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const { login } = useAuth();
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);

  function onLogin(e) {
    e.preventDefault();
    try {
      setError("");
      setLoading(true);
      login(email, password);
      if (
        auth.currentUser.metadata.creationTime ===
        auth.currentUser.metadata.lastSignInTime
      ) {
        navigation.navigate("Profile");
      } else {
        navigation.navigate("Home");
      }
    } catch {
      setError("Failed to login");
    }

    setLoading(false);
  }
```

```
  return (
    <SafeAreaView style={styles.container}>
      <ImageBackground
        source={require("../Myassets/white.png")}
        style={{ flex: 1 }}
      >
        <View
          style={{
            flex: 1,
            alignContent: "center",
          }}
        >
          <View
            style={{
              top: 60,
              height: 100,
            }}
          >
            <Image
              source={require("../Myassets/logo.png")}
              style={{ resizeMode: "contain", height: "50%", right: 225
}}
            ></Image>
          </View>
          <View
            style={{
              width: "30%",
              left: 265,
              position: "absolute",
              top: 70,
            }}
          >
            <Button
              title="Register"
              color={"#EC7063"}
              onPress={() => navigation.navigate("Register")}
            ></Button>
          </View>
          {error ? <ErrorMessage error={error} visible={true} /> : null}
          <View style={{ height: "30%", top: 70 }}>
            <Image
              source={require("./user.png")}
              style={{
                alignSelf: "center",
                resizeMode: "contain",
```

```
            height: "70%",
          }}
        ></Image>
      </View>
      <Text
        style={{
          left: 20,
          fontSize: 20,
          fontWeight: "bold",
          top: 20,
          color: "black",
        }}
      >
        Username
      </Text>
      <TextInput
        placeholderTextColor={"grey"}
        placeholder="Enter your email"
        style={{
          width: "80%",
          borderBottomWidth: 2,
          left: 20,
          borderBottomColor: "black",
          top: 15,
          color: "black",
          fontSize: 18,
        }}
        value={email}
        onChangeText={(text) => setEmail(text)}
      />
      <View style={{ top: 10 }}>
        <Text
          style={{
            left: 20,
            fontSize: 20,
            fontWeight: "bold",
            top: 20,
            color: "black",
          }}
        >
          Password
        </Text>
        <TextInput
          secureTextEntry={true}
          placeholder="Enter your password"
```

```
              placeholderTextColor={"grey"}
              style={{
                width: "80%",
                borderBottomWidth: 2,
                left: 20,
                borderBottomColor: "black",
                top: 15,
                color: "black",
                fontSize: 18,
              }}
              value={password}
              onChangeText={(text) => setPassword(text)}
            />
          </View>
          <View
            style={{
              width: "80%",
              left: 20,
              top: 50,
              fontSize: 20,
              fontWeight: "bold",
            }}
          >
            <Button title="Login" onPress={onLogin} disabled={loading}
/>

          </View>
          <Text style={{ left: 250, top: 50, color: "black" }}>Need
help?</Text>
        </View>
      </ImageBackground>
    </SafeAreaView>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
});

export default Login;
```

Sample Output



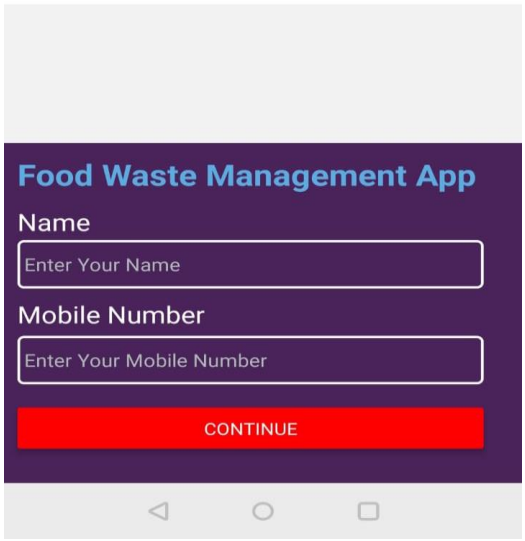
Profile Page

```
import {
  View,
  TextInput,
  Text,
  SafeAreaView,
  Button,
  Image,
} from "react-native";
import React from "react";
import { useState } from "react";
import { database } from "firebase";
import firebase from "firebase/app";
import "firebase/auth";

function Profile({ navigation }) {
  const user = firebase.auth().currentUser;
  const [name, onChangename] = useState("");
  const [ph, onChangephno] = useState("");

  const handleSubmit = () => {
    addprofile(name, ph);
    navigation.navigate("Home");
  };

  let addprofile = (name, ph) => {
    database().ref("/profile").push({
      mail: user.email,
      name: name,
      phno: ph,
    });
  };
  return (
    <SafeAreaView style={{ flex: 1 }}>
      <View>
        <Image
          source={require("../Myassets/donation.png")}
          style={{
            resizeMode: "cover",
            height: "70%",
            width: "100%",
          }}
        ></Image>
      </View>
      <View
```

```
        style={{
          backgroundColor: "#4A235A",
          height: "40%",
          top: -50,
        }}
      >
        <Text
          style={{
            marginLeft: 10,
            fontSize: 25,
            fontWeight: "bold",
            color: "#5DADE2",
            marginTop: 10,
          }}
        >
          Food Waste Management App
        </Text>
        <Text style={{ color: "white", marginLeft: 10, fontSize: 20,
top: 10 }}>
          Name
        </Text>
        <TextInput
          style={{
            width: "90%",
            borderColor: "white",
            borderWidth: 2,
            marginLeft: 10,
            padding: 5,
            top: 12,
            color: "white",
            fontSize: 15,
            borderRadius: 5,
          }}
          placeholder="Enter Your Name"
          placeholderTextColor={"#B2BABB"}
          onChangeText={(text) => onChangename(text)}
        ></TextInput>
        <Text
          style={{
            color: "white",
            marginLeft: 10,
            marginBottom: 10,
            fontSize: 20,
            top: 20,
          }}
```

```
        >
          Mobile Number
        </Text>
        <TextInput
          style={{
            width: "90%",
            borderColor: "white",
            borderWidth: 2,
            marginLeft: 10,
            padding: 5,
            top: 15,
            color: "white",
            fontSize: 15,
            borderRadius: 5,
          }}
          keyboardType="numeric"
          placeholder="Enter Your Mobile Number"
          placeholderTextColor={"#B2BABB"}
          onChangeText={(text) => onChangephno(text)}
        ></TextInput>
        <View
          style={{
            width: "90%",
            top: 30,
            justifyContent: "flex-end",
            marginLeft: 10,
            height: 40,
          }}
        >
          <Button
            color={"red"}
            title="Continue"
            onPress={handleSubmit}
          ></Button>
        </View>
      </View>
    </SafeAreaView>
  );
}
export default Profile;
```

Sample Output



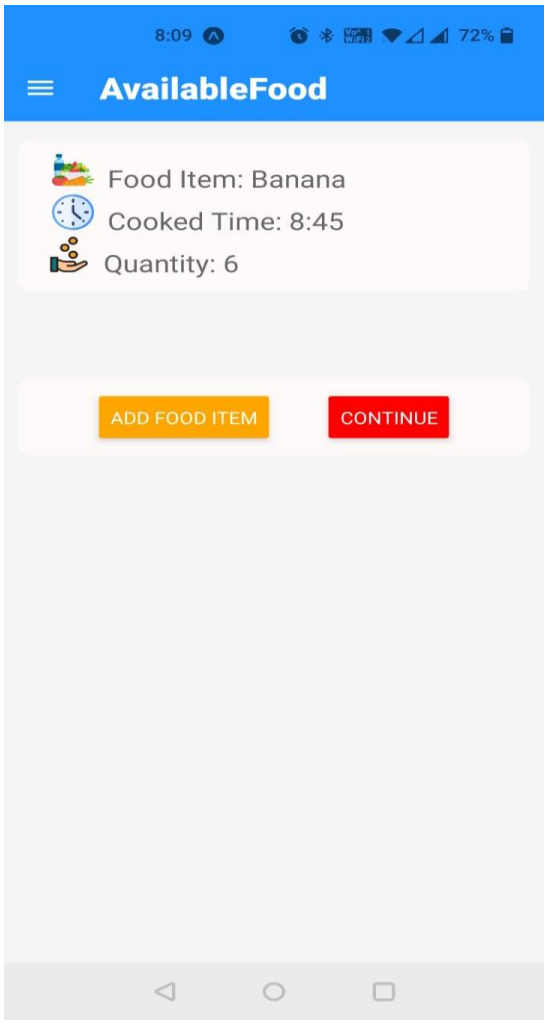
Customer Home Page

```
import { View, StyleSheet, TouchableOpacity, Image } from "react-
native";
import {
  createDrawerNavigator,
  DrawerItem,
  DrawerContentScrollView,
  DrawerItemList,
} from "@react-navigation/drawer";
import { createStackNavigator } from "@react-navigation/stack";
import AvailableFood from "./AvailableFood";
import Address from "./Address";

const Stack = createStackNavigator();
const Drawer = createDrawerNavigator();
import React from "react";
```

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

const NavigationDrawerStructure = (props) => {
  const toggleDrawer = () => {
    props.navigationProps.toggleDrawer();
  };
  return (
    <View style={{ flexDirection: "row" }}>
      <TouchableOpacity onPress={toggleDrawer}>
        <Image
          source={require("./navbar.png")}
          style={{ width: 25, height: 25, marginLeft: 5 }}
        />
      </TouchableOpacity>
    </View>
  );
};

function Home({ navigation }) {
  return (
    <Drawer.Navigator
      screenOptions={{
        activeTintColor: "#e91e63",
        itemStyle: { marginVertical: 5 },
        headerStyle: {
          backgroundColor: "dodgerblue",
        },
        headerTintColor: "white",
        headerTitleStyle: {
          fontWeight: "bold",
          fontSize: 25,
        },
      }}
      drawerContent={(props) => {
        return (
          <DrawerContentScrollView {...props}>
            <DrawerItemList {...props} />
            <DrawerItem
              label="Logout"
              onPress={() => {
                props.navigation.push("Login");
                auth.signOut();
              }}
              icon={() => (
                <Image
```

```
                style={{ height: 30, width: 25, marginLeft: 10 }}
                source={require("../Myassets/logout.png")}
              />
            )}
          />
        </DrawerContentScrollView>
      );
    }}
  >
    <Drawer.Screen
      name="AvailableFood"
      options={{
        drawerLabel: "Available Food",
        drawerIcon: ({ focused }) => (
          <Image
            source={require("./food.png")}
            style={[
              focused ? styles.drawerActive : styles.drawerInActive,
              { height: 30, width: 25, marginLeft: 10 },
            ]}
          />
        ),
      }}
      component={AvailableFood}
    />
  </Drawer.Navigator>
);
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "black",
  },
});
export default Home;
```

Sample Output



Available Food Page

```
import {
  Text,
  View,
  SafeAreaView,
  StyleSheet,
  Button,
  Image,
  Modal,
  TextInput,
  ImageBackground,
  Alert,
  TouchableOpacity,
} from "react-native";
import React from "react";
import firebase from "firebase/app";
```

```
import "firebase/auth";
import { useState } from "react";
import { database } from "firebase";
import DateTimePicker from "@react-native-community/datetimepicker";
import * as ImagePicker from "expo-image-picker";

let addItem = (eml, item, qty, time) => {
  database().ref("/items").push({
    mail: eml,
    name: item,
    quantity: qty,
    c_time: time,
  });
};
let itemsRef = database().ref("/items");

const AvailableFood = ({ navigation }) => {
  const user = firebase.auth().currentUser;
  const [ModalVisible, setModalVisible] = useState(false);
  const [name, onChangename] = useState("");
  const [quantity, onChangequantity] = useState("");
  const [time, setTime] = useState("00:00");
  const [date, setDate] = useState(new Date(1598051730000));
  const [mode, setMode] = useState("date");
  const [show, setShow] = useState(false);
  const [item, setitem] = useState("");
  const [image, setImage] = useState(null);

  const pickImage = async () => {
    let result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.All,
      allowsEditing: true,
      aspect: [4, 3],
      quality: 1,
    });

    console.log(result);

    if (!result.cancelled) {
      setImage(result.uri);
    }
  };

  const onChange = (event, selectedDate) => {
    const currentDate = selectedDate || date;
```

```
    setShow(Platform.OS === "ios");
    setDate(currentDate);
    let tempDate = new Date(currentDate);
    let ctime = tempDate.getHours() + ":" + tempDate.getMinutes();
    setTime(ctime);
  };

  const showMode = (currentMode) => {
    setShow(true);
    setMode(currentMode);
  };

  const showDatepicker = () => {
    showMode("date");
  };

  const showTimepicker = () => {
    showMode("time");
  };

  const handleSubmit = () => {
    addItem(user.email, name, quantity, time);
    Alert.alert("Item saved successfully");
    setTime("00:00");
    setModalVisible(!ModalVisible);
  };

  const [itemsArray, setItemsArray] = React.useState([]);
  React.useEffect(() => {
    itemsRef.on("value", (snapshot) => {
      let data = snapshot.val();
      const items = Object.values(data);
      setItemsArray(items);
    });
  }, []);

  return (
    <SafeAreaView style={styles.container}>
      <View>
        {itemsArray.length > 0 ? (
          <View>
            {itemsArray.map((item, index) => {
              flag = 0;
              if (item.mail == user.email) {
                flag = 1;
```

```
            return (
              <View
                key={index}
                style={{
                  borderRadius: 10,
                  padding: 6,
                  borderColor: "snow",
                  borderWidth: 5,
                  top: 5,
                  marginBottom: 10,
                  backgroundColor: "snow",
                  margin: 10,
                  shadowColor: "#171717",
                  shadowOffset: { width: -2, height: 4 },
                  shadowOpacity: 0.2,
                  shadowRadius: 3,
                }}
              >
                <Text
                  style={{
                    fontSize: 20,
                    paddingLeft: 3,
                    fontFamily: "notoserif",
                    color: "#696969",
                  }}
                >
                  <View>
                    <Image
                      source={require("../Myassets/food.png")}
                      style={{
                        resizeMode: "contain",
                        height: 30,
                        width: 50,
                      }}
                    ></Image>
                  </View>
                  Food Item: {item.name}
                </Text>
                <Text
                  style={{
                    fontSize: 20,
                    paddingLeft: 3,
                    color: "#696969",
                  }}
                >
```

```
                  <View>
                    <Image
                      source={require("../Myassets/time.png")}
                      style={{
                        resizeMode: "contain",
                        height: 30,
                        width: 50,
                      }}
                    ></Image>
                  </View>
                  Cooked Time: {item.c_time}
                </Text>
                <Text style={{ fontSize: 20, color: "#696969" }}>
                  <View>
                    <Image
                      source={require("../Myassets/qty.png")}
                      style={{
                        resizeMode: "contain",
                        height: 30,
                        width: 50,
                      }}
                    ></Image>
                  </View>
                  Quantity: {item.quantity}
                </Text>
                {image && (
                  <Image
                    source={{ uri: image }}
                    style={{ width: 200, height: 200 }}
                  />
                )}
              </View>
            );
          }
        })}
      </View>
    ) : (
      <View>
        <Image
          source={require("./nofood.png")}
          style={{ alignSelf: "center", top: 100, opacity: 0.3 }}
        ></Image>
        <Text style={{ alignSelf: "center", top: "30%" }}>
          No items Added
        </Text>
```

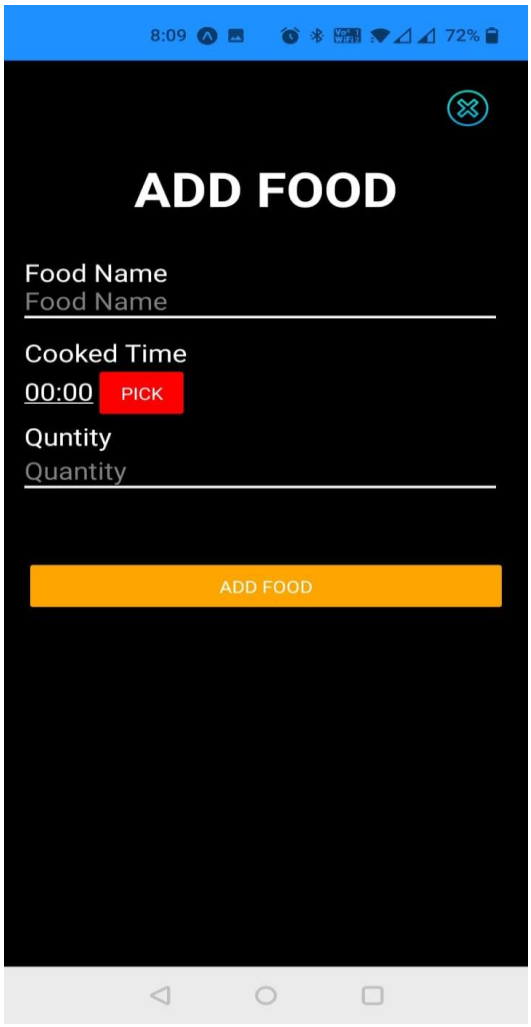
```
          </View>
        )}
        <View
          style={{
            justifyContent: "space-evenly",
            top: "15%",
            flexDirection: "row",
            backgroundColor: "snow",
            borderRadius: 10,
            margin: 10,
            borderWidth: 5,
            borderColor: "snow",
            padding: 10,
          }}
        >
          <Button
            title="Add Food Item"
            color={"orange"}
            onPress={() => setModalVisible(true)}
          ></Button>
          <Button
            title="Continue"
            color={"red"}
            onPress={() => navigation.navigate("Address")}
          />
        </View>
      </View>

      <Modal visible={ModalVisible} transparent={true}
animationType="slide">
        <View
          style={{
            alignSelf: "center",
            top: 5,
            width: "100%",
            height: "100%",
            position: "absolute",
            backgroundColor: "black",
          }}
        >
          <TouchableOpacity
            onPress={() => setModalVisible(!ModalVisible)}
            style={{
              width: 50,
              height: 50,
```

```
            borderWidth: 10,
            borderRadius: 10,
            backgroundColor: "transparent",
            left: "82%",
            borderColor: "transparent",
            top: 10,
          }}
        >
          <Image
            source={require("./close.png")}
            style={{
              width: 40,
              height: 40,
              alignSelf: "center",
              justifyContent: "center",
            }}
          ></Image>
        </TouchableOpacity>
        <Text
          style={{
            fontSize: 40,
            alignSelf: "center",
            color: "white",
            fontFamily: "Roboto",
            fontWeight: "bold",
            top: 30,
          }}
        >
          ADD FOOD
        </Text>
        <Text
          style={{ color: "white", marginLeft: 15, top: 60, fontSize:
20 }}
        >
          Food Name
        </Text>
        <TextInput
          placeholder="Food Name"
          placeholderTextColor={"grey"}
          style={{
            width: "90%",
            borderBottomWidth: 2,
            borderBottomColor: "white",
            top: 55,
            color: "white",
```

```
              marginLeft: 15,
              fontSize: 20,
            }}
            onChangeText={(text) => onChangename(text)}
          ></TextInput>
          <Text
            style={{ color: "white", marginLeft: 15, top: 70, fontSize:
20 }}
          >
            Cooked Time
          </Text>
          <View>
            <Text
              style={{
                color: "white",
                top: 75,
                marginLeft: 15,
                fontSize: 20,
                textDecorationLine: "underline",
              }}
            >
              {time}
            </Text>
            <View style={{ top: 45, width: "16%", marginLeft: 70 }}>
              <Button onPress={showTimepicker} title="pick"
color={"red"} />
            </View>
          </View>
          {show && (
            <DateTimePicker
              testID="dateTimePicker"
              value={date}
              mode={mode}
              is24Hour={true}
              display="default"
              onChange={onChange}
            />
          )}

          <Text
            style={{ color: "white", marginLeft: 15, top: 50, fontSize:
20 }}
          >
            Quntity
          </Text>
```

```
          <TextInput
            keyboardType="numeric"
            placeholderTextColor={"grey"}
            placeholder="Quantity"
            style={{
              width: "90%",
              borderBottomWidth: 2,
              borderBottomColor: "white",
              top: 50,
              color: "white",
              marginLeft: 15,
              fontSize: 20,
            }}
            onChangeText={(text) => onChangequantity(text)}
          ></TextInput>

          <View style={{ top: "15%", width: "90%", alignSelf: "center"
}}>
            <Button title="Add Food" color={"orange"}
onPress={handleSubmit} />
          </View>
        </View>
      </Modal>
    </SafeAreaView>
  );
};
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#F5F5F5",
  },
});
export default AvailableFood;
```

Sample Output



Address Page

```
import {
  SafeAreaView,
  View,
  Text,
  Image,
  TextInput,
  Button,
} from "react-native";
import MapView, { Marker, AnimatedRegion } from "react-native-maps";
import React, { useState } from "react";
import { useEffect } from "react";
import * as Location from "expo-location";
import { database } from "firebase";
import firebase from "firebase/app";
import "firebase/auth";
```

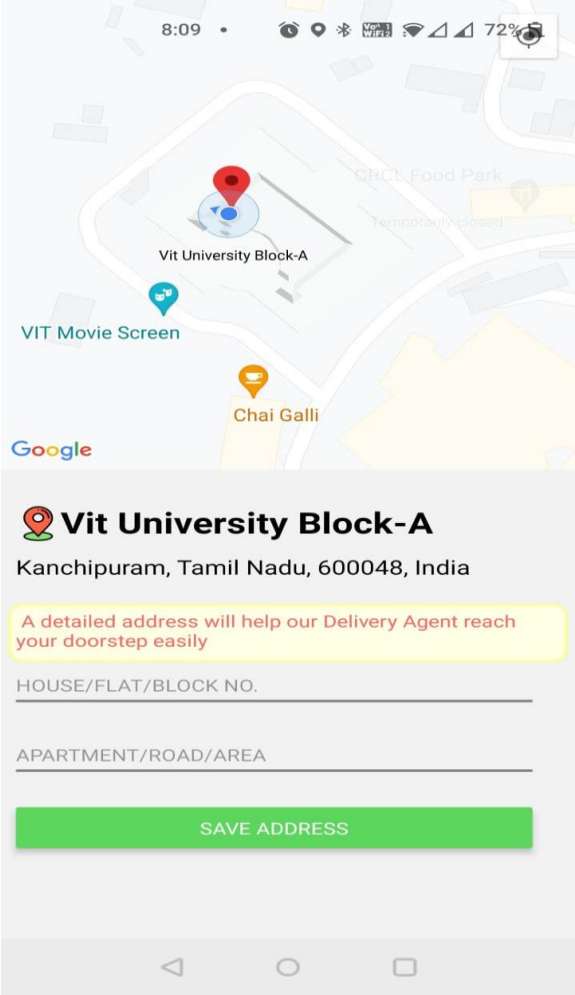
```
let addItem = (eml, lat, lng) => {
  database().ref("/address").push({
    mail: eml,
    latitude: lat,
    longitude: lng,
    stat: 0,
  });
};

function Address({ navigation }) {
  const user = firebase.auth().currentUser;
  const [pin, setPin] = useState({
    latitude: 12.8406,
    longitude: 80.1534,
  });

  const handleSubmit = () => {
    addItem(user.email, pin.latitude, pin.longitude);
    navigation.navigate("Confirm");
  };

  const [locationServiceEnabled, setLocationServiceEnabled] =
useState(false);
  const [displayCurrentAddress, setDisplayCurrentAddress] = useState(
    "Wait, we are fetching you location..."
  );
  const [displayCurrentAddress1, setDisplayCurrentAddress1] =
useState();

  useEffect(() => {
    CheckIfLocationEnabled();
  }, []);

  const CheckIfLocationEnabled = async () => {
    let enabled = await Location.hasServicesEnabledAsync();

    if (!enabled) {
      Alert.alert(
        "Location Service not enabled",
        "Please enable your location services to continue",
        [{ text: "OK" }],
        { cancelable: false }
      );
    } else {
```

```
      setLocationServiceEnabled(enabled);
    }
  };

  useEffect(() => {
    (async () => {
      let { status } = await
Location.requestForegroundPermissionsAsync();
      if (status !== "granted") {
        console.log("Permission to access location was denied");
        return;
      }

      let location = await Location.getCurrentPositionAsync({
        accuracy: Location.Accuracy.Highest,
        maximumAge: 10000,
      });

      setPin({
        latitude: location.coords.latitude,
        longitude: location.coords.longitude,
      });
    })();
  }, []);

  useEffect(() => {
    CheckIfLocationEnabled();
    GetCurrentLocation();
  }, []);

  const GetCurrentLocation = async () => {
    let { coords } = await Location.getCurrentPositionAsync();

    if (coords) {
      const { latitude, longitude } = coords;
      let response = await Location.reverseGeocodeAsync({
        latitude,
        longitude,
      });

      for (let item of response) {
        let address = `${item.name}`;
        let address1 = `${item.subregion}, ${item.region},
${item.postalCode}, ${item.country}`;
        setDisplayCurrentAddress(address);
```

```
        setDisplayCurrentAddress1(address1);
      }
    }
  };

  return (
    <View style={{ backgroundColor: " #f2f2f2", flex: 1 }}>
      <MapView
        style={{ height: "50%" }}
        initialRegion={{
          latitude: pin.latitude,
          longitude: pin.longitude,
          latitudeDelta: 0.002,
          longitudeDelta: 0.0002,
        }}
        showsUserLocation={true}
        followsUserLocation={true}
        zoomEnabled={true}
      >
        <Marker coordinate={pin} draggable={true}></Marker>
      </MapView>
      <Text
        style={{ top: 25, fontSize: 25, fontWeight: "bold", marginLeft:
10 }}
      >
        <Image
          source={require("../Myassets/location.png")}
          style={{ width: 30, height: 30, resizeMode: "contain" }}
        />
        {displayCurrentAddress}
      </Text>
      <Text style={{ top: 35, fontSize: 17, marginLeft: 10 }}>
        {displayCurrentAddress1}
      </Text>
      <Text
        style={{
          top: 50,
          backgroundColor: "#ffffe6",
          padding: 5,
          borderWidth: 2,
          borderRadius: 10,
          margin: 5,
          borderColor: "#ffff99",
          color: "#ff6666",
        }}
```

```
      >
        {" "}
        A detailed address will help our Delivery Agent reach your
doorstep
        easily{" "}
      </Text>
      <TextInput
        style={{
          width: "90%",
          borderBottomWidth: 2,
          borderBottomColor: "grey",
          top: 50,
          marginLeft: 10,
        }}
        placeholder="HOUSE/FLAT/BLOCK NO."
      ></TextInput>
      <TextInput
        style={{
          width: "90%",
          borderBottomWidth: 2,
          borderBottomColor: "grey",
          top: 80,
          marginLeft: 10,
        }}
        placeholder="APARTMENT/ROAD/AREA"
      ></TextInput>
      <View style={{ width: "90%", margin: 10, top: 100 }}>
        <Button
          title="Save Address"
          color={"#5cd65c"}
          onPress={handleSubmit}
        ></Button>
      </View>
    </View>
  );
}
export default Address;
```

Sample Output



Confirm Page

```
import { View, Text, SafeAreaView, StyleSheet, Image } from "react-
native";
import React from "react";
import firebase from "firebase/app";
import { database } from "firebase";
import "firebase/auth";

export default function Confirm() {
  const user = firebase.auth().currentUser;
  let itemsRef = database().ref("/Assigned");
  const [itemsArray, setItemsArray] = React.useState([]);
  React.useEffect(() => {
    itemsRef.on("value", (snapshot) => {
      let data = snapshot.val();
```

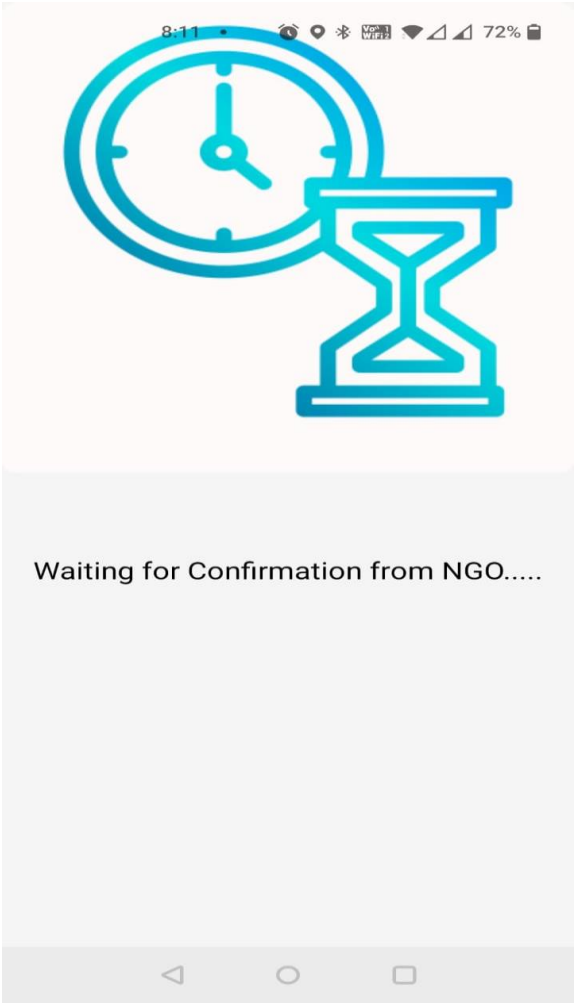
```
      const items = Object.values(data);
      setItemsArray(items);
    });
  }, []);
  let flag = 0;
  function check() {
    if (flag == 0) {
      return (
        <View
          style={{
            borderRadius: 10,
            padding: 10,
            borderColor: "snow",
            borderWidth: 5,

            backgroundColor: "snow",

            shadowColor: "#171717",
            shadowOffset: { width: -2, height: 4 },
            shadowOpacity: 0.2,
            shadowRadius: 3,
            marginBottom: 20,
          }}
        >
          <Image
            source={require("../Myassets/wait.png")}
            style={{ alignSelf: "center" }}
          />
          <Text style={{ alignSelf: "center", top: "30%", fontSize: 20
}}>
            Waiting for Confirmation from NGO.....
          </Text>
        </View>
      );
    }
  }
  return (
    <SafeAreaView style={styles.container}>
      <View>
        {itemsArray.length > 0 ? (
          <View>
            {itemsArray.map((item, index) => {
              if (item.user_mail == user.email) {
                flag = 1;
                return (
```

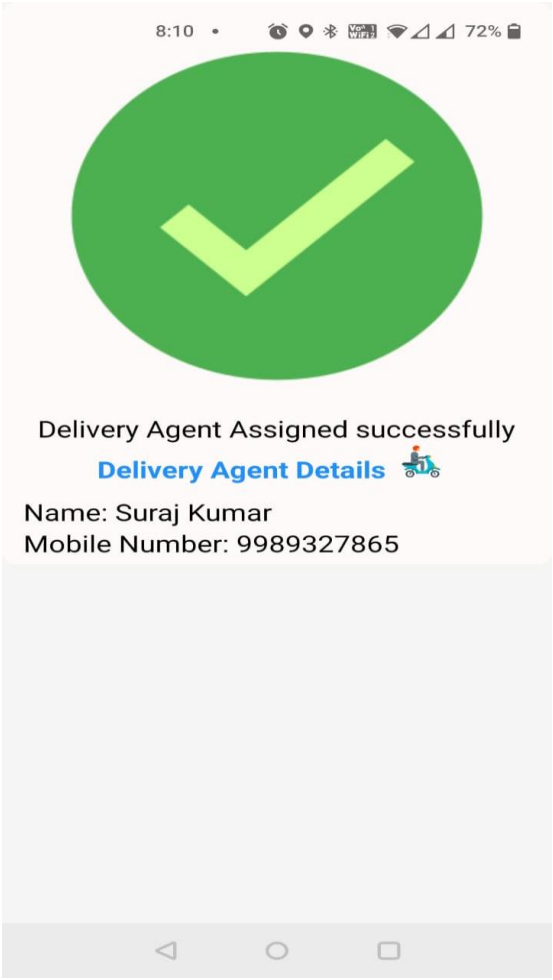
```
                    <View
                      key={index}
                      style={{
                        borderRadius: 10,
                        padding: 10,
                        borderColor: "snow",
                        borderWidth: 5,

                        backgroundColor: "snow",

                        shadowColor: "#171717",
                        shadowOffset: { width: -2, height: 4 },
                        shadowOpacity: 0.2,
                        shadowRadius: 3,
                        marginBottom: 20,
                      }}
                    >
                      <View>
                        <Image
                          source={require("../Myassets/ok.png")}
                          style={{
                            alignSelf: "center",
                          }}
                        ></Image>
                        <Text style={{ fontSize: 20, alignSelf: "center"
}}>

                          Delivery Agent Assigned successfully
                        </Text>
                      </View>
                      <Text
                        style={{
                          fontSize: 20,
                          fontWeight: "bold",
                          alignSelf: "center",
                          color: "#1E90FF",
                        }}
                      >
                        Delivery Agent Details
                        <View>
                          <Image
                            source={require("../Myassets/deliveryman.png")
}

                            style={{
                              resizeMode: "contain",
                              height: 30,
```

```
                        width: 50,
                      }}
                    ></Image>
                  </View>
                </Text>
                <Text
                  style={{
                    fontSize: 20,
                    top: 10,
                  }}
                >
                  Name: {item.deliveryagent_name}
                </Text>
                <Text style={{ fontSize: 20, top: 10 }}>
                  Mobile Number: {item.deliveryagent_phno}
                </Text>
              </View>
            );
          }
        })}
        {check()}
      </View>
    ) : (
      <View
        style={{
          borderRadius: 10,
          padding: 10,
          borderColor: "snow",
          borderWidth: 5,

          backgroundColor: "snow",

          shadowColor: "#171717",
          shadowOffset: { width: -2, height: 4 },
          shadowOpacity: 0.2,
          shadowRadius: 3,
          marginBottom: 20,
        }}
      >
        <Text style={{ alignSelf: "center", top: "30%" }}>
          Waiting for Confirmation from NGO.....
        </Text>
      </View>
    )}
  </View>
```

```
    </SafeAreaView>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "#F5F5F5",
  },
});
```

Sample Output





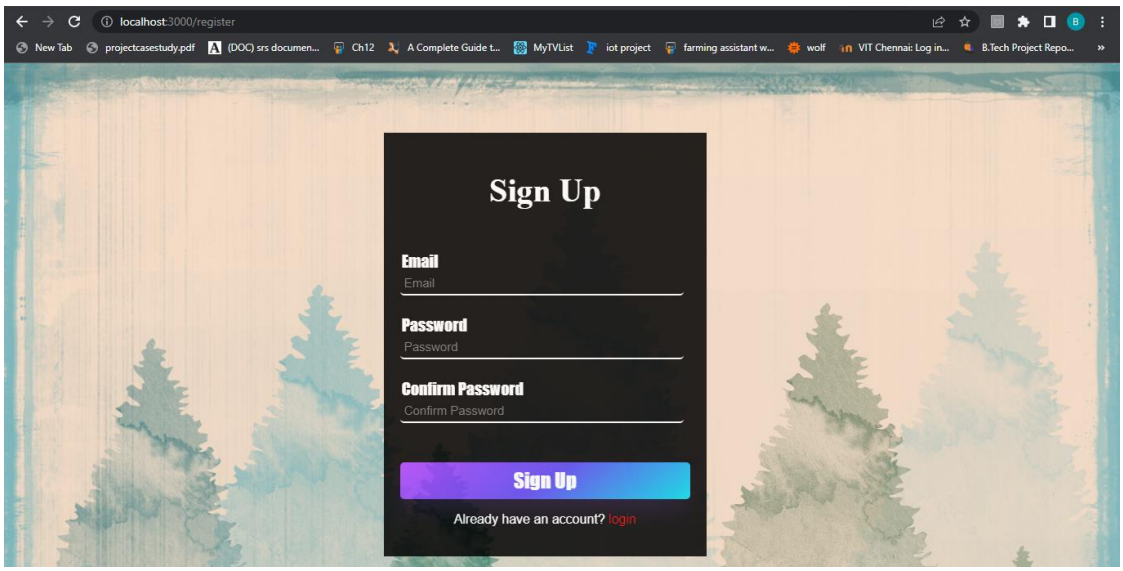
NGO Admin Module

Signup Page

```
import { useState, useRef } from "react";
import { useAuth } from "../context/AuthContext";
import { Link, useNavigate } from "react-router-dom";
function Register() {
  const emailRef = useRef();
  const passwordRef = useRef();
  const passwordConfirmRef = useRef();
  const { signup } = useAuth();
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();

  async function handleSubmit(e) {
    e.preventDefault();
```

```
    if (passwordRef.current.value !== passwordConfirmRef.current.value)
{
      return setError("Passwords do not match");
    }

    try {
      setError("");
      setLoading(true);
      await signup(emailRef.current.value, passwordRef.current.value);
      navigate("/");
    } catch {
      setError("Failed to create an account");
    }

    setLoading(false);
  }
  return (
    <div className="register_div">
      <center>
        <h1 style={{ fontFamily: "lucida" }}>Sign Up</h1>
        <center>{error && <div className="err-
msg">{error}</div>}</center>
      </center>
      <br />
      <form onSubmit={handleSubmit}>
        <legend>Email</legend>
        <input
          type="text"
          placeholder="Email"
          className="email"
          name="email"
          ref={emailRef}
        ></input>
        <br />
        <br />
        <legend>Password</legend>
        <input
          type="password"
          placeholder="Password"
          className="psw"
          name="password"
          ref={passwordRef}
        ></input>
        <br />
        <br />
```

```
        <legend>Confirm Password</legend>
        <input
          type="password"
          placeholder="Confirm Password"
          className="cpsw"
          ref={passwordConfirmRef}
        ></input>
        <br />
        <br />
        <br />
        <button type="submit" className="sub" disabled={loading}>
          Sign Up
        </button>
        <center>
          <p className="member">
            Already have an account?{" "}
            <Link to="/" style={{ color: "red", textDecoration: "none"
}}>
              login
            </Link>
          </p>
        </center>
      </form>
    </div>
  );
}
export default Register;
```

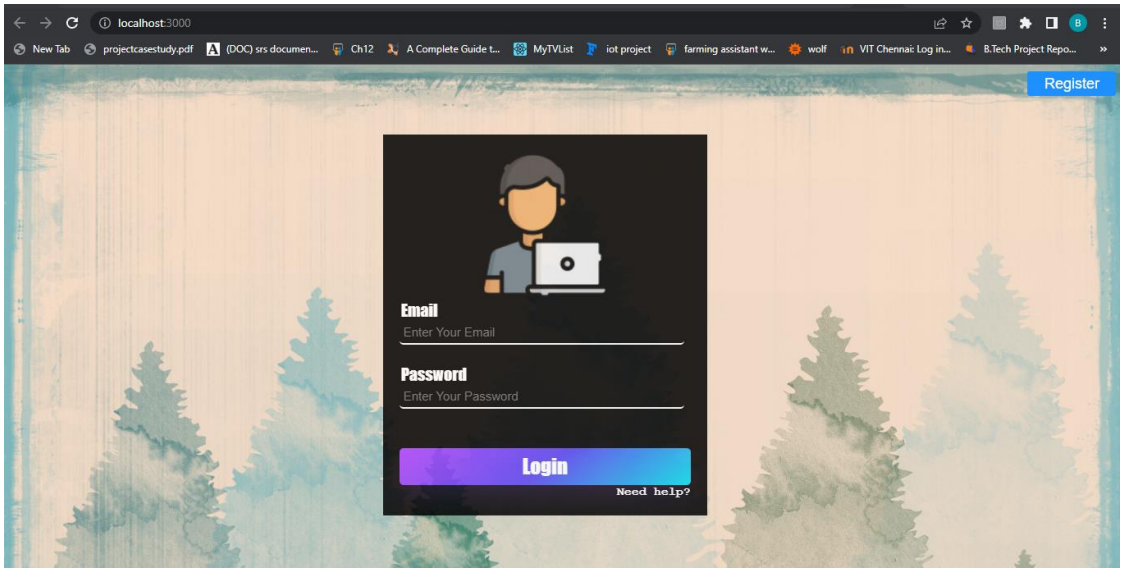


Login Page

```
import user_icon from "./user.png";
import { Link, useNavigate } from "react-router-dom";
import { useState, useRef } from "react";
import { useAuth } from "../context/AuthContext";

function Login() {
  const emailRef = useRef();
  const passwordRef = useRef();
  const { login } = useAuth();
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();

  async function handleSubmit(e) {
    e.preventDefault();

    try {
      setError("");
      setLoading(true);
      await login(emailRef.current.value, passwordRef.current.value);
      navigate("/home");
    } catch {
      setError("Failed to login");
    }

    setLoading(false);
  }
  function anchorMouseOverHandeler(e) {
    e.target.style.textDecoration = "underline";
  }

  function anchorMouseOutHandeler(e) {
    e.target.style.textDecoration = "none";
  }
  function capture(e) {
    if (e.target.value === "") {
      e.target.style.borderColor = "red";
    } else {
      e.target.style.borderColor = "white";
    }
  }
  const newLocal = (
    <form onSubmit={handleSubmit}>
      <button
```

```
        className="register"
        onMouseOver={anchorMouseOverHandeler}
        onMouseOut={anchorMouseOutHandeler}
      >
        <Link to="/register" style={{ textDecoration: "none", color:
"white" }}>
          Register
        </Link>
      </button>
      <div className="login_div">
        <center>{error && <div className="err-
msg">{error}</div>}</center>
        <center>
          <img
            src={user_icon}
            alt="user"
            style={{ heigh: "50%", width: "50%" }}
          ></img>
        </center>
        <legend style={{ float: "left" }}>Email</legend>
        <br></br>
        <input
          type="text"
          className="email"
          placeholder="Enter Your Email"
          onBlur={capture}
          ref={emailRef}
        ></input>
        <br></br>
        <br></br>
        <legend>Password</legend>
        <input
          type="password"
          className="psw"
          placeholder="Enter Your Password"
          onBlur={capture}
          ref={passwordRef}
        ></input>
        <br></br>
        <br></br>
        <br></br>

        <button type="submit" className="login" disabled={loading}>
          Login
        </button>
```

```
        <a
          className="fpsw"
          href="/register"
          onMouseOver={anchorMouseOverHandeler}
          onMouseOut={anchorMouseOutHandeler}
        >
          Need help?
        </a>
      </div>
    </form>
  );
  return newLocal;
}
export default Login;
```



NGO Module Home Page

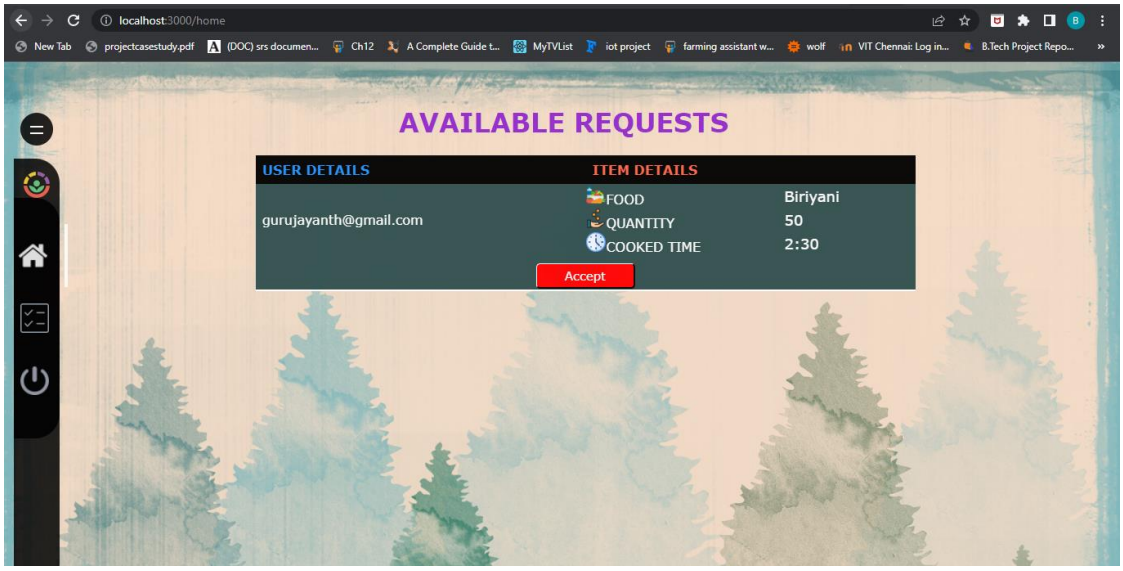
```
import React from "react";
import { useAuth } from "../context/AuthContext";
import { Alert } from "react-bootstrap";
import Sidebar from "./Sidenav";
import MotionHoc from "./MotionHoc";
import { database } from "firebase";
import food from "./food.png";
import qty from "./qty.png";
import time from "./time.png";
import { Link, useNavigate } from "react-router-dom";
import { useState, useRef } from "react";

function HomeComponent() {
  const [error, setError] = useState("");
  const [userMail, setUserMail] = useState("");
  const { currentUser, logout } = useAuth();
  const navigate = useNavigate();
  const [loading, setLoading] = useState(false);
  const itemsRef = database().ref("/items");
  const assignedRef = database().ref("/Assigned");
  const [itemsArray, setItemsArray] = React.useState([]);
  const [assignedArray, setassignedArray] = React.useState([]);

  React.useEffect(() => {
    itemsRef.on("value", (snapshot) => {
      let data = snapshot.val();
      const items = Object.values(data);
      setItemsArray(items);
    });
  }, []);

  React.useEffect(() => {
    assignedRef.on("value", (snapshot) => {
      let data1 = snapshot.val();
      const items1 = Object.values(data1);
      setassignedArray(items1);
    });
  }, []);

  function handleSubmit(mail) {
    try {
      setError("");
      setLoading(true);
```

```
      navigate("/assign", { state: { id: 1, email: mail } });
    } catch {
      setError("Failed to Accept");
    }

    setLoading(false);
  }
  async function handleLogout() {
    setError("");

    try {
      await logout();
      navigate("/");
    } catch {
      setError("Failed to log out");
    }
  }
  return (
    <div>
      {error && <Alert variant="danger">{error}</Alert>}
      <Sidebar />
      <div>
        <div style={{ marginTop: "4%" }}>
          <center>
            <h1 style={{ color: "DarkOrchid", fontFamily: "verdana" }}>
              AVAILABLE REQUESTS
            </h1>
          </center>

          {itemsArray.map((item, index) => {
            {
              const list = [];
              assignedArray.map((agent) => {
                list.push(agent.user_mail);
              });
              {
                var flag = 0;
                for (var i = 0; i < list.length; i++) {
                  if (item.mail === list[i]) {
                    flag = 1;
                  }
                }
                if (flag === 0) {
                  return (
                    <table
```

```
                key={index}
                style={{
                  width: "60%",
                  marginLeft: "22%",
                  borderBottom: "2px solid white",
                  backgroundColor: "DarkSlateGrey",
                  color: "white",
                  opacity: 0.95,
                  borderCollapse: "collapse",
                  fontFamily: "verdana",
                }}
              >
                <tr style={{ backgroundColor: "black" }}>
                  <th
                    style={{
                      color: "dodgerblue",
                      textAlign: "left",
                      padding: "8px",
                      fontFamily: "verdana",
                    }}
                  >
                    USER DETAILS
                  </th>
                  <th
                    style={{
                      color: "tomato",
                      textAlign: "left",
                      padding: "8px",
                      fontFamily: "verdana",
                    }}
                    colSpan="2"
                  >
                    ITEM DETAILS
                  </th>
                </tr>
                <tr>
                  <td
                    rowSpan="3"
                    style={{
                      textAlign: "left",
                      padding: "8px",
                      width: "50%",
                    }}
                  >
                    {item.mail}
```

```
                        </td>
                        <td style={{ width: "30%" }}>
                          <img
                            src={food}
                            style={{
                              width: "10%",
                            }}
                          ></img>
                          FOOD
                        </td>
                        <td style={{ fontSize: 18 }}>{item.name}</td>
                      </tr>
                      <tr>
                        <td style={{ width: "30%" }}>
                          <img src={qty} style={{ width: "10%" }}></img>
                          QUANTITY
                        </td>
                        <td style={{ fontSize: 18
}}>{item.quantity}</td>
                      </tr>
                      <tr>
                        <td style={{ width: "30%" }}>
                          <img src={time} style={{ width: "10%"
}}></img>

                          COOKED TIME
                        </td>

                        <td style={{ fontSize: 18 }}>{item.c_time}</td>
                      </tr>
                      <tr>
                        <td
                          colSpan="3"
                          style={{ paddingTop: "10px", textAlign:
"center" }}
                        >
                          <button
                            style={{
                              width: "15%",
                              color: "white",
                              backgroundColor: "red",
                              fontSize: 15,
                              padding: 5,
                              fontFamily: "inherit",
                              borderRadius: "5px",
                              borderRight: "2px solid black",
```

```
                        borderBottom: "2px solid black",
                        borderLeft: "1px solid white",
                        borderTop: "1px solid white",
                        cursor: "pointer",
                      }}
                      onClick={() => {
                        handleSubmit(item.mail);
                      }}
                    >
                      Accept
                    </button>
                  </td>
                </tr>
              </table>
            );
          }
        }
      }
    })}
  </div>
</div>
</div>
);
}
const Home = MotionHoc(HomeComponent);
export default Home;
```



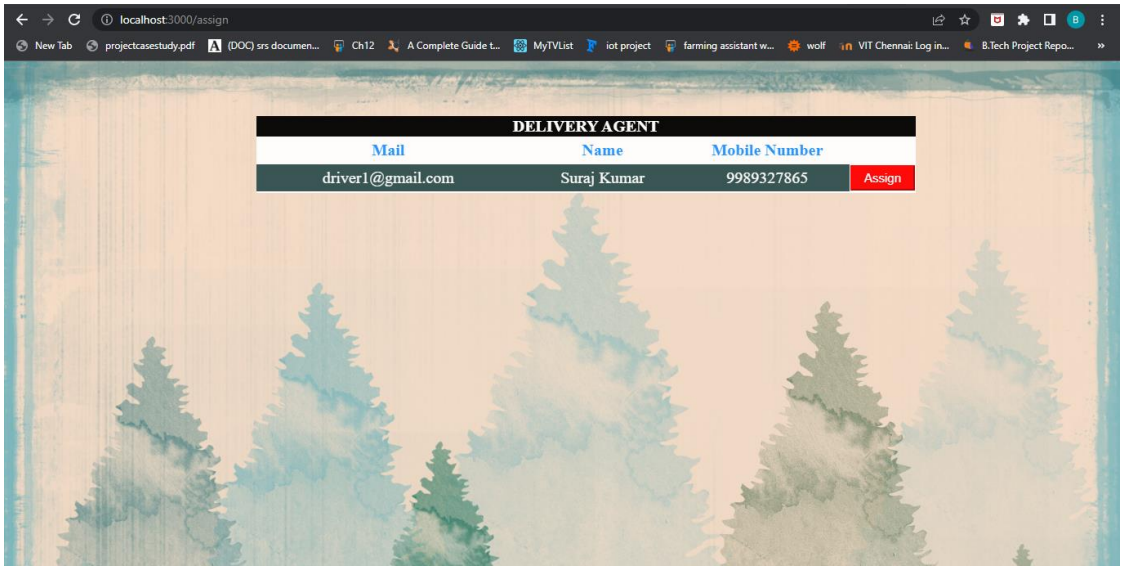
Assign Page

```
import { database } from "firebase";
import React from "react";
import firebase from "firebase";
import { useNavigate, useLocation } from "react-router-dom";

function Assign() {
  const itemsRef = database().ref("/DA_profile");
  const assignedRef = database().ref("/Assigned");
  const user = firebase.auth().currentUser;
  const [itemsArray, setItemsArray] = React.useState([]);
  const [assignedArray, setassignedArray] = React.useState([]);

  const location = useLocation();
  const navigate = useNavigate();

  React.useEffect(() => {
    itemsRef.on("value", (snapshot) => {
      let data = snapshot.val();
      const items = Object.values(data);
      setItemsArray(items);
    });
  }, []);

  React.useEffect(() => {
    assignedRef.on("value", (snapshot) => {
      let data1 = snapshot.val();
      const items1 = Object.values(data1);
      setassignedArray(items1);
    });
  }, []);

  function handleSubmit(email, name, phno) {
    database().ref("Assigned").push({
      user_mail: location.state.email,
      deliveryagent_mail: email,
      deliveryagent_name: name,
      deliveryagent_phno: phno,
    });
    alert("Delivery Agent Assigned Successful");
    navigate("/home");
  }
  return (
    <div>
      {itemsArray.length > 0 ? (
```

```
        <div style={{ marginTop: "5%" }}>
          {itemsArray.map((item, index) => {
            {
              const list = [];
              assignedArray.map((agent) => {
                list.push(agent.deliveryagent_mail);
              });
              var flag = 0;
              for (var i = 0; i < list.length; i++) {
                if (item.email === list[i]) {
                  flag = 1;
                }
              }
              if (flag === 0) {
                return (
                  <table
                    key={index}
                    style={{
                      borderColor: "#2ECC71",
                      borderWidth: 5,
                      width: "60%",
                      marginLeft: "22%",
                      borderBottom: "2px solid white",
                      backgroundColor: "DarkSlateGrey",
                      color: "white",
                      fontFamily: "revert",
                      opacity: 0.95,
                      borderCollapse: "collapse",
                      fontSize: 20,
                    }}
                  >
                    <tr style={{ backgroundColor: "black" }}>
                      <th colSpan="4">DELIVERY AGENT</th>
                    </tr>
                    <tr
                      style={{
                        color: "dodgerblue",
                        backgroundColor: "white",
                        textAlign: "center",
                      }}
                    >
                      <th style={{ width: "40%" }}>Mail</th>
                      <th style={{ width: "25%" }}>Name</th>
                      <th style={{ width: "25%" }}>Mobile Number</th>
                      <th
```

```
                        rowSpan="2"
                        style={{
                          width: "10%",
                          textAlign: "end",
                          paddingTop: "20px",
                        }}
                      >
                        <button
                          style={{
                            color: "white",
                            backgroundColor: "red",
                            width: "100%",
                            padding: "5px",
                            borderColor: "red",
                            fontSize: 15,
                            marginTop: "15px",
                            cursor: "pointer",
                          }}
                          onClick={() => {
                            handleSubmit(item.email, item.name,
item.phno);

                          }}
                        >
                          Assign
                        </button>
                      </th>
                    </tr>
                    <tr>
                      <td style={{ color: "white", textAlign: "center"
}}>

                        {item.email}
                      </td>
                      <td style={{ color: "white", textAlign: "center"
}}>

                        {item.name}
                      </td>
                      <td style={{ color: "white", textAlign: "center"
}}>

                        {item.phno}
                      </td>
                    </tr>
                  </table>
                );
              }
            }
```

```
          })}
        </div>
      ) : (
        <div>
          <h3 style={{ alignSelf: "center", top: "30%" }}>
            NO DELIVERY AGENT AVAILABLE
          </h3>
        </div>
      )}
    </div>
  );
}
export default Assign;
```



Delivery Module

Home Page

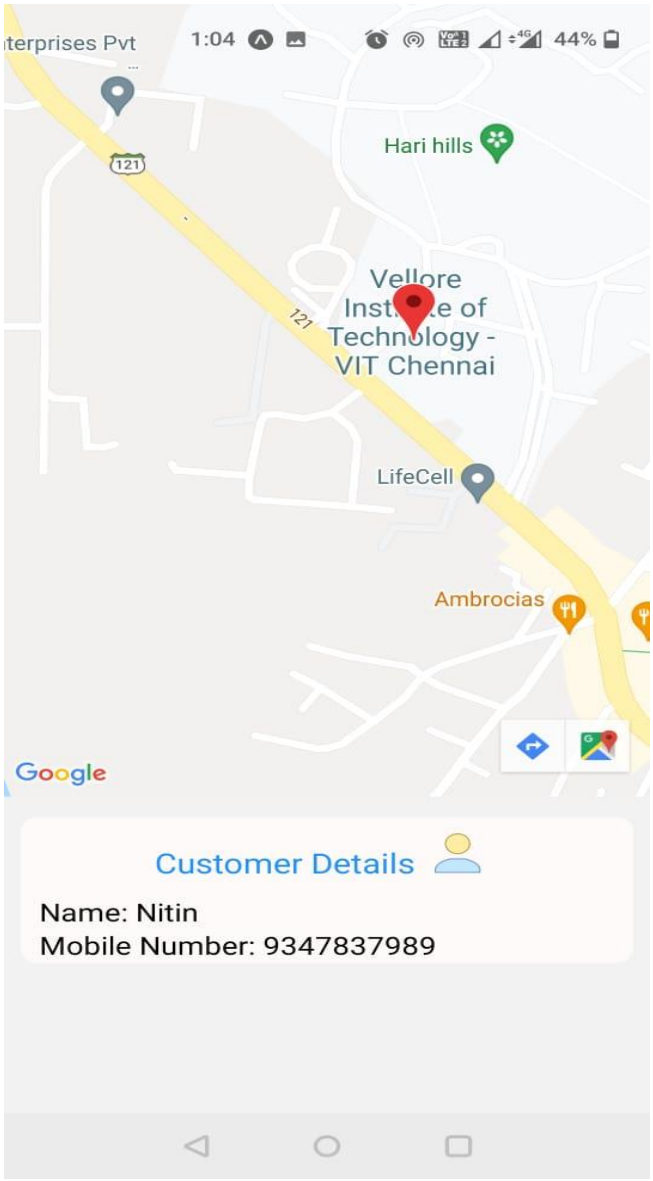
```
import { View, SafeAreaView, Text, Image, StyleSheet } from "react-
native";
import firebase from "firebase/app";
import "firebase/auth";
import { database } from "firebase";
import React from "react";
import { useState } from "react";
import MapView, { Marker } from "react-native-maps";

let itemsRef = database().ref("/address");
let userprofile = database().ref("/profile");
let assignRef = database().ref("/Assigned");

function Home() {
  const user = firebase.auth().currentUser;

  const [itemsArray, setItemsArray] = React.useState([]);
  React.useEffect(() => {
    itemsRef.on("value", (snapshot) => {
      let data = snapshot.val();
      const items = Object.values(data);
      setItemsArray(items);
    });
  }, []);

  const [assignArray, setassignArray] = React.useState([]);
  React.useEffect(() => {
    assignRef.on("value", (snapshot) => {
      let data = snapshot.val();
      const items = Object.values(data);
      setassignArray(items);
    });
  }, []);

  const [pin, setPin] = useState({
    latitude: 12.8406,
    longitude: 80.1534,
  });

  const [userArray, setuserArray] = React.useState([]);
  React.useEffect(() => {
    userprofile.on("value", (snapshot) => {
```

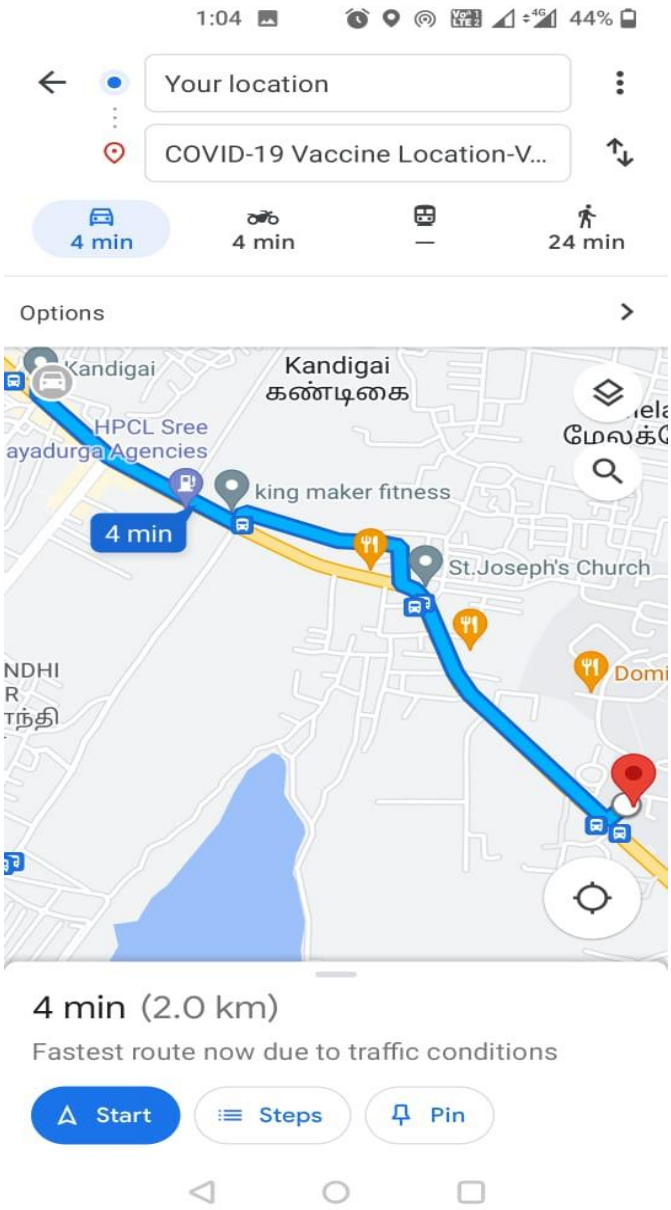
```
      let data = snapshot.val();
      const items = Object.values(data);
      setuserArray(items);
    });
  }, []);

  React.useEffect(() => {
    CheckIfLocationEnabled();
  }, []);

  const CheckIfLocationEnabled = async () => {
    let enabled = await Location.hasServicesEnabledAsync();

    if (!enabled) {
      Alert.alert(
        "Location Service not enabled",
        "Please enable your location services to continue",
        [{ text: "OK" }],
        { cancelable: false }
      );
    } else {
      setLocationServiceEnabled(enabled);
    }
  };

  React.useEffect(() => {
    (async () => {
      let { status } = await
Location.requestForegroundPermissionsAsync();
      if (status !== "granted") {
        console.log("Permission to access location was denied");
        return;
      }

      let location = await Location.getCurrentPositionAsync({
        accuracy: Location.Accuracy.Highest,
        maximumAge: 10000,
      });

      setPin({
        latitude: location.coords.latitude,
        longitude: location.coords.longitude,
      });
    })();
  }, []);
```

```
  return (
    <View>
      {itemsArray.length > 0 ? (
        <View>
          {itemsArray.map((item, index) => {
            const list = [];
            const list1 = [];

            assignArray.map((agent) => {
              list.push(agent.deliveryagent_mail);
              list1.push(agent.user_mail);
            });

            for (var i = 0; i < list.length; i++) {
              if (item.mail === list1[i] && user.email === list[i]) {
                return (
                  <View key={index}>
                    <MapView
                      style={{ height: "80%" }}
                      initialRegion={{
                        latitude: item.latitude,
                        longitude: item.longitude,
                        latitudeDelta: 0.002,
                        longitudeDelta: 0.0002,
                      }}
                      sshowsUserLocation={true}
                      followsUserLocation={true}
                      zoomEnabled={true}
                    >
                      <Marker coordinate={pin}
draggable={true}></Marker>
                    </MapView>
                    {userArray.map((us, index) => {
                      if (us.mail === list1[i]) {
                        return (
                          <View
                            key={index}
                            style={{
                              borderRadius: 10,
                              padding: 6,
                              borderColor: "snow",
                              borderWidth: 5,
                              top: 5,
                              marginBottom: 10,
```

```
                            backgroundColor: "snow",
                            margin: 10,
                            shadowColor: "#171717",
                            shadowOffset: { width: -2, height: 4 },
                            shadowOpacity: 0.2,
                            shadowRadius: 3,
                          }}
                        >
                          <Text
                            style={{
                              fontSize: 20,
                              color: "dodgerblue",
                              alignSelf: "center",
                            }}
                          >
                            Customer Details
                            <View>
                              <Image
                                source={require("../Myassets/user.png"
)}

                                style={{
                                  resizeMode: "contain",
                                  height: 30,
                                  width: 50,
                                }}
                              ></Image>
                            </View>
                          </Text>

                          <Text style={{ fontSize: 18, top: 10 }}>
                            Name: {us.name}
                          </Text>
                          <Text style={{ fontSize: 18, top: 10 }}>
                            Mobile Number: {us.phno}
                          </Text>
                        </View>
                      );
                    }
                  })}
                </View>
              );
            }
          }
        })}
      </View>
```

```
      ) : (
        <View>
          <Text>No Orders Available</Text>
        </View>
      )}
    </View>
  );
}
export default Home;
```




1:04
44%
Your location
COVID-19 Vaccine Location-V...
4 min
4 min
24 min
Options
Kandigai
கண்டிகை
HPCL Sree
ayadurga Agencies
king maker fitness
4 min
St.Joseph's Church
4 min (2.0 km)
Fastest route now due to traffic conditions
Start
Steps
Pin