

**MAKE IT NATURAL:APPLICATION FOR  
MANAGING FOOD WASTE AND EXCESS FOOD AND  
PLANTS IN HOMES: A STEP TO MINIMIZE FOOD  
WASTAGE**

**A PROJECT REPORT**

*Submitted by*

**B.C.DHEVESH(211420104065)  
M.P.DINESH KUMAR(211420104068)  
T.LOKESH BHARATHWAJ(211420104149)**

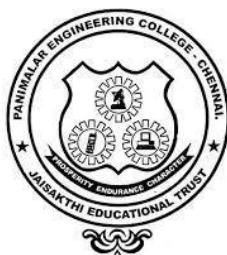
*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**APRIL 2024**

## A PROJECT REPORT

*Submitted by*

**B.C.DHEVESH [211420104065]**

**M.P.DINESH KUMAR [211420104068]**

**T.LOKESH BHRATHWAJ [211420104149]**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**



**PANIMALAR ENGINEERING COLLEGE**

(An Autonomous Institution, Affiliated to Anna University, Chennai)

**APRIL 2024**

**PANIMALAR ENGINEERING COLLEGE**  
**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report "**MAKE IT NATURAL:APPLICATION FOR MANAGING FOOD WASTE AND EXCESS FOOD AND PLANTS IN HOMES: A STEP TO MINIMIZE FOOD WASTAGE**" is the bonafide work of "**B.C.DHEVESH, M.P.DINESH KUMAR, T.LOKESH BHARATHWAJ**" who carried out the project work under my supervision.

**Signature of the HOD with date**

**DR L. JABASHEELA M.E.,Ph.D.,  
Professor and Head,**

Department of Computer Science and  
Engineering,

Panimalar Engineering College,  
Chennai - 123

**Signature of the Supervisor with date**

**Mrs. S.T. SANTHANA LAKSHMI  
Associate Professor, M.Tech (PhD),**

Department of Computer Science and  
Engineering,

Panimalar Engineering College,  
Chennai – 123

Submitted for the Project Viva – Voce examination held on \_\_\_\_\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **DECLARATION BY THE STUDENT**

We B.C.DHEVESH, M.P.DINESH KUMAR, T.LOKESH BHARATHWAJ Name of the Students 211420104065, 211420104068, 211420104149 hereby declare that this project report titled “ MAKE IT NATURAL:APPLICATION FOR MANAGINGFOOD WASTE AND EXCESS FOOD AND PLANTS IN HOMES: A STEP TO MINIMIZE FOOD WASTAGE” , under the guidance of Mrs.S.T. SANTHANA LAKSHMI is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**B.C.DHEVESH(211420104065)**

**M.P.DINESH KUMAR(211420104068)**

**T.LOKESH BHARATHWAJ(211420104149)**

## **ACKNOWLEDGEMENT**

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D.**, for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.**, for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.**, whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA,M.E., Ph.D.**, Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator Dr. N. PUGHAZENDI, M.E, PhD** and **Project Guide Mrs. S. T. SANTHANA LAKSHMI** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

**B.C.DHEVESH(211420104065)**

**M.P.DINESH KUMAR(211420104068)**

**T.LOKESH BHARATHWAJ(211420104149)**

## **ABSTRACT**

In our world today, a pressing goal is to combat food waste by repurposing available food sources within local communities and utilizing leftover food items from homes, restaurants, stores, and distribution centers. To address this challenge, an innovative app has been developed. This app serves as a comprehensive platform for managing food waste and excess food in households. It offers a multifaceted approach by converting food waste into natural fertilizer, redistributing excess food to those in need, and facilitating the purchase of the resultant fertilizer for use in home gardening. Users can interact with the app through chat-based or booking functionalities. They have the option to either manage the process independently by conversing with a designated agent or to book an agent through the app. Once booked, an agent collects the resources and oversees the further processing of both food waste and excess food. Additionally, users have the opportunity to acquire fertilizer derived from the waste generated in their own homes, thereby closing the loop on food waste management. This app not only addresses the pressing issue of food waste but also fosters community engagement and sustainability efforts on a local scale.

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>vi</b>
	<b>TABLE OF CONTENTS</b>	<b>vii</b>
	<b>LIST OF TABLES</b>	<b>ix</b>
	<b>LIST OF FIGURES</b>	<b>x</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Overview	1
1.2	Problem Definition	4
<b>2.</b>	<b>LITERATURE REVIEW</b>	<b>5</b>
<b>3.</b>	<b>THEORETICAL BACKGROUND</b>	<b>6</b>
3.1	Implementation Environment	6
3.2	System Architecture	7
3.3	Proposed Methodology	8
	3.3.1 Database Design	8
	3.3.2 Input Design (UI)	9
	3.3.3 Module Design	10

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>4.</b>	<b>SYSTEM IMPLEMENTATION</b>	15
4.1	Android studio	15
4.2	Firebase console	15
4.3	Google console	17
4.4	Google services	18
4.5	Visual studio code	18
4.6	WhatsApp Business	18
<b>5.</b>	<b>RESULTS &amp; DISCUSSION</b>	20
5.1	Performance Parameters and Testing	20
5.2	Results & Discussion	21
<b>6.</b>	<b>CONCLUSION AND FUTURE WORK</b>	26
	<b>APPENDICES</b>	27
A.1	SDG Goals	27
A.2	Source Code	28
A.3	Screen Shots	49
A.4	Plagiarism Report	52
A.5	Paper Publication	62
	<b>REFERENCES</b>	63

## **LIST OF TABLES**

<b>TABLE NO.</b>	<b>TABLE NAME</b>	<b>PAGE NO.</b>
6.1	Login and Register activity	21
6.2	Food waste and Excess food activity	21
6.3	Buy fertilizer activity	22
6.4	Test cases and Results	22

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>FIGURE NAME</b>	<b>PAGE NO.</b>
1.1	Kitchen Waste	2
1.2	Organic Fertilizer	3
3.1	System Architecture	7
3.2	Entity Relationship	8
3.3	Data Flow Diagram	10
3.4	Use case Diagram	10
3.5	Class diagram	11
3.6	Sequence diagram	11
3.7	Collaboration diagram	12
3.8	Activity diagram	12
3.9	State chart diagram	13
3.10	Package diagram	13
3.11	Component diagram	14
3.12	Deployment diagram	14
4.1	Firebase Authentication	16
4.2	Firebase Storage	16
4.3	Google maps on App & Current location maps on App	17
4.4	WhatsApp Business Account	19
5.1	Performance Parameter	20

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Overview**

#### **A) Food Waste:**

Food loss and waste refer to any food that remains uneaten throughout the entire food system, spanning from production to consumption. Globally, approximately one-third of the world's food is discarded due to multifaceted reasons, occurring at various stages including production, processing, distribution, retail, and consumption. Addressing this issue requires comprehensive solutions to promote sustainability in food systems worldwide. Efforts should focus on efficient resource utilization, minimizing wastage, ensuring food security, and conserving the environment. Collaboration among stakeholders, technological innovations, policy interventions, and public awareness campaigns are crucial in achieving these goals and creating a more resilient and equitable food system for present and future generations.

**Categories of food waste than can be separated as: -**

- 1) Kitchen waste (Converted into organic fertilizer)
- 2) Dining waste (Veg- Category 1, Non-veg- Category 3)
- 3) Non-edible/Spoiled food (Must be disposed properly)

These can be seen in homes, marriage halls, when a function takes place at home and anyplace where cooking or dining takes place.

**Kitchen wastes include: -**

- 1) Rotten vegetables and fruits.
- 2) Peels of vegetables and fruits.
- 3) Egg shell.
- 4) Scrapped portions of vegetables/fruits or slurries.
- 5) Non-veg waste before cooking (NOTE: HAS TO BE ADDED TO NON-EDIBLE)
- 6) Even bread waste.

**Dinning waste includes:-**

- 1) Inedible parts of a meal that we separate (both veg and non-veg).
- 2) Small amount of the leftovers in meal.



**Fig 1.1 Kitchen Waste**

**Non-edible/Spoiled food includes:-**

- 1) Spoiled cooked food
- 2) Non-veg waste before cooking
- 3) Non-veg waste after dining
- 4) Spoiled dairy products, oil

**B) Excess food:**

Excess food in schools refers to surplus, unopened, and unexpired items designated for inclusion in reimbursable meals but remain unused after providing breakfast and lunch to students. Efficient meal planning and management are crucial to minimize waste and optimize resource allocation. Strategies like portion control, food donation programs, and educational campaigns can address this issue, promoting sustainable food management practices within educational institutions and fostering social equity by redistributing excess food to those in need through collaboration with local organizations.

Leftovers denote the surplus food remaining unconsumed following a meal, which individuals may opt to store in containers for future consumption. Waste such as inedible bones is categorized separately from leftovers. Determining whether to keep or dispose of leftovers hinges on several factors, such as the amount and type of food, as well as the specific situation. Excess food refers to any uneaten, unexpired, and unopened food or meal that cannot be used for a present or future meal provision.

**This includes: -**

- 1) Small number of leftovers at home.
- 2) Large number of leftovers in home functions, marriage halls and in any place whereas large amount of cooking or dining takes place.

**C) Organic Fertilizer:**

**Organic/natural fertilizer is made by the following process: -**

1. Add Kitchen Waste in a Container (For continuous cycle maintain two containers).
2. Add the Browns for better fertilization (Browns like dry shredded leaves, sawdust or

cocopeat).

3. Collect some garden waste (Collecting grass clippings and leaves from your lawn and placing into your compost bin).
4. Layer with soil and repeat the process for two more times (add two more layers of above).
5. Provide Oxygen and water (You can make perforated holes and sprinkle water in your container once in every four days).
6. Once the compost reaches dark colour, it is ready to be used as a fertilizer.



**Fig 1.2 Organic Fertilizer**

## **1.2 Problem Definition**

Food waste is a pressing global issue, with an alarming one-third of all food produced for human consumption ending up in landfills. This wastage significantly contributes to greenhouse gas emissions, exacerbates resource depletion, and intensifies food insecurity worldwide. Consequently, there's an urgent need to address food waste both at its source and throughout the waste management process. To tackle this challenge effectively, we must focus on reducing food waste before it spoils and implementing efficient waste management strategies. One approach involves repurposing food waste into natural fertilizer, thereby promoting healthier soil and more sustainable agricultural practices. By recycling food waste in this manner, we not only minimize environmental harm but also create a valuable resource for enhancing food production. Furthermore, redistributing excess food to nearby homes and communities can play a pivotal role in alleviating hunger and reducing food waste during cooking and consumption. By donating surplus food to those in need, we not only prevent it from ending up in landfills but also ensure that it serves its original purpose of nourishing people. In essence, by adopting a multi-faceted approach that encompasses waste reduction, efficient management, and community engagement, we can make significant strides in mitigating the adverse impacts of food waste on the environment and society. This holistic approach not only benefits the planet but also contributes to building more resilient and equitable food systems for the future.

## **CHAPTER 2**

### **LITERATURE REVIEW**

In the realm of food waste management and sustainable living, several innovative approaches have emerged, as evidenced by recent scholarly works. One such endeavor is the development of SeVa, a groundbreaking mobile application aimed at revolutionizing food resource accessibility within local communities. Authored by Christina Varghese, Drashti Pathak, and Aparna S. Varde, SeVa provides users with a user-friendly interface to visualize and locate nearby food resources seamlessly. This app represents a significant stride in the integration of AI into smart living for urban centers, offering invaluable insights into food availability crucial for both health and sustainability. However, it's important to note that SeVa's functionality is currently limited to pandemic situations, restricting its broader application. Another notable contribution comes from Divy Chhibber, Aditi Tripathi, and Sandip Ray, who present an innovative approach to food donation distribution through their mobile application, Living Do VIR. This system utilizes cloud-based services to create a virtualized infrastructure, ensuring precise and timely food donations. While it excels in managing excess food, its focus is narrow, omitting broader aspects of food waste management. Elena-Diana Ungureau-Comanit et al. delve into the critical issue of human health risks associated with inadequate food waste management. Their study identifies potential risks and recommends management strategies, yet it falls short in providing practical implementation guidelines. Meanwhile, Shinta Oktaviana R, Diana Ambarwati Febriani, Intan Yoshana, and LR. Payanta present FoodX, a comprehensive system designed to minimize food waste. Through collaboration with communities and volunteers, FoodX ensures surplus food reaches those in need. While its features offer promise, the necessity for cooperation with NGOs and local governments may pose logistical challenges.

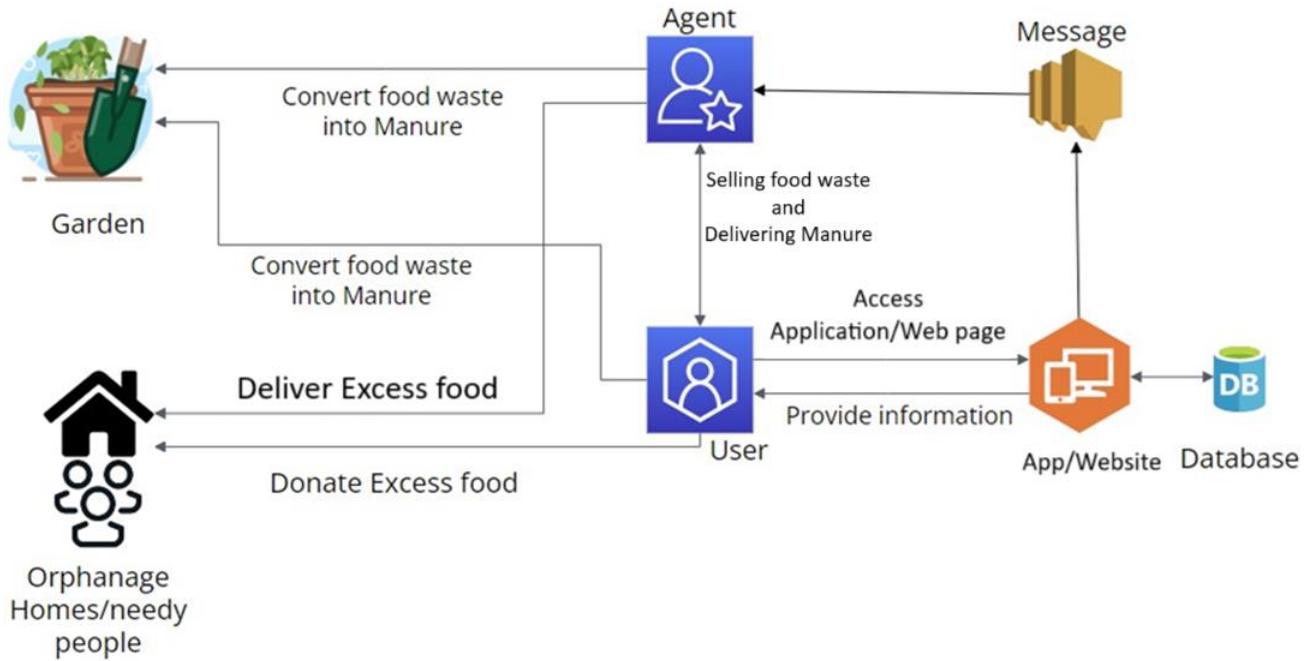
## **CHAPTER 3**

### **THEORETICAL BACKGROUND**

#### **3.1 IMPLEMENTATION ENVIRONMENT**

The application stands as a monumental solution, poised to tackle the myriad complexities of food waste management while championing sustainable agricultural practices on a global scale. Within households, users wield unprecedented authority, not merely to discard their food waste, but to metamorphose it into potent organic fertilizers, thereby fostering circularity and optimizing resource utilization. For gardeners and farmers, the app provides an effortlessly accessible platform to procure organic fertilizers, nurturing soil health and enriching biodiversity. Moreover, culinary institutions find themselves empowered to seamlessly manage and capitalize on food waste generated during culinary endeavors, while establishments spanning restaurants, hotels, and event venues streamline their waste management processes, thereby bolstering food security initiatives and substantially diminishing their carbon footprint. Educational institutions, too, are afforded the opportunity to seamlessly integrate the application into their waste management frameworks, instilling sustainable practices among students and faculty alike. By cultivating a culture of collective responsibility and community engagement in confronting global challenges, this application emerges as a catalytic force for profound positive change, propelling environmental sustainability and societal impact to unprecedeted heights. Through collaborative efforts galvanized by the app, individuals hailing from diverse backgrounds unite in a concerted effort to combat food waste at its very core, laying the groundwork for a more resilient and sustainable food system destined to endure for generations to come.

### 3.2 SYSTEM ARCHITECTURE

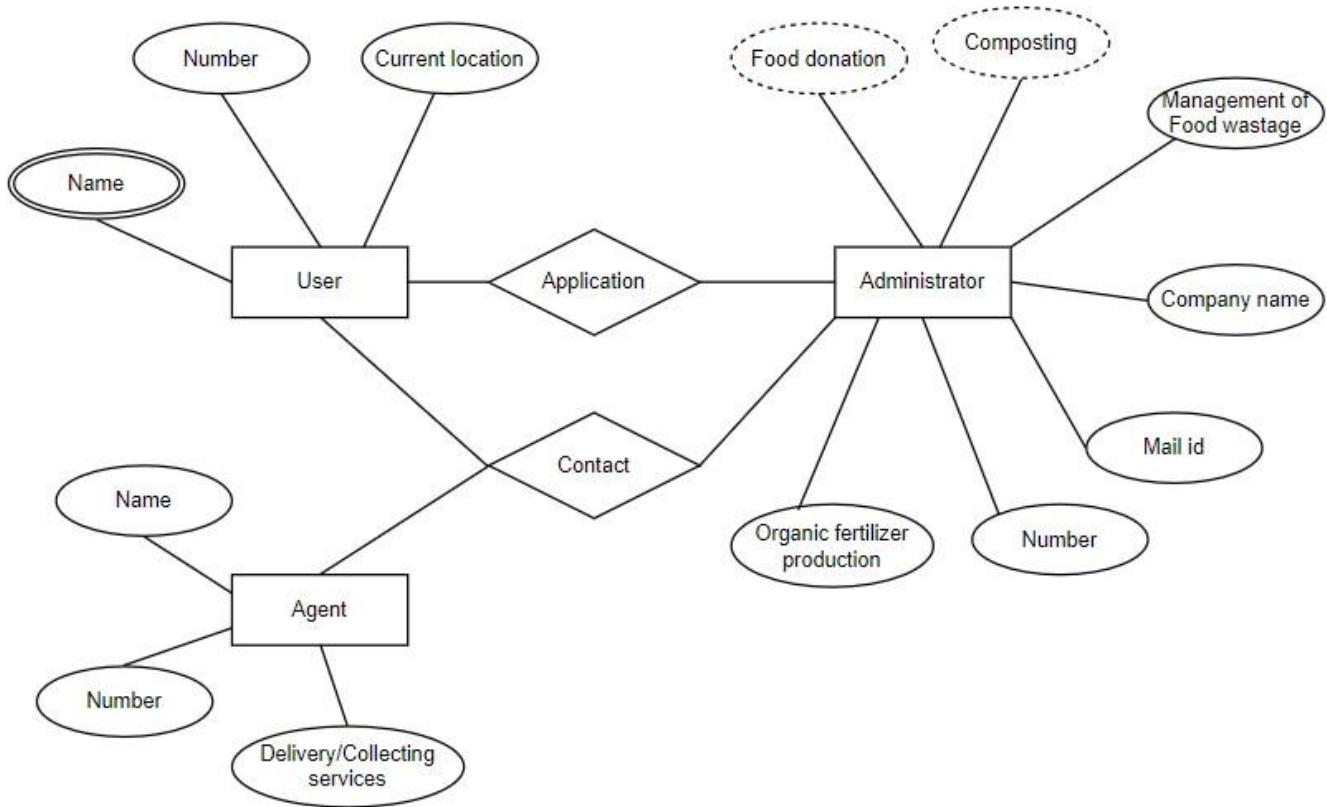


**Fig 3.1 System Architecture**

The user uses the application/website to manage his food waste or excess food or garden. User can chat with an associate to get idea what to do with the food waste that he/she has or book an agent to sell the resources or can get it back as manure in few days. To manage the excess food user can visit nearby orphanage homes to donate the food by the maps or book an agent to do the job. Users can also buy Organic fertilizer using the app , and which is made naturally by the food wastes. The booking process, which uses the database to store the user's data and helps to book an agent by alerting him or send message to an agent. The agent role here is to arrive at the user's place collect the food waste, excess food or to deliver the organic fertilizer based on which purpose the user books. And the agent takes the food waste to do the manure process, takes the excess food to the nearby orphanage homes.

### 3.3 PROPOSED METHODOLOGY

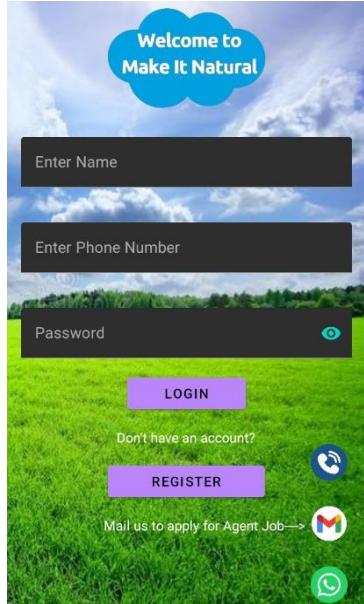
#### 3.3.1 DATABASE DESIGN



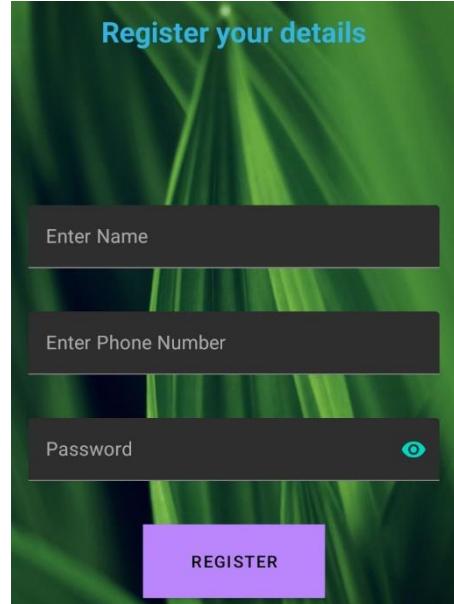
**Fig 3.2 Entity Relationship**

The diagram illustrates a system where people utilize the assistance of admin and associates to manage activities such as composting, food donation, and waste reduction. Users interact with admins through an application, where they can ask questions, seek help, or manage their food wastage. Additionally, agents play a role by facilitating delivery or collection services. The collaborative effort aims to promote environmental care and efficient resource utilization, with users, admin, associates, and agents working together through the application to achieve these goals.

### 3.3.2 Input Design (UI)



Log in



Register

UPLOAD

Upload 4 photos of your resource

Dhevish

8015153603

Select Your Choice:

1)Food waste-Sell: You get money according to your food waste.(10-100)  
 2)Food waste-Convert and get: Your food waste is converted and given to you. You will be charged certain amount.(50-100)  
 3)Excess food: your leftovers will be delivered.(No charges)

Food waste-Sell

Food waste-Convert and get

Excess food

Book Agent

**Organic Fertilizer**

Select your quantity below: There are 500g, 1kg, 2Kg, and 5Kg.

1Kg - Rs 350

SELECT QUANTITY

Products	Qty	Amt
500g - Rs 200	0	0
1Kg - Rs 350	2	700
2Kg - Rs 650	0	0
5Kg - Rs 1300	0	0
Total	2	700

Dhevish

8015153603

CHECKOUT

Buy fertilizer

### 3.3.3 Module Design

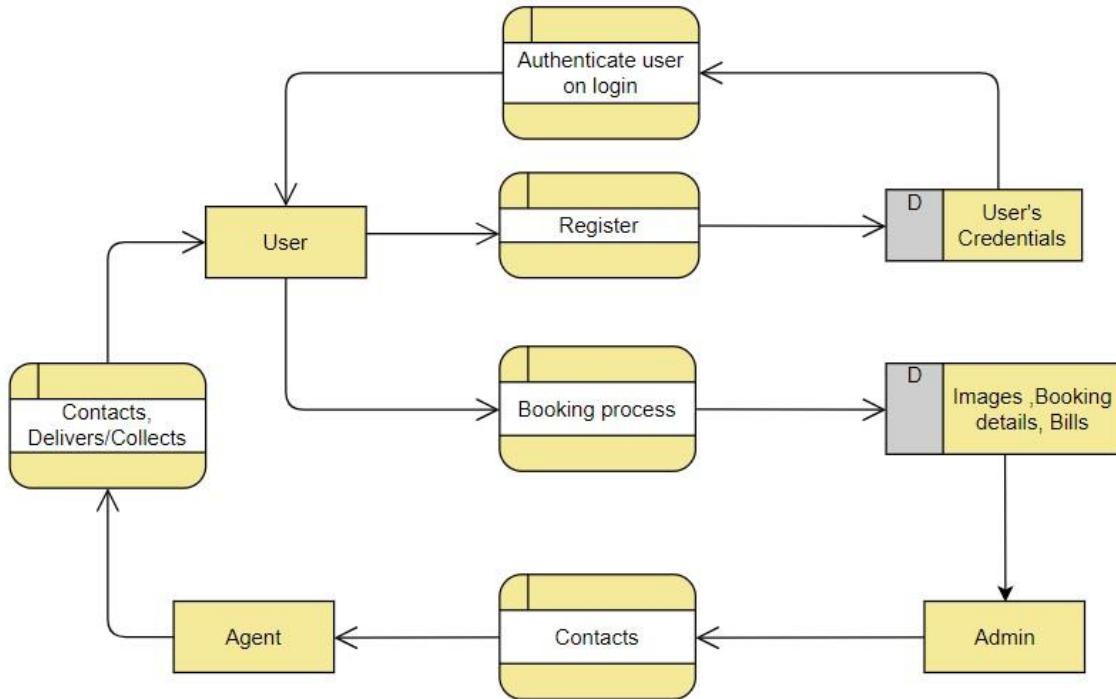


Fig 3.3 Data Flow Diagram

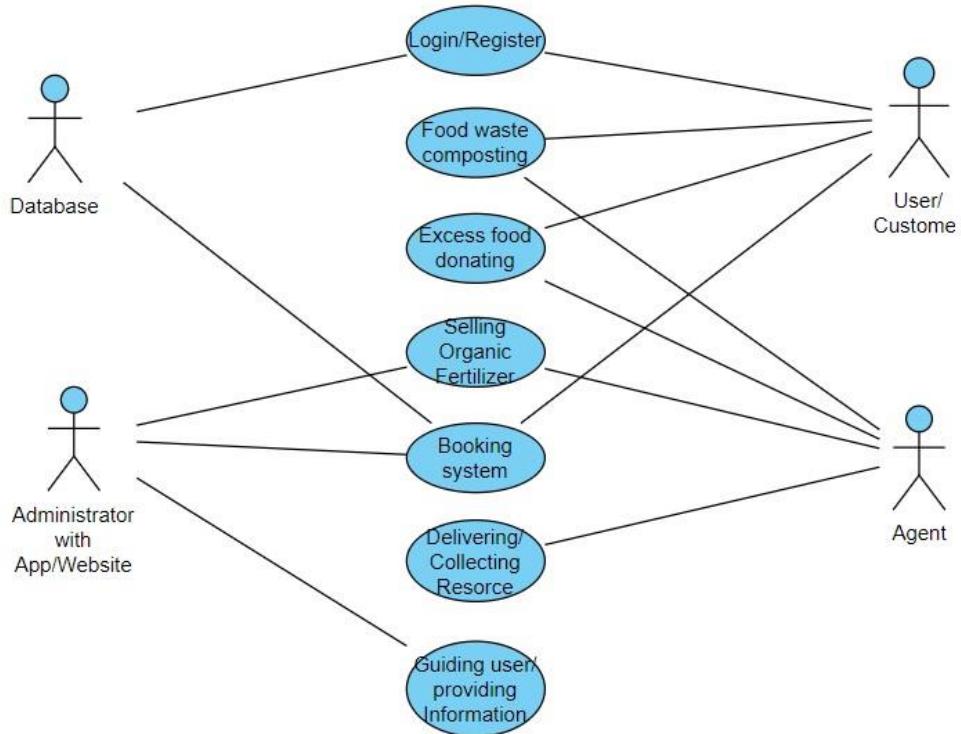
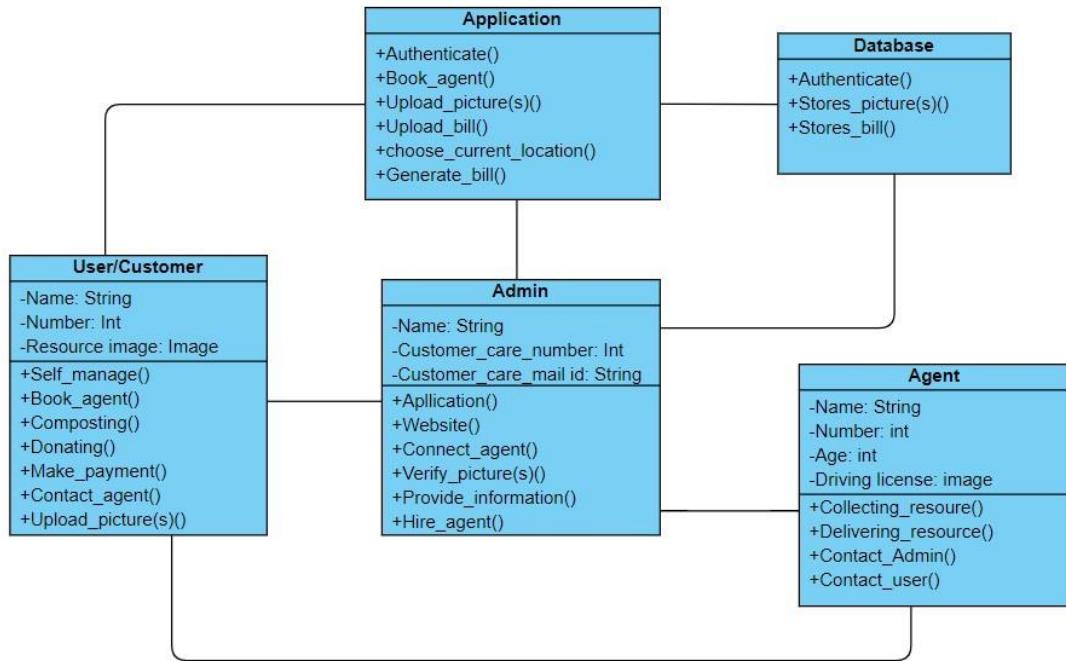
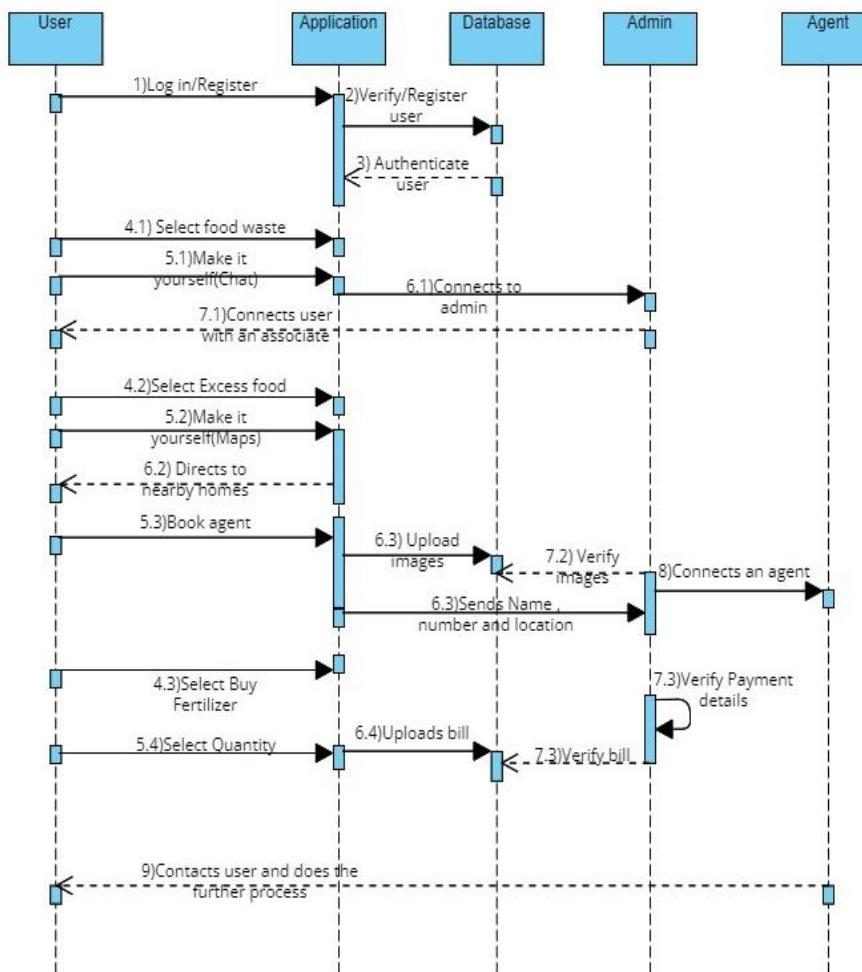


Fig 3.4 Use case Diagram



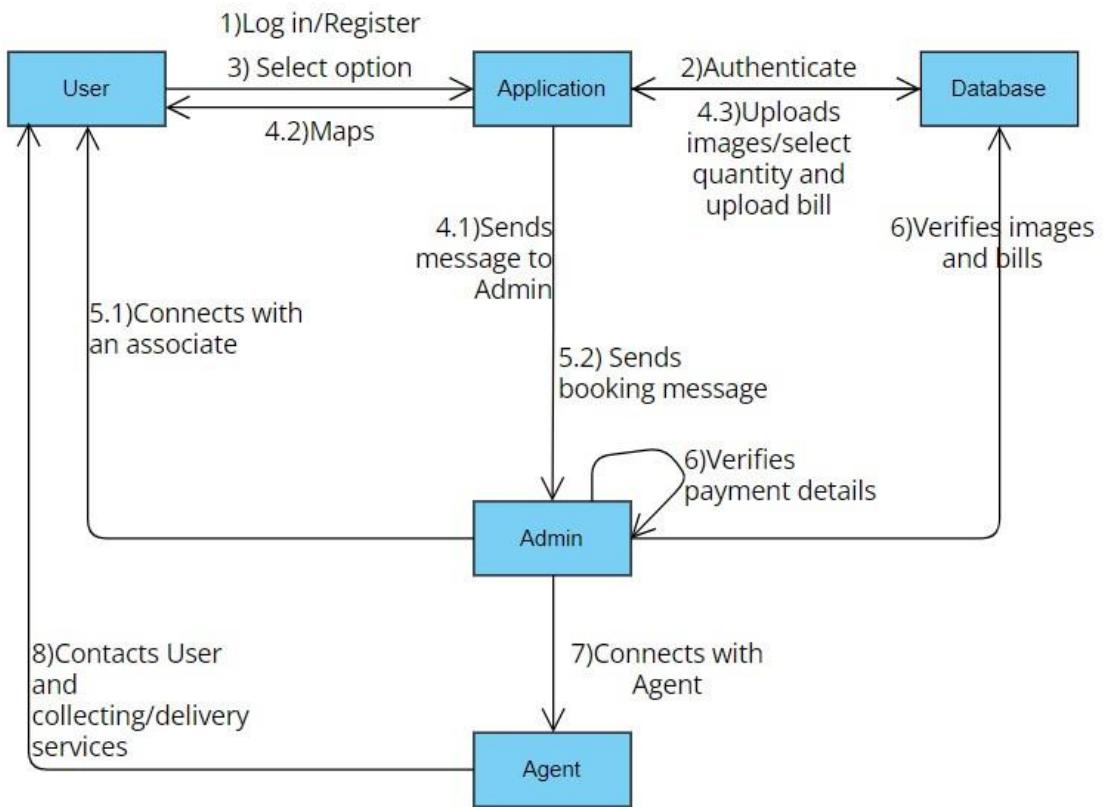
**Fig 3.5 Class diagram**



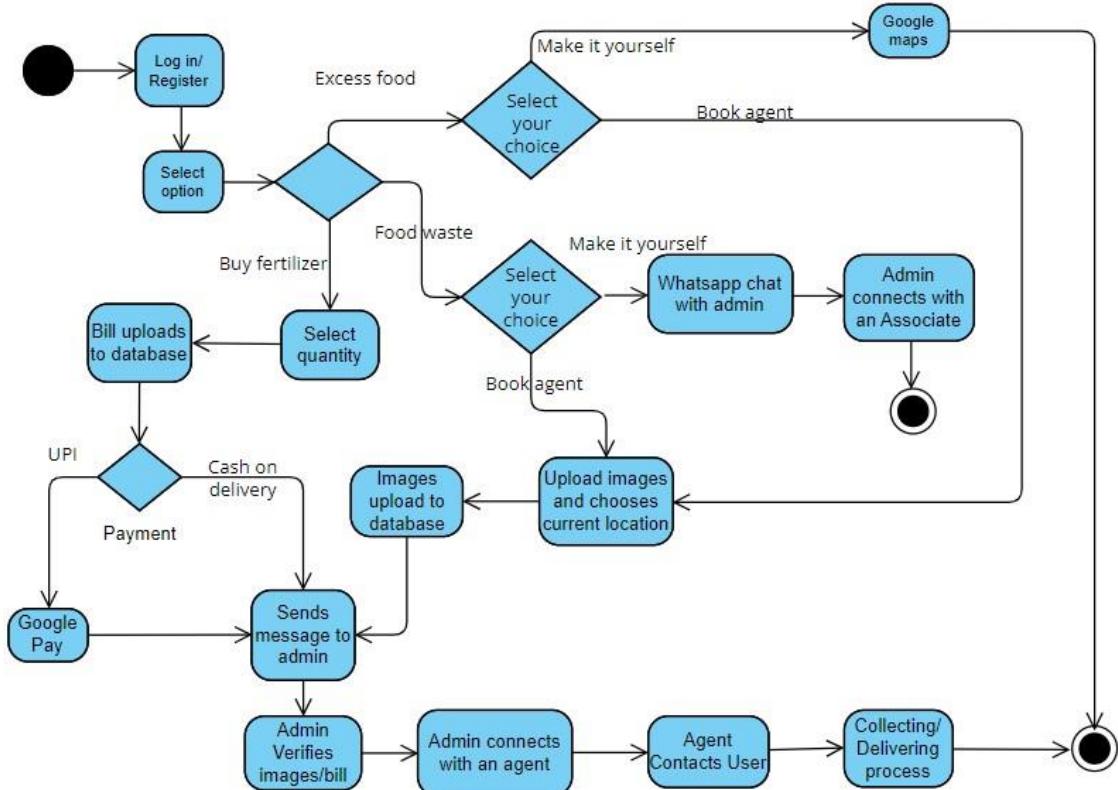
#### Further process:-

- 1) Collect Food waste/Excess food
- 2) Gives/Gets money
- 3) Converts and gives as Fertilizer
- 4) Delivers Excess food
- 5) Delivers/Organic fertilizer

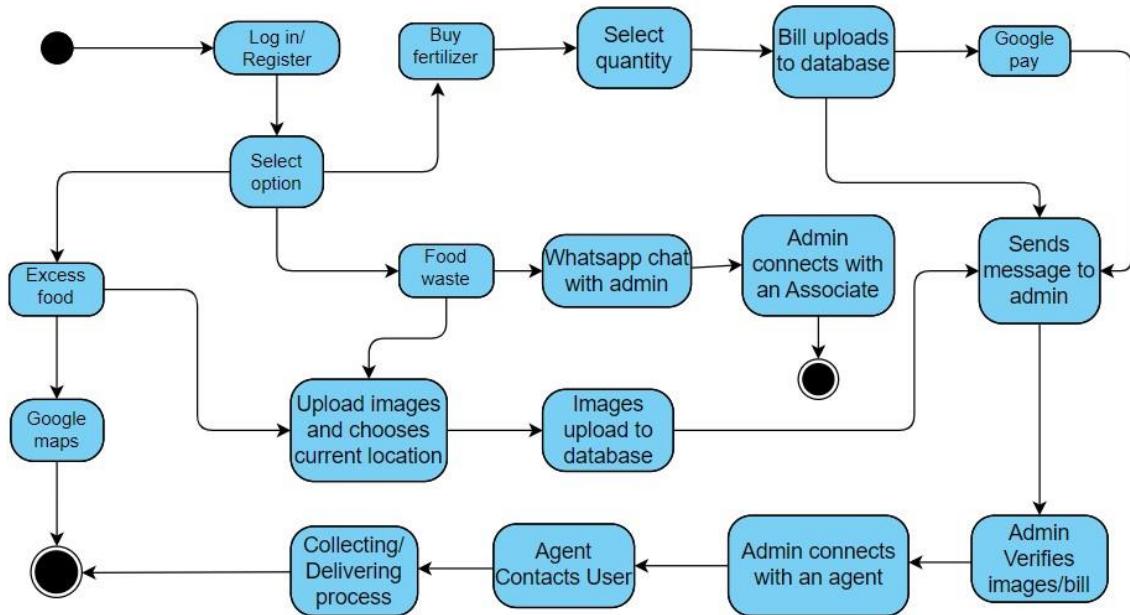
**Fig 3.6 Sequence diagram**



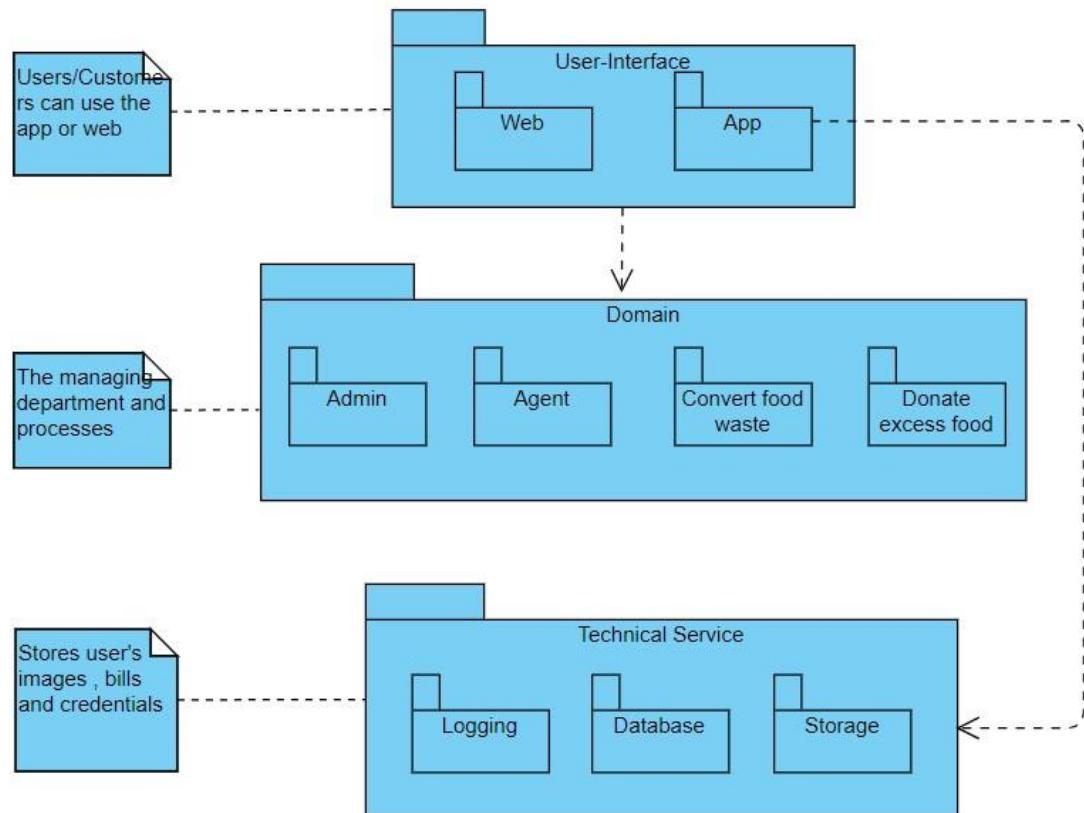
**Fig 3.7 Collaboration diagram**



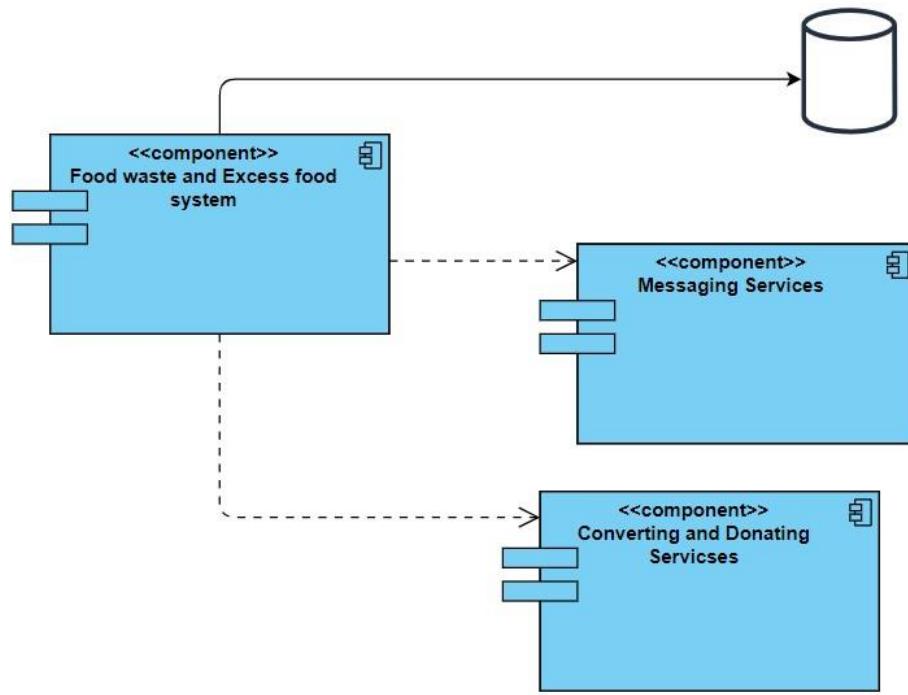
**Fig 3.8 Activity diagram**



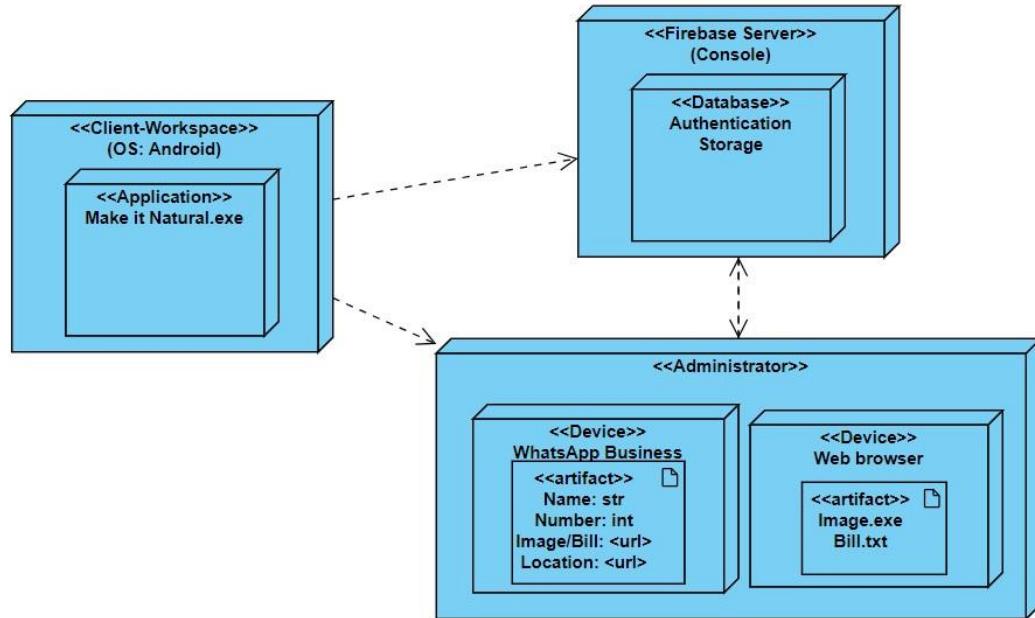
**Fig 3.9 State chart diagram**



**Fig 3.10 Package diagram**



**Fig 3.11 Component diagram**



**Fig 3.12 Deployment diagram**

## CHAPTER 4

### SYSTEM IMPLEMENTATION

#### 4.1 Android studio

Android Studio is a free development suite for Android apps. It's the official integrated development environment (IDE) for Google's Android operating system. It's built on JetBrains' IntelliJ IDEA software and is available for download on Windows, macOS, and Linux based operating systems. Android Studio is a comprehensive integrated development environment (IDE) featuring a code editor, virtual Android emulator, and code templates. It provides tools for development, debugging, testing, and performance analysis. Android Studio can run on 4GB RAM, but it lags a bit. 8GB RAM is ideal for running Android Studio as it helps in running their emulators.

#### **XML: -**

XML (Extensible Markup Language) is a lightweight markup language used in Android Studio to implement UI-related data. It's a simple, scalable way to store and organize data. Using this language, the front end of the application is built. The front-end of an application, is the layer or element that the user has the ability to use, see, and interact with through buttons, images, interactive elements, navigational menus, and text.

#### **Java: -**

Android Studio supports Java 11+ APIs without requiring a minimum API level for your app. This means that if you use an API introduced in Android 13, the code will also work on all previous versions. Some say that Java is the most suitable programming language for developing mobile apps because it allows easy multitasking and offers advanced exception-handling opportunities. Using this language, the back-end of the application is built. Java is a popular choice for building Android app backends. Here are some benefits of using Java for Android app backends. Adaptability: Java's memory management system is versatile and allows for multi-threading. Security: Java is a secure language. Cross-platform use: Java allows for cross-platform use. Developer-friendly: Java is a simple, object-oriented language that's easy to learn. Frameworks: Java developers can use frameworks like Spring and Hibernate to build scalable and secure web applications.

#### 4.2 Firebase Console

The Firebase console is a web-based interface that lets you manage your Firebase projects. You can use the console to: Add and manage your Firebase projects, Configure Firebase features for your projects, View and manage your Firebase data and Monitor your Firebase apps

- **Database:** This section lets you manage the Firebase Realtime Database for your project.

- **Storage:** This section lets you manage the Firebase Cloud Storage for your project.
- **Authentication:** This section lets you manage the Firebase Authentication for your project.
- Using this Firebase Console as a Database for Storage and Authentication, it stores user's credentials, images and bills given by the user.

The screenshot shows the Firebase Authentication interface for a project named "managefood2003". The left sidebar includes links for Project Overview, Authentication (which is selected), Firestore Database, Realtime Database, Extensions (NEW), Functions, Messaging, and Storage. The main content area is titled "Authentication" and shows a table of users. One user is listed: "Identifier" is "8015153603@example...", "Providers" is an envelope icon, "Created" is "Jan 25, 2024", "Signed In" is "Jan 28, 2024", and "User UID" is "oVtjkCMN60ZzQyZc9jYRqZQ...". A search bar at the top allows searching by email or user ID. Buttons for "Add user" and "Extensions" are also present.

**Fig 4.1 Firebase Authentication**

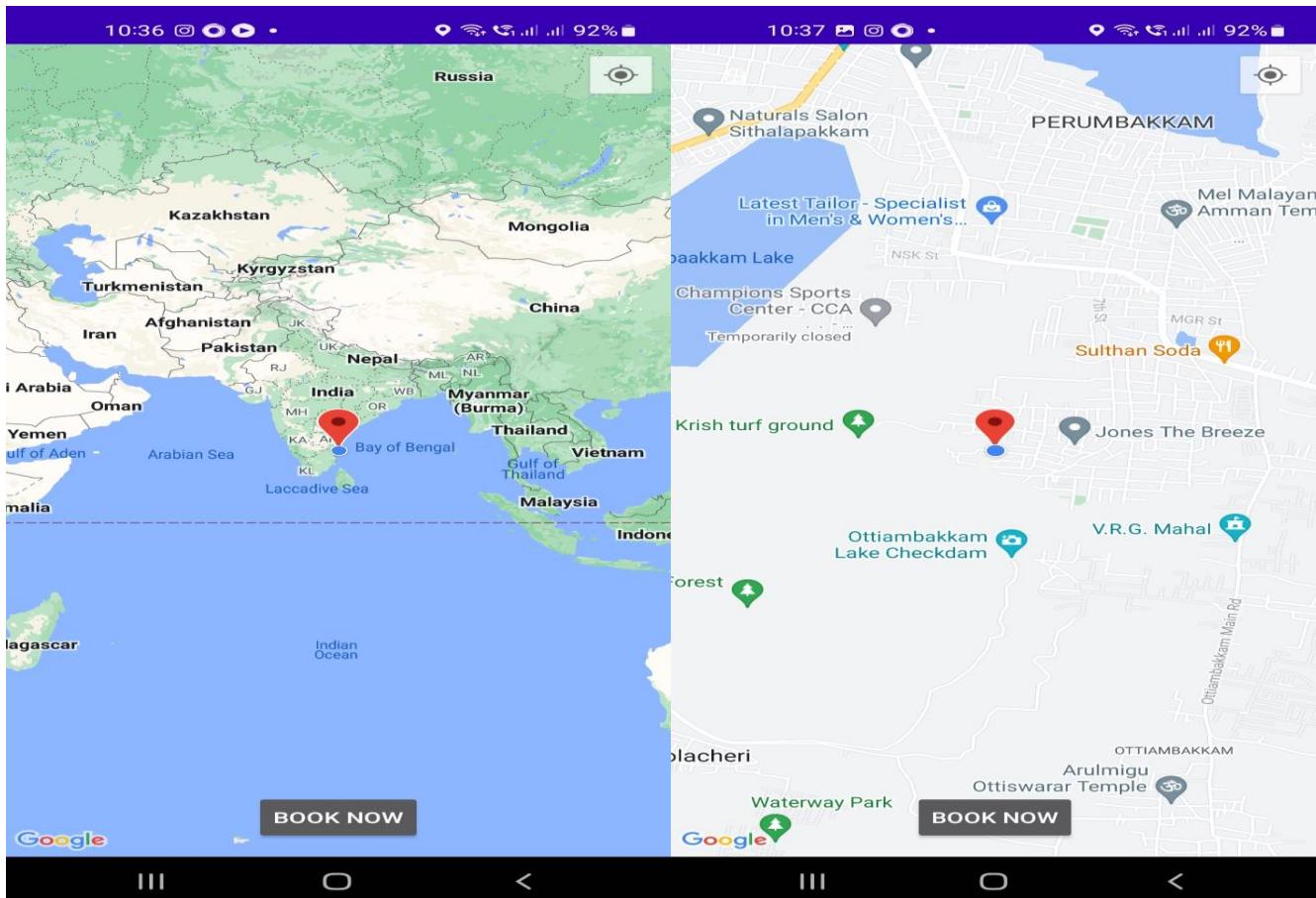
The screenshot shows the Firebase Storage interface for the same project. The left sidebar includes links for Project Overview, Storage (which is selected), Authentication, Firestore Database, Realtime Database, Extensions (NEW), Functions, Messaging, and Storage. The main content area is titled "Storage" and shows a file structure under "gs://managefood2003.appspot.com". It lists three items: "Name" (with checkboxes), "Size" (all marked as "-"), "Type" (both marked as "Folder"), and "Last modified" (both marked as "-"). The items are "bill/" and "images/". A "Upload file" button is visible at the top right of the list area.

**Fig 4.2 Firebase Storage**

### 4.3 Google Console

Google Search Console is a free service that offers invaluable insights for website owners, developers, and SEO professionals. It provides detailed information on how Google crawls, indexes, and ranks website pages in search results. Users can access data on clicks, impressions, click-through rates, and average position, helping them understand their site's performance better. Additionally, features like the

URL Inspection Tool enable users to check indexing status on individual pages and submit XML sitemaps for improved site understanding by Google. It also alerts users to any security issues or manual actions taken against their site, ensuring a secure and optimized online presence. On the other hand, the Maps JavaScript API empowers developers to integrate dynamic maps into their websites. This API offers extensive customization options, allowing developers to style maps to match their website's design and incorporate features like geocoding and directions. With the ability to add markers and info windows, developers can provide users with rich location-based information. Moreover, the API seamlessly integrates with other Google services like Places API, enabling developers to leverage a wide range of local data to enhance user experiences on their websites. Overall, both Google Search Console and the Maps JavaScript API serve as essential tools for website optimization and enhancing user engagement. Using this Google console, this application has access to map and finds current location.



**Fig 4.3 Google maps on App**

**Fig 4.4 Current location maps on App**

#### **4.4 Google Services**

Google Services refers to the programs, products, services, websites, documentation, and software offered by Google LLC and its affiliates. Google Mobile Services (GMS) is a collection of Google applications and APIs that work together to ensure a good user experience. GMS includes networking, unified communications, and security. Google Services refers to the suite of products and services offered by Google, accessible through various computer and electronic technologies, networks (both syndicated and otherwise), and systems. These encompass mobile wireless services and Internet-based services accessible through Google Sites and affiliated syndication partner sites. Specifically for this application, Gmail (utilized for agent job applications) and Google Pay (used for payment purposes) are employed.

#### **4.5 Visual studio code**

Visual Studio Code (VS Code) is a free, standalone source-code editor developed by Microsoft. It supports many programming languages, including Python, Java, C++, and JavaScript. VS Code offers a comprehensive suite of development tools, including debugging, task running, and version control capabilities. Its primary aim is to furnish developers with everything necessary for a streamlined code-build-debug cycle. VS Code is a top pick for JavaScript and web developers, with extensions to support almost any programming language. HTML, short for Hyper Text Markup Language, is the fundamental code utilized to organize the structure of a web page along with its content. It enables the structuring of content through various elements such as paragraphs, bulleted lists, images, and data tables. CSS stands for Cascading Style Sheets. It's a computer language that's used to structure and lay out web pages. CSS is a key technology of the World Wide Web, along with HTML and JavaScript. Using this CSS, the front-end of the webpage is built. JavaScript is a text-based programming language used for creating interactive web pages. Alongside HTML and CSS, it forms the core technologies of the World Wide Web, and it's essential for building both client-side and server-side functionality.

#### **4.6 WA business**

WhatsApp Business is a free app for small businesses that allows them to communicate with customers through WhatsApp. It is available on Android and iPhone. WhatsApp Business includes all the features of WhatsApp Messenger, such as: Sending multimedia, Free calls, Free international messaging, Group chat, Offline messages.

WhatsApp Business also provides tools to:

- Automate messages
- Sort messages
- Quickly respond to messages

A WhatsApp business account has been created for this application which attaches business email, business website and used for Customer-Administrator interaction and connect and contact Agent for the Collecting/Delivery service.

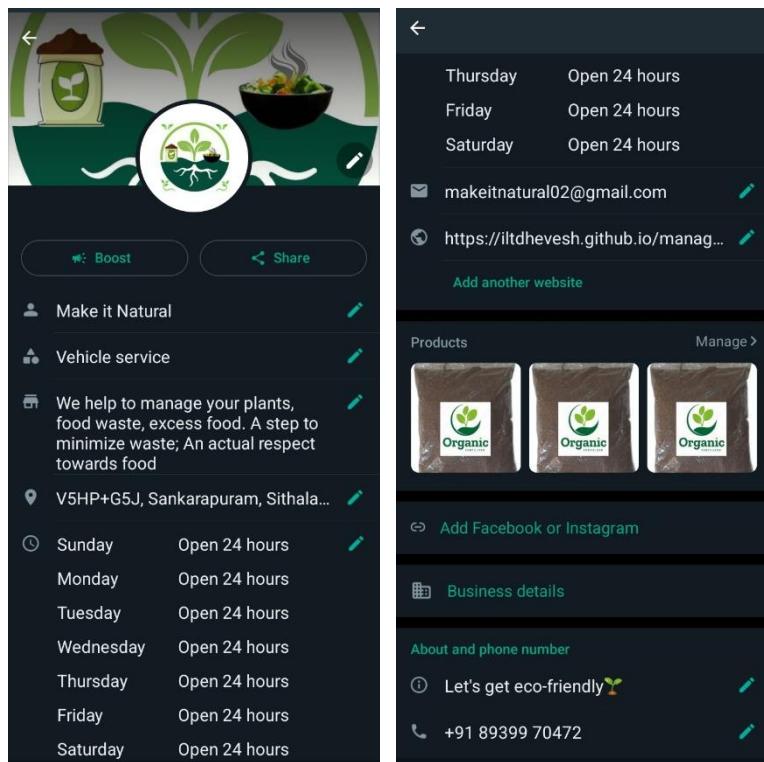
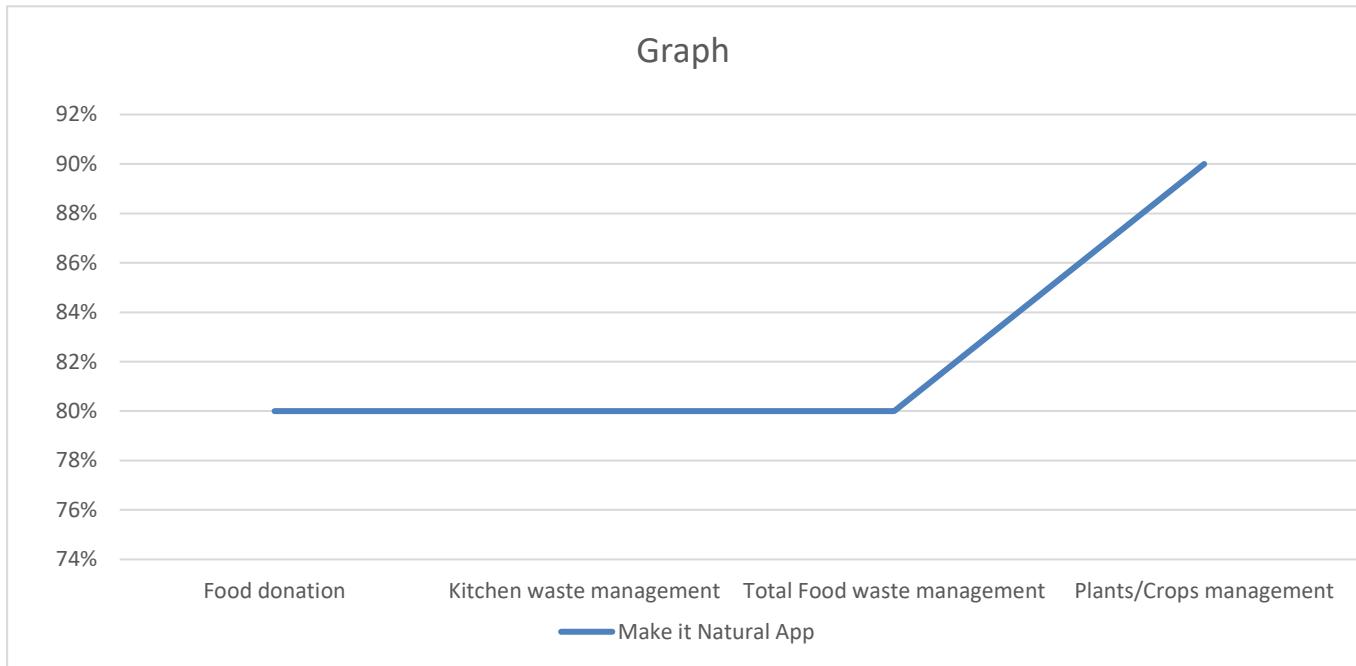


Fig 4.4 WhatsApp Business Account

## CHAPTER 5

### RESULTS & DISCUSSION

#### 5.1 PERFORMANCE PARAMETERS



**Fig 5.1 Performance parameter**

Make it Natural App is a comprehensive solution for food waste management and community support. It facilitates year-round food donation to homes or individuals in need nearby, operating consistently throughout the year, contributing to about 80% of total food waste management. Additionally, the app emphasizes efficient kitchen waste management through composting, with a focus on vegetative scraps, achieving an 80% rating in this aspect. The app goes further by offering users the option to purchase organic fertilizer made from composted food waste or convert their own waste at reduced charges, enhancing plants/crops management, achieving a 90% rating in this area. It proactively tackles food waste issues and supports sustainable practices in the community.

## 5.2 RESULTS & DISCUSSION

**Table 6.1 Login and Register activity**

<b>INPUT NAME</b>	<b>INPUT TYPE</b>	<b>DESCRIPTION</b>	<b>CONSTRAINT</b>
Name	Text	Name of the user	NOT NULL
Number	Text	Contact number of the user	NOT NULL
Password	Password	Password	NOT NULL
Log in	Button	Verify User	-
Register	Button	Registers user's credentials	-

**Table 6.2 Food waste and Excess food activity**

<b>INPUT NAME</b>	<b>INPUT TYPE</b>	<b>DESCRIPTION</b>	<b>CONSTRAINT</b>
Name	Text	Name of the user	NOT NULL
Number	Text	Contact number of the user	NOT NULL
Upload images	Images	Pictures of user's resources	NOT NULL
Type	Spinner	Which type of resource	-
Choose current location	Button	User's current location	-

**Table 6.3 Buy fertilizer activity**

<b>INPUT NAME</b>	<b>INPUT TYPE</b>	<b>DESCRIPTION</b>	<b>CONSTRAINT</b>
Name	Text	Name of the user	NOT NULL
Number	Text	Contact number of the user	NOT NULL
Fertilizer quantity in g/Kg	Spinner	Which g/Kg of Fertilizer	-
Select quantity	Button	Quantity in number	-
Choose current location	Button	User's current location	-

**Table 6.4 Test cases and Results**

<b>TESTCASE/ ACTION TO BE PERFORMED</b>	<b>EXPECTEDRESULT</b>	<b>ACTUALRESULT</b>	<b>PASS/ FAIL</b>
Clicking “Register” button	Register’s user	Register’s user	Pass
Clicking “Log in” button	Logs in user	Logs in user	Pass
Clicking “Food waste”button	Ask make it yourself or book an agent	Ask make it yourself or book an agent	Pass
Clicking “Excess food” Button	Ask make it yourself or book an agent	Ask make it yourself or book an agent	Pass

Clicking “Buy fertilizer” button	Go to Select quantity page	Go to Select quantity page	Pass
Selecting “Make it yourself” button on Food waste	Connects with Manager	Connects with Manager	Pass
Selecting “Make it yourself” button on Excess food	Go to Google maps with nearby homes	Go to Google maps with nearby homes	Pass
Clicking “Book an agent” button	Enable user to upload images , type , Autofill user’s name and number	Enable user to upload images , type , Autofill user’s name and number	Pass
Clicking “Choose current location” button	Uploads images to database and Gets user’s current location	Uploads images to database and Gets user’s current location	Pass
Clicking ”Book” Button	Uploads booking details to database and Shows user’s details and selections	Uploads booking details to database and Shows user’s details and selections	Pass
Clicking “Confirm book”button	Sends booking details link to manager	Sends booking details link to manager	Pass
Clicking “Select quantity”button on Food waste	Increment and decrement buttons appears, quantity and amount values changes and autofills user’s name and number	Increment and decrement buttons appears, quantity and amount values changes and autofills user’s name and number	Pass

Clicking "Checkout" button	Gets user's selections and current location	Gets user's selections and current location	Pass
Clicking "Payment method" Button	Asks Cash on delivery or UPI	Asks Cash on delivery or UPI	Pass
Selecting "Cash on Delivery"	Enables "Book" button	Enables "Book" button	Pass
Selecting "UPI"	Enables "Book" button and directs to google pay	Enables "Book" button and directs to google pay	Pass
Clicking "Book" button	Uploads booking details to database Sends booking details link to manager	Uploads booking details to database Sends booking details link to manager	Pass

#### User Registration and Login Functionality:

- Both registration and login processes are successful based on test cases.
- Registration and login buttons function as expected, allowing smooth user access to the system.

#### Food Waste and Excess Food Activities:

- Users can choose between "Make it yourself" or "Book an agent" options for food waste and excess food activities.
- Test cases confirm proper navigation and interaction within these activities.
- Activity scheduling feature allows users to plan their participation in advance, ensuring efficient resource utilization.

#### Booking Functionality:

- Booking process for both self-making and agent booking works smoothly, including image upload, quantity selection, and confirmation.
- Real-time availability updates ensure users can book preferred slots without conflicts.

- Integration with calendar applications enables users to synchronize booked activities with their personal schedules.

#### Location-Based Functionality:

- The system effectively utilizes user location for activity selection and booking, as demonstrated by test cases.
- Geo-fencing feature alerts users to nearby available activities based on their current location.
- Localized content and language support cater to users from different regions, improving accessibility and user experience.
- Route optimization functionality provides users with the most efficient travel routes to activity locations, saving time and resources.

#### Payment Method Integration:

- Integration of payment methods like Cash on Delivery and UPI is seamless, allowing users to choose their preferred method without issues.
- One-click payment option streamlines the checkout process, reducing friction and increasing conversion rates.
- Subscription-based payment model offers users flexibility in payment frequency and reduces transactional hassles.

#### Overall System Reliability and Robustness:

- Consistent passing of test cases indicates a reliable and robust system design and implementation.
- Automated error tracking and logging system identify and resolve issues proactively, minimizing service disruptions.
- Load balancing and failover mechanisms ensure high availability and performance during peak usage periods.

#### User Experience Considerations:

- Autofill feature and quantity selection buttons enhance user experience by reducing manual input and adding convenience.
- Intuitive user interface with clear navigation paths promotes ease of use and reduces user frustration.
- Personalized recommendations based on user preferences and past activities enhance user engagement and satisfaction.

#### Data Management and Communication:

- Uploading booking details to the database and communication with the manager are successful, indicating proper data management and communication protocols.
- Regular data sanitization processes ensure the security and privacy of user information, complying with data protection regulations.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE WORK**

#### **Conclusion**

This Project conclude that most of the food wastes can be reused for providing good nutrition to plants as a natural fertilizer by composting them. But some should be disposed of in a proper way and the excess food especially the larger number of excess foods can be given to the hungry-needy people or nearby orphanage homes. Thus, by using natural fertilizer to the plants, the plants give us healthy food. Therefore, this helps to be eco-friendly and to be good as humans with humanity by providing food to the needy people.

#### **Future Work**

- 1) Build an agent app and send ride request to agent.
- 2) Implementing this project can replace artificial manure as natural manure and major hunger can be reduced.

The above mentioned are the future enhancements that can be done to make this project much more eco-friendly and easily usable.

## **APPENDICES**

### **SDG GOALS**

- SDG 2: Zero Hunger: Your project directly contributes to reducing hunger by donating excess food to nearby homes or needy people, thereby ensuring that food resources are utilized effectively.
- SDG 12: Responsible Consumption and Production: By managing food waste and promoting the use of organic fertilizer, your project supports responsible consumption and production practices, helping to minimize waste and environmental impact.
- SDG 13: Climate Action: Food waste contributes to greenhouse gas emissions, and your project addresses this by reducing the amount of food that ends up in landfills, thus mitigating its impact on climate change.
- SDG 15: Life on Land: Converting food waste into organic fertilizer promotes soil health and biodiversity, contributing to the preservation and restoration of terrestrial ecosystems.
- SDG 17: Partnerships for the Goals: Your project involves collaboration between individuals, businesses, and institutions to address food waste and excess food, demonstrating the importance of partnerships in achieving sustainable development objectives.

## SOURCE CODE

### Login page(XML)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@drawable/ic_background"
    android:padding="16dp"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/tvTitle"
        android:layout_width="335dp"
        android:layout_height="119dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp"
        android:layout_marginBottom="24dp"
        android:gravity="center"
        android:src="@drawable/ic_title" />
    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/nameTextInputLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvTitle"
        android:layout_marginTop="8dp">
        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editTextName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="text"
            android:hint="Enter Name"/>
    </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/phoneTextInputLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/nameTextInputLayout"
        android:layout_marginTop="40dp">
        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editTextPhoneNumber"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="phone"
            android:hint="Enter Phone Number"/>
    </com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/textInputLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/phoneTextInputLayout"
    android:layout_marginTop="40dp"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="@color/car_ui_color_accent">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"/>
</com.google.android.material.textfield.TextInputLayout>
<Button
    android:id="@+id/btnLogin"
    android:layout_width="136dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textInputLayout"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp"
    android:text="Login" />
<TextView
    android:id="@+id/tvDontHaveAccount"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Don't have an account?"
    android:textColor="@color/white"
    android:layout_below="@+id/btnLogin"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp" />
<Button
    android:id="@+id/btnRegister"
    android:layout_width="180dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvDontHaveAccount"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp"
    android:text="Register" />
<ImageView
    android:id="@+id/whatsappButton"
    android:layout_width="42dp"
    android:layout_height="49dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="5dp"
    android:layout_marginBottom="9dp"
```

```

        android:clickable="true"
        android:onClick="openWhatsApp"
        android:src="@drawable/ic_whatsapp_icon" />
<ImageView
        android:id="@+id/mailButton"
        android:layout_width="42dp"
        android:layout_height="49dp"
        android:layout_above="@+id/whatsappButton"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="5dp"
        android:layout_marginBottom="21dp"
        android:clickable="true"
        android:onClick="sendMail"
        android:src="@drawable/ic_mail_icon" />
<ImageView
        android:id="@+id/phoneButton"
        android:layout_width="42dp"
        android:layout_height="49dp"
        android:layout_above="@+id/mailButton"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="5dp"
        android:layout_marginBottom="21dp"
        android:clickable="true"
        android:onClick="callPhoneNumber"
        android:src="@drawable/ic_phone_icon" />
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/btnRegister"
        android:text="Mail us to apply for Agent Job---->"
        android:textColor="@color/white"
        android:layout_marginLeft="95dp"
        android:layout_marginTop="16dp"/>
</RelativeLayout>

```

### **Map page(XML)**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MapsActivity4" />

```

```

<Button
    android:id="@+id/payButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="@color/quantum_purple"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="84dp"
    android:text="Choose Payment Method" />
<Button
    android:id="@+id/bookButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/quantum_purple"
    android:text="Book Now"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="16dp" />
</RelativeLayout>

```

### **Uploading images and details(XML)**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".UploadPhotosActivity">
    <Button
        android:id="@+id	btnUpload"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Upload"
        android:layout_marginTop="16dp"/>
    <TextView
        android:id="@+id/textViewUploadMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Upload 4 photos of your resource"
        android:layout_below="@+id	btnUpload"
        android:layout_marginTop="16dp"/>
    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Name (required)"
        android:layout_below="@+id/textViewUploadMessage"
        android:layout_marginTop="16dp"/>

```

```

<EditText
    android:id="@+id/editTextNumber"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:hint="Number (required)"
    android:layout_below="@+id/editTextName"
    android:layout_marginTop="16dp"/>
<TextView
    android:id="@+id/select"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Select Your Choice:-\n\n
1)Food waste-Sell: You get money according to your food waste.(10-100)\n
2)Food waste-Convert and get: Your food waste is converted and given to you. You will be charged certain amount.(50-100)\n
3)Excess food: your leftovers will be delivered.(No charges)"
    android:layout_below="@+id/editTextNumber"
    android:layout_marginTop="16dp"/>
<Spinner
    android:id="@+id/spinnerFoodType"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/select"
    android:layout_marginTop="16dp"/>
<Button
    android:id="@+id	btnLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Choose Current Location"
    android:layout_below="@+id/spinnerFoodType"
    android:layout_marginTop="16dp"/>
</RelativeLayout>

```

### **Buy fertilizer page(XML)**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".buyfertilizer">
    <TextView
        android:id="@+id/titleText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Organic Fertilizer"
        android:textSize="28sp"
        android:textStyle="bold"

```

```

    android:layout_centerHorizontal="true" />
<ImageView
    android:id="@+id/fertilizerImage"
    android:layout_width="142dp"
    android:layout_height="183dp"
    android:layout_below="@+id/titleText"
    android:layout_centerHorizontal="true"
    android:scaleType="centerCrop"
    android:src="@drawable/your_fertilizer_image" />
<TextView
    android:id="@+id/selectQuantityText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/fertilizerImage"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp"
    android:text="Select your quantity below: There are 500g, 1kg, 2Kg, and 5Kg."/>
<Spinner
    android:id="@+id/optionsSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/selectQuantityText"
    android:layout_marginTop="16dp"
    android:entries="@array/fertilizer_options"
    android:layout_centerHorizontal="true"/>
<Button
    android:id="@+id/addToCartButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/optionsSpinner"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="12dp"
    android:text="Select quantity"/>
<RelativeLayout
    android:id="@+id/quantityLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/optionsSpinner"
    android:layout_marginTop="12dp"
    android:visibility="gone">
    <Button
        android:id="@+id/decrementButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_alignParentStart="true"/>
    <TextView
        android:id="@+id/quantityText"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        android:layout_toStartOf="@+id/incrementButton"
        android:layout_toEndOf="@+id/decrementButton"
        android:gravity="center"/>
<Button
    android:id="@+id/incrementButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="+"
    android:layout_alignParentEnd="true"/>
</RelativeLayout>
<!-- Table to display products, prices, and saved quantities -->
<TableLayout
    android:id="@+id/productsTable"
    android:layout_width="238dp"
    android:layout_height="wrap_content"
    android:layout_below="@+id/optionsSpinner"
    android:layout_centerInParent="true"
    android:layout_marginTop="70dp">
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Products"
            android:textStyle="bold" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="32dp"
            android:text="Qty"
            android:textStyle="bold" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="32dp"
            android:text="Amt"
            android:textStyle="bold" />
    </TableRow>
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="500g - Rs 200" />
        <TextView
            android:id="@+id/qty500g"
            android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
    <TextView
        android:id="@+id/amt500g"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
</TableRow>
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1Kg - Rs 350" />
    <TextView
        android:id="@+id/qty1Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
    <TextView
        android:id="@+id/amt1Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
</TableRow>
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2Kg - Rs 650" />
    <TextView
        android:id="@+id/qty2Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
    <TextView
        android:id="@+id/amt2Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
</TableRow>
<TableRow>
    <TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="5Kg - Rs 1300" />
    <TextView
        android:id="@+id/qty5Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
    <TextView
        android:id="@+id/amt5Kg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
</TableRow>
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Total" />
    <!-- Add an empty TextView for the "Qty" column in the "Total" row -->
    <TextView
        android:id="@+id/totalQty"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
    <!-- Add the TextView for the "Amt" column in the "Total" row -->
    <TextView
        android:id="@+id/totalAmt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="32dp"
        android:text="0" />
</TableRow>
</TableLayout>
<EditText
    android:id="@+id/editTextName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="2dp"
    android:layout_marginEnd="16dp"
    android:layout_below="@+id/productsTable"
    android:hint="Name"
    android:textColor="@color/white"
    android:inputType="textPersonName"

```

```

        android:minHeight="48dp" />
<EditText
    android:id="@+id/editTextNumber"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editTextName"
    android:layout_marginStart="16dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="16dp"
    android:hint="Number"
    android:textColor="@color/white"
    android:inputType="phone"
    android:minHeight="48dp" />
<Button
    android:id="@+id/checkoutButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editTextNumber"
    android:layout_marginTop="2dp"
    android:text="Checkout"
    android:layout_centerHorizontal="true"/>
</RelativeLayout>

```

### **Login page(Java)**

```

package com.example.MakeITNatural;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
public class MainActivity extends AppCompatActivity {
private EditText editTextName, editTextPhoneNumber, editTextPassword;
private Button btnLogin, btnRegister;
private TextView tvRg;
private LinearLayout layoutLoggedIn;
private DataManager dataManager;
private FirebaseAuth mAuth;
@Override

```

```

protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
dataManager = new DataManager(this);
mAuth = FirebaseAuth.getInstance();
editTextName = findViewById(R.id.editTextName);
editTextPhoneNumber = findViewById(R.id.editTextPhoneNumber);
editTextPassword = findViewById(R.id.editTextPassword);
btnLogin = findViewById(R.id.btnLogin);
btnRegister = findViewById(R.id.btnRegister);
tvRg = findViewById(R.id.tvDontHaveAccount);
if (mAuth.getCurrentUser() != null) {
Intent intent = new Intent(MainActivity.this, MainActivity2.class);
startActivity(intent);
}
btnLogin.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
String name = editTextName.getText().toString().trim();
String phoneNumber = editTextPhoneNumber.getText().toString().trim();
String password = editTextPassword.getText().toString().trim();
dataManager.saveUserData(name, phoneNumber);
if (name.isEmpty() || phoneNumber.isEmpty() || password.isEmpty()) {
showToast("Please fill in all the fields.");
} else {
loginUser(name, phoneNumber, password);
}
}
});
btnRegister.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
startActivity(intent);
}
});
}

public void openWhatsApp(View view) {
String phoneNumber = "8939970472";
Uri uri = Uri.parse("smsto:" + phoneNumber);
Intent intent = new Intent(Intent.ACTION_SENDTO, uri);
intent.setPackage("com.whatsapp");
startActivity(intent);
}
public void sendMail(View view) {
String email = "makeitnatural02@gmail.com";
Intent intent = new Intent(Intent.ACTION_SENDTO);
intent.setData(Uri.parse("mailto:" + email));
}

```

```
startActivity(intent);
}
public void callPhoneNumber(View view) {
String phoneNumber = "tel:" + "8939970472";
Intent intent = new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse(phoneNumber));
startActivity(intent);
}
private void showToast(String message) {
Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
}
private void loginUser(String name, String phoneNumber, String password) {
mAuth.signInWithEmailAndPassword(phoneNumber + "@example.com", password)
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {
showToast("Login Successful!");
Intent intent = new Intent(MainActivity.this, MainActivity2.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
startActivity(intent);
} else {
showToast("Authentication failed. Please check your credentials.");
}
}
});
}
}
```

## **Uploading images and details(Java)**

```
package com.example.MakeITNatural;
import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import java.util.ArrayList;
import java.util.List;
public class UploadPhotosActivity extends BaseActivity {
```

```

private static final int LOCATION_PERMISSION_REQUEST_CODE = 123;
private static final int PICK_IMAGES_REQUEST_CODE = 456;
private static final int MAX_SELECTED_IMAGES = 4;
private ProgressDialog progressDialog;
StorageReference storageReference;
private List<Uri> selectedImageUris = new ArrayList<>();
private EditText editTextName;
private EditText editTextNumber;
private String storedName;
private String storedNumber;
private DataManager dataManager;
private Spinner spinnerFoodType;
private ArrayAdapter<CharSequence> foodTypeAdapter;
private String selectedFoodType;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_upload_photos);
    dataManager = new DataManager(this);
    editTextName = findViewById(R.id.editTextName);
    editTextNumber = findViewById(R.id.editTextNumber);
    spinnerFoodType = findViewById(R.id.spinnerFoodType);
    foodTypeAdapter = ArrayAdapter.createFromResource(this, R.array.food_types,
        android.R.layout.simple_spinner_item);
    foodTypeAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinnerFoodType.setAdapter(foodTypeAdapter);
    storedName = dataManager.getUserName();
    storedNumber = dataManager.getUserPhoneNumber();
    editTextName.setText(storedName);
    editTextNumber.setText(storedNumber);
    spinnerFoodType.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int position, long id) {
            selectedFoodType = (String) adapterView.getItemAtPosition(position);
        }
        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    });
    Button btnLocation = findViewById(R.id.btnLocation);
    btnLocation.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            openMapView();
        }
    });
    Button btnUpload = findViewById(R.id.btnUpload);
    btnUpload.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View view) {
    openImagePicker();
}
});

}

private void openMapView() {
    if (selectedImageUris.size() != MAX_SELECTED_IMAGES) {
        Toast.makeText(this, "Please select exactly 4 images before choosing the location",
Toast.LENGTH_SHORT).show();
        return;
    }
    progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Please wait, images are uploading...");
    progressDialog.setCancelable(false);
    progressDialog.show();
    storageReference = FirebaseStorage.getInstance().getReference().child("images");
    List<String> imageUrls = new ArrayList<>();
    for (int i = 0; i < selectedImageUris.size(); i++) {
        Uri imageUri = selectedImageUris.get(i);
        String imageName = "image_" + System.currentTimeMillis();
        StorageReference imageRef = storageReference.child(imageName);
        imageRef.putFile(imageUri)
            .addOnSuccessListener(taskSnapshot -> {
                imageRef.getDownloadUrl().addOnSuccessListener(uri -> {
                    String imageUrl = uri.toString();
                    imageUrls.add(imageUrl);
                    if (imageUrls.size() == MAX_SELECTED_IMAGES) {
                        passDataToMapActivity(imageUrls);
                    }
                });
            });
    }
    .addOnFailureListener(e -> {
        Toast.makeText(this, "Failed to upload image: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
    });
}
}

private void passDataToMapActivity(List<String> imageUrls) {
    if (imageUrls.size() != MAX_SELECTED_IMAGES) {
        Toast.makeText(this, "Please wait for all images to upload", Toast.LENGTH_SHORT).show();
        return;
    }
    if (progressDialog != null && progressDialog.isShowing()) {
        progressDialog.dismiss();
    }
    Intent mapIntent = new Intent(this, MapsActivity2.class);
    mapIntent.putExtra("name", storedName);
}

```

```

        mapIntent.putExtra("number", storedNumber);
        mapIntent.putExtra("foodType", selectedFoodType);
        mapIntent.putExtra("imageLink1", imageUrls.get(0));
        mapIntent.putExtra("imageLink2", imageUrls.get(1));
        mapIntent.putExtra("imageLink3", imageUrls.get(2));
        mapIntent.putExtra("imageLink4", imageUrls.get(3));
        mapIntent.putParcelableArrayListExtra("selectedImages", new ArrayList<>(selectedImageUrils));
        startActivity(mapIntent);
    }
}

private void openImagePicker() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    intent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, true);
    startActivityForResult(intent, PICK_IMAGES_REQUEST_CODE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == PICK_IMAGES_REQUEST_CODE && resultCode == RESULT_OK &&
data != null) {
        int selectedImageCount = 0;
        if (data.getClipData() != null) {
            selectedImageCount = data.getClipData().getItemCount();
        } else if (data.getData() != null) {
            selectedImageCount = 1;
        }
        if (selectedImageCount <= MAX_SELECTED_IMAGES) {
            selectedImageUrils.clear();
            if (data.getClipData() != null) {
                for (int i = 0; i < selectedImageCount; i++) {
                    Uri imageUri = data.getClipData().getItemAt(i).getUri();
                    selectedImageUrils.add(imageUri);
                }
            } else if (data.getData() != null) {
                Uri imageUri = data.getData();
                selectedImageUrils.add(imageUri);
            }
        } else {
            Toast.makeText(this, "You can select up to 4 photos", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

### **Buy fertilizer page(Java)**

```

package com.example.MakeITNatural;
import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.Map;
public class buyfertilizer extends BaseActivity {
    private Spinner optionsSpinner;
    private Button addToCartButton;
    private RelativeLayout quantityLayout;
    private Button decrementButton;
    private TextView quantityText;
    private Button incrementButton;
    private TableLayout productsTable;
    private Button checkoutButton;
    EditText editTextName;
    EditText editTextNumber;
    private TextView totalAmt; // Added TextView for total amount
    private TextView totalQty; // Added TextView for total quantity
    private ProgressDialog progressDialog;
    private Map<String, Integer> quantityMap = new HashMap<>();
    private String selectedOption;
    private int quantity = 1;
    private DataManager dataManager;
    private FirebaseStorage storage = FirebaseStorage.getInstance();
    String timestamp = String.valueOf(System.currentTimeMillis());
    String fileName = "bill_" + timestamp + ".txt";
    private StorageReference storageReference = storage.getReference().child("bill/" + fileName);
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.buy_fertilizer);
    editTextName = findViewById(R.id.editTextName);
    editTextNumber = findViewById(R.id.editTextNumber);
    dataManager = new DataManager(this);
    ImageView fertilizerImage = findViewById(R.id.fertilizerImage);
    fertilizerImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(buyfertilizer.this, ImageViewerActivity.class);
            intent.putExtra("IMAGE_RESOURCE", R.drawable.your_fertilizer_image);
            startActivity(intent);
        }
    });
    optionsSpinner = findViewById(R.id.optionsSpinner);
    addToCartButton = findViewById(R.id.addToCartButton);
    quantityLayout = findViewById(R.id.quantityLayout);
    decrementButton = findViewById(R.id.decrementButton);
    quantityText = findViewById(R.id.quantityText);
    incrementButton = findViewById(R.id.incrementButton);
    productsTable = findViewById(R.id.productsTable);
    checkoutButton = findViewById(R.id.checkoutButton);
    totalAmt = findViewById(R.id.totalAmt);
    totalQty = findViewById(R.id.totalQty);
    String storedName = dataManager.getUserName();
    String storedNumber = dataManager.getUserPhoneNumber();
    editTextName.setText(storedName);
    editTextNumber.setText(storedNumber);
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
        this,
        R.array.fertilizer_options,
        android.R.layout.simple_spinner_item
    );
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    optionsSpinner.setAdapter(adapter);
    optionsSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> adapterView, View view, int position, long id) {
            saveQuantityForOption(selectedOption, quantity);
            selectedOption = optionsSpinner.getSelectedItem().toString();
            quantity = getQuantityForOption(selectedOption);
            updateUI();
        }
        @Override
        public void onNothingSelected(AdapterView<?> adapterView) {
        }
    });
}

```

```

addToCartButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Show the quantity layout and hide the "Add to Cart" button
        quantityLayout.setVisibility(View.VISIBLE);
        addToCartButton.setVisibility(View.GONE);
        quantityText.setText(String.valueOf(quantity));
        updateQuantityInTable(selectedOption, quantity);
        updateAmountForOption(selectedOption, quantity);
        updateTotalQuantity();
    }
});

decrementButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        decrementQuantity();
    }
});

incrementButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        incrementQuantity();
    }
});

checkoutButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        handleCheckout();
    }
});

private void saveQuantityForOption(String option, int quantity) {
    quantityMap.put(option, quantity);
}

private int getQuantityForOption(String option) {
    return quantityMap.containsKey(option) ? quantityMap.get(option) : 1;
}

private void decrementQuantity() {
    if (quantity > 0) {
        if (quantity > 1) {
            quantity--;
            updateQuantityText();
            updateQuantityInTable(selectedOption, quantity);
            updateAmountForOption(selectedOption, quantity);
        } else {
            // If quantity is 1, set it to 0
            quantity = 0;
            updateQuantityText();
        }
    }
}

```

```

        updateQuantityInTable(selectedOption, quantity);

        // Show the "Add to Cart" button and hide the quantity layout
        showAddToCartButton();
    }
}

updateAmountForOption(selectedOption, quantity);
updateTotalQuantity();
}

private void incrementQuantity() {
    quantity++;
    updateQuantityText();
    updateQuantityInTable(selectedOption, quantity);
    updateAmountForOption(selectedOption, quantity);
    updateTotalQuantity();
}

private void updateQuantityText() {
    quantityText.setText(String.valueOf(quantity));
}

private void showAddToCartButton() {
    addToCartButton.setVisibility(View.VISIBLE);
    quantityLayout.setVisibility(View.GONE);
}

private void updateUI() {
    if (quantityMap.containsKey(selectedOption)) {
        // If the selected option has a saved quantity, update the UI to show that quantity
        quantity = quantityMap.get(selectedOption);
        quantityText.setText(String.valueOf(quantity));

        // Update the quantity in the corresponding TextView in the TableLayout
        updateQuantityInTable(selectedOption, quantity);
        updateAmountForOption(selectedOption, quantity);

        showAddToCartButton();
    } else {
        // If the selected option does not have a saved quantity, show the "Add to Cart" button
        addToCartButton.setVisibility(View.VISIBLE);
        quantityLayout.setVisibility(View.GONE);
    }
}

private void updateQuantityInTable(String option, int quantity) {
    switch (option) {
        case "500g - Rs 200":
            ((TextView) findViewById(R.id.qty500g)).setText(String.valueOf(quantity));
            break;
        case "1Kg - Rs 350":
            ((TextView) findViewById(R.id.qty1Kg)).setText(String.valueOf(quantity));
    }
}

```

```

        break;
    case "2Kg - Rs 650":
        ((TextView) findViewById(R.id.qty2Kg)).setText(String.valueOf(quantity));
        break;
    case "5Kg - Rs 1300":
        ((TextView) findViewById(R.id.qty5Kg)).setText(String.valueOf(quantity));
        break;
    }
}
private int getPricePerUnit(String option) {
    switch (option) {
        case "500g - Rs 200":
            return 200;
        case "1Kg - Rs 350":
            return 350;
        case "2Kg - Rs 650":
            return 650;
        case "5Kg - Rs 1300":
            return 1300;
        default:
            return 0;    }    }
private void updateAmountForOption(String option, int quantity) {
    int pricePerUnit = getPricePerUnit(option);
    int amount = quantity > 0 ? quantity * pricePerUnit : 0; // Set amount to 0 when quantity is 0
    updateAmountInTable(option, amount);
    updateTotalAmount();    }
private void updateAmountInTable(String option, int amount) {
    switch (option) {
private void updateTotalAmount() {
    int totalAmount = calculateTotalAmount();
    totalAmt.setText(String.valueOf(totalAmount));    }
private int calculateTotalAmount() {
    int totalAmount = 0;
    totalAmount += Integer.parseInt(((TextView) findViewById(R.id.amt500g)).getText().toString());
    totalAmount += Integer.parseInt(((TextView) findViewById(R.id.amt1Kg)).getText().toString());
    totalAmount += Integer.parseInt(((TextView) findViewById(R.id.amt2Kg)).getText().toString());
    totalAmount += Integer.parseInt(((TextView) findViewById(R.id.amt5Kg)).getText().toString());
    return totalAmount;    }
private void updateTotalQuantity() {
    int totalQuantity = calculateTotalQuantity();
    totalQty.setText(String.valueOf(totalQuantity));    }
private void handleCheckout() {
    StringBuilder tableData = new StringBuilder();
    tableData.append("Products\tQty\tAmt\n");
    tableData.append("500g - Rs 200\t")
        .append(((TextView) findViewById(R.id.qty500g)).getText().toString()).append("\t")
        .append(((TextView) findViewById(R.id.amt500g)).getText().toString()).append("\n");
    tableData.append("1Kg - Rs 350\t")
}

```

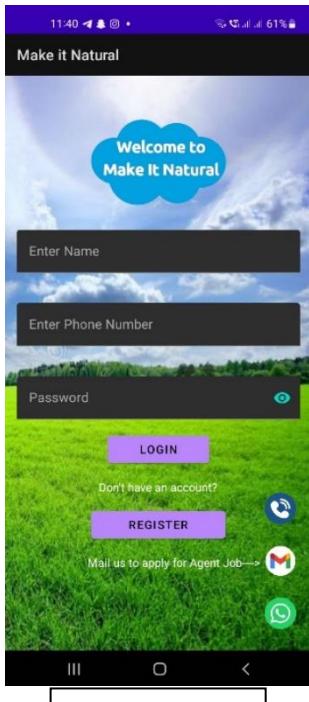
```

.append(((TextView) findViewById(R.id.qty1Kg)).getText().toString()).append("\t")
.append(((TextView) findViewById(R.id.amt1Kg)).getText().toString()).append("\n");
tableData.append("2Kg - Rs 650\t")
.append(((TextView) findViewById(R.id.qty2Kg)).getText().toString()).append("\t")
.append(((TextView) findViewById(R.id.amt2Kg)).getText().toString()).append("\n");
tableData.append("5Kg - Rs 1300\t")
.append(((TextView) findViewById(R.id.qty5Kg)).getText().toString()).append("\t")
.append(((TextView) findViewById(R.id.amt5Kg)).getText().toString()).append("\n");
tableData.append("Total\t")
.append(((TextView) findViewById(R.id.totalQty)).getText().toString()).append("\t")
.append(((TextView) findViewById(R.id.totalAmt)).getText().toString()).append("\n");
uploadToFirebaseStorage(tableData.toString());
}

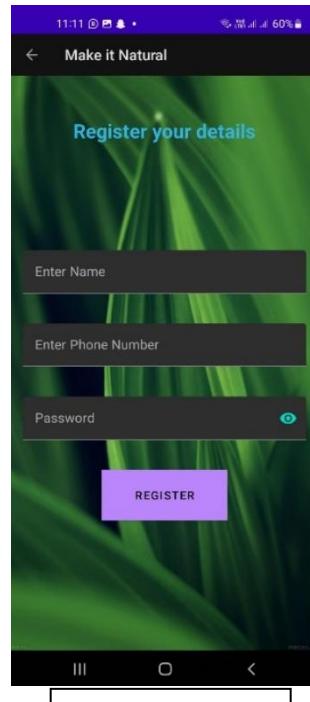
private void uploadToFirebaseStorage(String data) {
    InputStream stream = new ByteArrayInputStream(data.getBytes(StandardCharsets.UTF_8));
    UploadTask uploadTask = storageReference.putStream(stream);
    progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Please wait, your bill is being generated...");
    progressDialog.setCancelable(false);
    progressDialog.show();
    uploadTask.addOnCompleteListener(new OnCompleteListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {
            if (task.isSuccessful()) {
                StorageReference fileRef = storage.getReference().child("bill/"+"bill_" + timestamp +
'.txt');
                fileRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {
                        String downloadUrl = uri.toString();
                        String totalAmtValue = totalAmt.getText().toString();
                        Intent intent = new Intent(buyfertilizer.this, MapsActivity4.class);
                        intent.putExtra("FIREBASE_STORAGE_URL", downloadUrl);
                        intent.putExtra("TOTAL_AMOUNT", totalAmtValue);
                        startActivity(intent);
                        progressDialog.dismiss();
                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                    }
                });
            } else {
                progressDialog.dismiss();
            }
        }
    });
}
}

```

## SCREEN SHOTS



Login Module



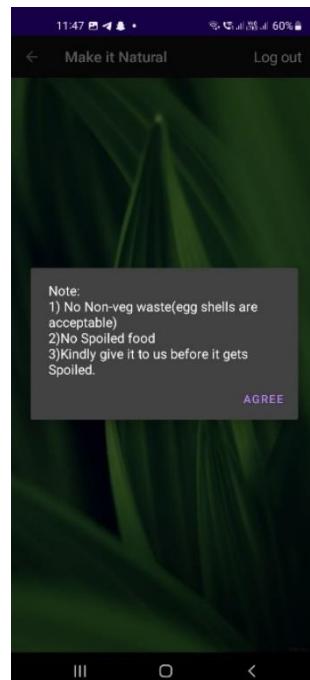
Register Module



Main Module



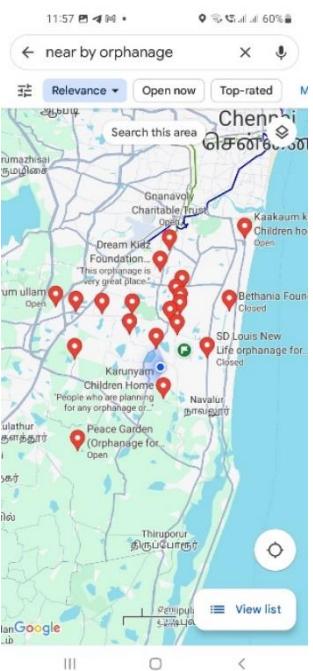
Food Waste clicked



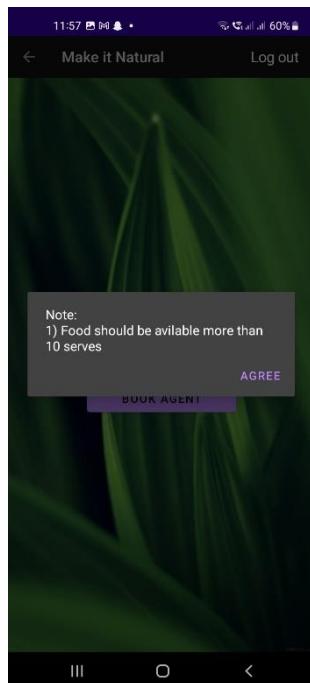
Book agent (Food waste)



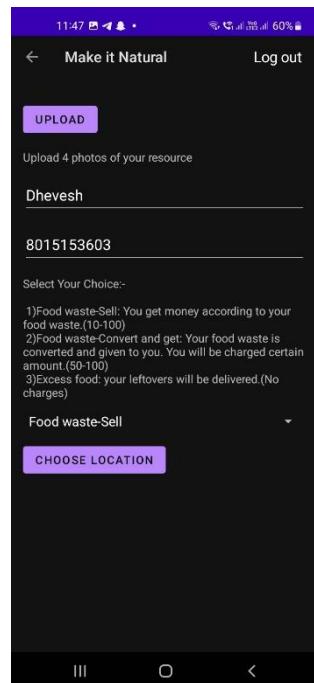
Excess food clicked



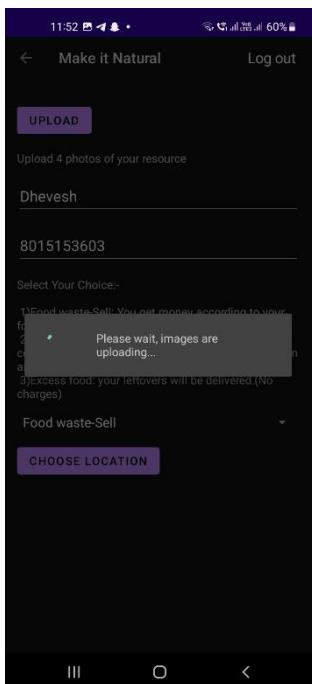
Make it Yourself clicked (nearby homes)



Book agent (Excess food)



Upload Module



Choose location clicked



Choosing current location



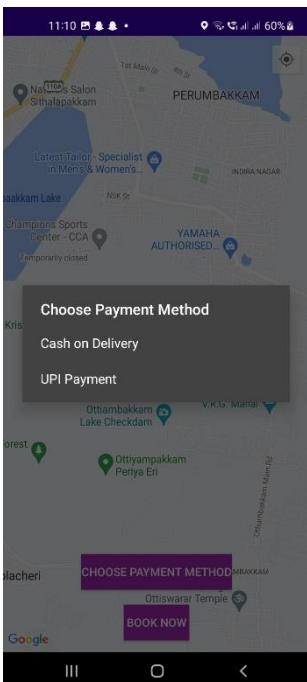
Confirm details



Check location

Buy Fertilizer clicked

Checkout button clicked



Payment method

On UPI Payment

Sending Booking details

# PLAGIARISM REPORT

## MAKE IT NATURAL: Application for Managing Food Waste and Excess Food and Plants in Homes: A Step to Minimize Food Wastage

S.T.Santhanalakshmi<sup>1</sup>, Dr. N. Pughazendi Narayanan, B. C. Dhevish, M.P. Dinesh Kumar and T. Lokesh Bharathwaj

*Department of Computer Science Panimdar Engineering college, Chennai, India*

santhalalakshmi.phd2020@gmail.com, pughazendi@gmail.com, iltdhevish@gmail.com, dineshmariappan12@gmail.com, mailtolokesh2272003@gmail.com

**Abstract**—An important goal in our world today is to eliminate food waste by reutilizing available food sources within local communities and leftover food items in homes, restaurants, stores and food distribution centres. An app is created for managing the food waste, excess food and plants at home by converting the food waste into natural fertilizer, providing the excess food to hunger- needy people or to nearby orphanage homes and the users can buy the natural fertilizer to use it on their plants. This app allows us to chat or book, the process can be done on own by chatting with an agent/associate or book an agent in this app, one of the agents collects the resources and do the further process for both food waste and excess food. There is an option the user to get the fertilizer that is made using the food waste from the user's house.

**Keywords**— App Development; Web Development; Food Waste Management; Hunger Alleviation;

### I. INTRODUCTION

#### A. Food Waste:

Food loss and waste encompass any food that goes uneaten. The reasons behind this wastage are multifaceted and span the entire food system, occurring during production, processing, distribution, retail, food service sales, and consumption. Globally, approximately one-third of the world's food ends up being discarded.

Categories of food waste than can be separated as:-

1. Kitchen waste
2. Dining waste
3. Non-edible/Spoiled food

These can be seen in homes, marriage halls, when a function takes place at home and any place where cooking or dining takes place.

Kitchen wastes include:-

1. Rotten vegetables and fruits
2. Peels of vegetables and fruits
3. Egg shell
4. Scraped portions of vegetables/fruits or slurries.
5. Non-veg waste before cooking (NOTE: HAS TO BE ADDED TO NON-EDIBLE)
6. Even bread waste



Fig. 1. Kitchen waste

Dinning waste includes

1. Inedible parts of a meal that we separate (both veg and non-veg).
2. Small amount of the leftovers in meal.



Fig. 2. Dining waste

Non-edible/Spoiled food includes: -

1. Spoiled cooked food
2. Non-veg waste before cooking
3. Non-veg waste after dining Spoiled dairy products, oil.



Fig.3. Non-edible/Spoiled food

#### B. Excess food:

Excess food refers to any surplus, unopened, and unexpired food designated for inclusion in a reimbursable meal that remains unused after a school has provided breakfast and lunch to students throughout the school day.

Leftovers denote the surplus food remaining unconsumed following a meal, which individuals may opt to store in containers for future consumption. Waste such as inedible bones is categorized separately from leftovers. Determining whether to keep or dispose of leftovers hinges

on several factors, such as the amount and type of food, as well as the specific situation. Excess food refers to any uneaten, unexpired, and unopened food or meal that cannot be used for a present or future meal provision.

**This includes:-**

1. Small number of leftovers at home.
2. Large number of leftovers in home functions, marriage halls and in any place where large amount of cooking or dining takes place.

## II. Related work

Recent research has delved into diverse approaches to address the multifaceted challenges of food waste management and donation using innovative technological solutions. Chhibber, Tripathi, and Ray [1] proposed a comprehensive framework integrating mobile application development and cloud-based supply chain management systems to virtualize food donation distribution processes. Their system aims to enhance the efficiency and reach of food donation initiatives, facilitating the seamless transfer of surplus food from donors to beneficiaries. In a complementary study, Ungureau-Comanit et al. [2] conducted a comprehensive investigation into the human health risks associated with various food waste management practices. Their findings underscored the critical importance of implementing proper disposal methods to mitigate health hazards and environmental contamination.

Building upon the foundational work in this field, Patil et al. [3] harnessed blockchain technology to design and implement a sustainable food waste management and tracking system. Leveraging the inherent transparency and security features of blockchain, their system enables stakeholders to monitor and trace food waste throughout the supply chain, thereby facilitating more informed decision-making and resource allocation. Meanwhile, Pandey and Patel [4] focused on developing an Android application that utilizes geographical location data to optimize the matching and routing of food donations. By leveraging real-time location information, their app enhances the efficiency and timeliness of food distribution efforts, ensuring that surplus food reaches those in need promptly.

In another innovative endeavor, Andres et al. [5] introduced FoodLifeSavr, a smartphone application designed to streamline the sharing of surplus food among individuals and communities. Employing user-friendly interfaces and social networking features, FoodLifeSavr fosters a sense of community engagement and cooperation in combating food waste. Similarly, Kavitha et al. [6] explored the potential of machine learning algorithms to automate the identification and donation of surplus food items. Their Replate app utilizes image recognition and classification algorithms to assess the quality and suitability of food donations, streamlining the donation process and minimizing logistical challenges.

Furthermore, Agyemang et al. [7] proposed novel techniques for improving food waste processing efficiency through water removal methods in university dining halls. By reducing moisture content in food waste, their approach

not only mitigates foul odors and bacterial growth but also facilitates downstream processing and recycling efforts. Jayalath et al. [8] contributed valuable insights through a detailed case study on waste minimization and performance enhancement in vegetable supply chains, highlighting the potential benefits of optimizing logistics and storage practices in reducing food waste at various stages of the supply chain.

In addition to these pioneering efforts, works such as FoodX by Shinta Oktaviana et al. [9, 13], Sudheepa et al.'s [10] food wastage management application, and Varghese et al.'s [11] food donation app SeVa have further enriched the discourse on mitigating food waste and facilitating donations through technological interventions. Together, these studies underscore the transformative potential of technology in addressing the complex challenges of food waste management and redistribution, offering promising avenues for enhancing sustainability, equity, and efficiency in food systems worldwide.

### A. Kitchen Waste Management

The majority of food waste typically ends up in either the trash can or garbage disposal, a habit that is inherently wasteful, particularly when considering the potential for these materials to be repurposed into something highly valuable: fertilizer. Food waste primarily consists of organic matter, which lends itself well to composting for the production of fertilizer. This method offers an efficient and environmentally friendly means of managing food waste within the confines of your kitchen.

Composting is when food and plant waste materials decomposes biologically. There are three main types of composting methods,

1. Aerobic (with oxygen)
2. Anaerobic (without oxygen)
3. Using vermicompost (using worms instead of bacteria).



Fig. 4. Composting

This process can be done within your garden by implementing these following steps:

#### 1. Add Kitchen Waste in a Container

Add kitchen waste like fruit peels, vegetables, eggshells, coffee or tea powder to a container (bucket, drum, terracotta pot) but don't fill the container completely rather add some small layer of these wastes. Add some soil over the kitchen wastes filled (Alluvial soil is preferable). After layering pour some water and repeat this process to several layers until the container is filled. For a continuous cycle maintain two containers so that if one is filled, you can continue the process

in the other.

#### 2. Add the Browns for better fertilization

To control moisture in your compost, include dry shredded leaves, sawdust, or cocopeat with the wet garbage. Use cardboard waste without stickers or tapes for faster fertilization. Additionally, accelerate composting by adding sawdust along with wood ash. Livestock manure can also be added for extra nutrients. If sawdust or manure is unavailable, they can be purchased from gardening supplies stores.

#### 3. Collect some garden waste

To create nutrient-rich fertilizer, collect grass clippings and leaves from your lawn while mowing. These organic materials are conveniently gathered in the lawnmower bag. Transfer the contents of the bag to your compost bin to enrich the compost with natural waste.

#### 4. Layering

Layer with soil and repeat the process for two more times, If the mix becomes soggy, add more browns to maintain the ratio.

#### 5. Provide Oxygen and water

In aerobic composting, providing oxygen is very crucial for a smell-free composting cycle and to avoid crawlies like maggots. Either you can make perforated holes in your container or churn the pile of waste once every four days. Pour water after composting for every 2 to 3 days. If the mix becomes soggy, add more browns to maintain the ratio.

#### 6. Apply to your garden

Wait for your compost to achieve a soil-like mixture that is dark in color. Once the compost reaches this appearance, it is ready to be spread. Use a garden fork to spread the compost on the garden fields you want to fertilize. Apply the right amount of compost and wait for the fertilizer to seep in and see some effect on the areas applied before you adding more. Aerobic composting will prepare manure in 40-45 days. All you have to do is keep adding kitchen waste daily and ensure oxygen flow.

Composting your kitchen waste offers several benefits which include getting rid of unwanted rubbish and having some fertilizer you can use on your lawn. Before throwing your leftovers and other food remnants, determine if they will work great as compost materials first.

#### B. Dining waste Management

Veg dining waste can be added to KITCHEN WASTE. That can be further processed to manure. Non-veg dining waste can be either given to some kind of pets or can be added to NON-EDIBLE.

#### C. Non-edible/ spoiled food Management

Sadly, these cannot be reused, maybe they also be composted but it is not advised because spoiled cooked rice/food, raw and cooked meat waste can contain harmful

bacteria, which could spread through your compost and later contaminate your crops. Oil and Dairy products like milk, sour cream, yogurt, cheese, and butter should not be composted because they also attract pests. So as these are completely waste, they have to move to the trash i.e. disposed properly.

#### D. Excess food Management



Fig. 5. Donating Food

Since the leftovers at home is very small amount, it can either be consumed afterwards or can be added to Spoiled food and further it will be We highly recommend you to consume afterwards.[4] For Large number of leftovers, those can be given to the hungry-needy people or orphanage homes.

Our Make it Natural app Book an agent to sell the food waste for certain amount or there is an option like manure is processed and given back to the user with certain amount charged. Book an agent to deliver the excess food to nearby homes.

### III. APP Development: DESIGN AND IMPLEMENTATION

#### A. Design of the Make it Natural App

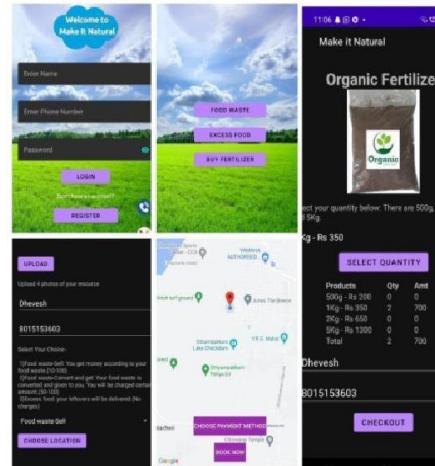


Fig. 6. App design

Food Donation: The Make it Natural app facilitates seamless food donation to homes or individuals in need nearby, operating consistently throughout the year. Kitchen Waste Management: A primary goal of "Make it

"Natural" is the efficient reuse of kitchen waste as nutrient-rich manure through composting. This proactive approach significantly reduces overall food waste, with a focus on vegetative scraps. Total Food Waste Management: "Make it Natural" tackles about 80% of total food waste management by repurposing kitchen waste and maximizing donation potential, while non-vegetarian waste poses a challenge, it remains committed to minimizing waste. Plants/crops management: Make it natural App sells organic fertilizer, which is prepared by composting the food waste, if user find it costly then there is an option like he/she can give their own food waste to convert and get , where less charges are applied.

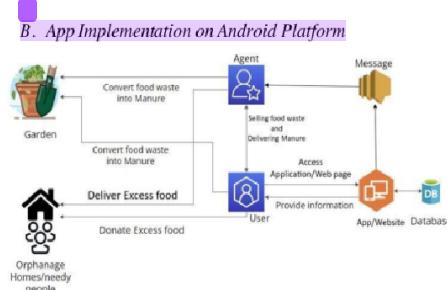


Fig. 7. Architecture of the App

#### Algorithm

1. Initialize the application/website interface.
2. Prompt the user to choose from available options:
  - a. Manage food waste
  - b. Manage excess food
  - c. Manage garden resources
  - d. Purchase organic fertilizer
  - e. Chat with an associate
  - f. Exit
3. If the user selects "Manage food waste":
  - a. Prompt the user to choose an action:
    - i. Book an agent for collection
    - ii. Get manure from processed food waste
  - b. If (i) is selected:
    - i. Book an agent for collection process
    - ii. Store user data in the database
    - iii. Alert the agent or send a message to them
    - iv. Agent arrives at the user's location to collect food waste
  - c. If (ii) is selected:
    - i. Provide information on the manure process
    - ii. Notify the user about the processing period
    - iii. Deliver processed manure to the user's location
4. If the user selects "Manage excess food":
  - a. Prompt the user to choose an action:
    - i. Locate nearby orphanage homes
    - ii. Book an agent for delivery
  - b. If (i) is selected:
    - i. Provide a map interface to locate orphanage homes

- c. If (ii) is selected:
  - i. Book an agent for food delivery to orphanages
  - ii. Store user data in the database
  - iii. Alert the agent or send a message to them
  - iv. Agent arrives at the user's location to collect excess food
  - v. Deliver excess food to nearby orphanage homes

5. If the user selects "Manage garden resources":
  - a. Provide options for garden resource management
  - b. Implement relevant functionalities based on user choice

6. If the user selects "Purchase organic fertilizer":
  - a. Display available organic fertilizer options
  - b. Enable user to make a purchase transaction

7. If the user selects "Chat with an associate":
  - a. Initiate a chat interface with a support associate
  - b. Provide assistance and guidance based on user queries

8. If the user selects "Exit":
  - a. Terminate the application/website interface

9. Repeat the process until the user chooses to exit.

The user uses the application/website to manage his food waste or excess food or garden. User can chat with an associate to get idea what to do with the food waste that he/she has or book an agent to sell the resources or can get it back as manure in few days. To manage the excess food user can visit nearby orphanage homes to donate the food by the maps or book an agent to do the job. Users can also buy Organic fertilizer using the app , and which is made naturally by the food wastes. The booking process, which uses the database to store the user's data and helps to book an agent by alerting him or send message to an agent. The agent role here is to arrive at the user's place collect the food waste, excess food or to deliver the organic fertilizer based on which purpose the user books. And the agent takes the food waste to do the manure process, takes the excess food to the nearby orphanage homes.

The implementation of the Make it Natural app is conducted using the Android Platform. The reasons for choosing Android are: (1) Android Studio, (2) Firebase console, (3) Google console, (4) Google services, (5) WhatsApp Business (WA Business), (6) Visual studio code and these provides various notable features that are highly creative and easy to use and all of them are stated here.

Android Studio is a free development suite for Android apps. It's the official integrated development environment (IDE) for Google's Android operating system. It's built on JetBrains' IntelliJ IDEA software and is available for download on Windows, macOS, and Linux based operating systems. Android Studio is a comprehensive integrated development environment (IDE) featuring a code editor, virtual Android emulator, and code templates. It provides tools for development, debugging, testing, and performance analysis. Android Studio can run on 4GB RAM, but it lags a bit. 8GB RAM is ideal for running Android Studio as it helps in

running their emulators.

XML (Extensible Markup Language) is a lightweight markup language used in Android Studio to implement UI-related data. It's a simple, scalable way to store and organize data. Using this language, the front end of the application is built. The front-end of an application, is the layer or element that the user has the ability to use, see, and interact with through buttons, images, interactive elements, navigational menus, and text.

Android Studio supports Java 11+ APIs without requiring a minimum API level for your app. This means that if you use an API introduced in Android 13, the code will also work on all previous versions. Some say that Java is the most suitable programming language for developing mobile apps because it allows easy multitasking and offers advanced exception-handling opportunities. Using this language, the back-end of the application is built.

Java is a popular choice for building Android app backends. Here are some benefits of using Java for Android app backends: Adaptability: Java's memory management system is versatile and allows for multi-threading. Security: Java is a secure language. Cross-platform use: Java allows for cross-platform use. Developer-friendly: Java is a simple, object-oriented language that's easy to learn. Frameworks: Java developers can use frameworks like Spring and Hibernate to build scalable and secure web applications.

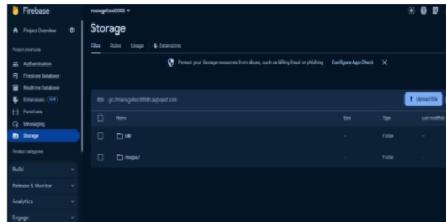


Fig. 8. Firebase storage

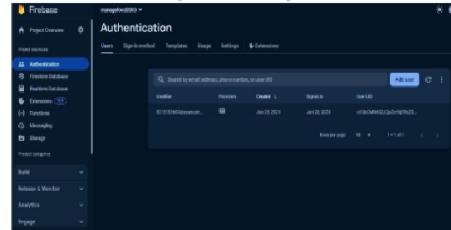


Fig. 9. Firebase Authentication

The Firebase console is a web-based interface that lets you manage your Firebase projects. Add and manage your Firebase projects. Configure Firebase features for your projects. View and manage your Firebase data. Monitor your Firebase apps. Database: This section lets you manage the Firebase Realtime Database for your project.. Storage: This section lets you manage the Firebase Cloud Storage for your project. Authentication: This section lets you manage the Firebase Authentication for your project. Using this Firebase Console as a Database for Storage and Authentication, it stores user's credentials, images and bills given by the user.

Google Search Console is a free service that helps website owners, developers, and SEO professionals understand how their site is performing in Google Search. You can also use the Maps JavaScript API to add a map to your website. This provides imagery and local data from the same source as Google Maps. You can also style the map, visualize your own data, and use services like geocoding and directions. Using this Google console, this application has access to map and finds current location.

Google Services refers to the programs, products, services, websites, documentation, and software offered by Google LLC and its affiliates. Google Mobile Services (GMS) is a collection of Google applications and APIs that work together to ensure a good user experience.

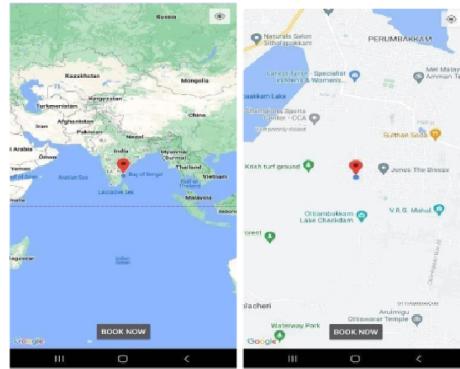


Fig.10. Google maps on app and current location

GMS includes networking, unified communications, and security. Google Services means Google's products and services that are accessible through and otherwise provided by various computer and electronic technologies, networks (syndicated and otherwise) and systems, including without limitation, mobile wireless services and Internet-based services accessible through the Google Sites and any Google syndication partner sites. For this Application, Gmail (Apply for agent job) and Google pay (Payment) are used.

WhatsApp Business is a free app for small businesses that allows them to communicate with customers through WhatsApp. It is available on Android and iPhone. WhatsApp Business includes all the features of WhatsApp Messenger, such as: Sending multimedia, Free calls, Free international messaging, Group chat, Offline messages. WhatsApp Business also provides tools to: Automate messages, Sort messages, Quickly respond to messages. A WhatsApp business account has been created for this application which attaches business email, business website and used for Customer-Administrator interaction and connect and contact Agent for the Collecting/Delivery service.

Visual Studio Code (VS Code) is a free, standalone source-code editor developed by Microsoft. It supports many programming languages, including Python, Java, C++, and JavaScript. VS Code offers a comprehensive suite of

development tools, including debugging, task running, and version control capabilities. Its primary aim is to furnish developers with everything necessary for a streamlined code-build-debug cycle. VS Code is a top pick for JavaScript and web developers, with extensions to support almost any programming language. HTML (Hyper Text Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables. Using this HTML, the webpage for the project is built. CSS stands for Cascading Style Sheets. It's a computer language that's used to structure and lay out web pages. CSS is a key technology of the World Wide Web, along with HTML and JavaScript. Using this CSS, the front-end of the webpage is built. JavaScript is a text-based programming language used for creating interactive web pages. Alongside HTML and CSS, it forms the core technologies of the World Wide Web, and it's essential for building both client- side and server-side functionality.

#### The Modules of the Application:-

1. Login Activity
2. Register Activity
3. Food waste
  - a. Make it yourself (Chat)
  - b. Book an agent
4. Excess food
  - a. Make it yourself(Maps)
  - b. Book an agent
5. Buy fertilizer
  - a. Select items >Location >Payment >Book
6. Book an agent
  - a. Selects photos>Type>location>Conform>Book

#### **IV. EVALUATION AND DISCUSSION**

##### **A. Environmental Implementation and Evaluation**

As the application is used for managing the food waste and excess food and sell organic fertilizers, the environment on which the application can be implemented are Home/House: Users can sell their Food waste or even convert and get it back as manure for their plants using the Application. Garden: To maintain a garden, to keep the plants healthy, users can buy Organic fertilizer from the Application. Agri-land: To make better and healthy crops/plants, users/farmers can buy Organic fertilizer from the Application. Cooking institutions: Teaching cooking can lead to more food waste; thus, they can be sold through the Application.

Mess, hotels, restaurants:

These places where several times cooking and dining takes place, and so the food waste and even excess food, both can be managed by the Application. Function halls: In such places large number of cooking and dining takes place, and for sure more food waste and the excess food, and by the application, the food waste can be sold and the Excess food can be given to nearby orphanage homes. Schools and colleges: Schools and colleges can contain canteen, which has the role cooking, and so the food waste can be managed by the Application.

#### **B. Discussion on App Evaluation**

User Registration and Login Functionality: The registration and login processes seem to be functioning correctly, as indicated by the successful execution of test cases. Both the "Register" and "Log in" buttons perform as expected, allowing users to register their credentials and subsequently log in to the system without any issues.

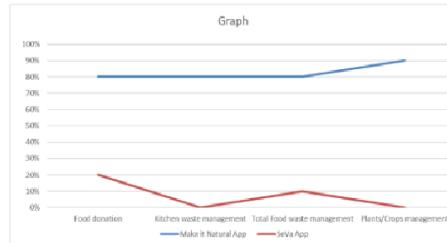


Fig. 11. Performance Parameter

**Food Waste and Excess Food Activities:** The "Food waste" and "Excess food" activities appear to function properly, presenting users with the option to either "Make it yourself" or "Book an agent." Test cases related to these activities demonstrate successful navigation and interaction, with users being prompted appropriately based on their selections.

**Booking Functionality:** The booking process, whether for making it yourself or booking an agent, seems to work smoothly according to the provided test cases. Actions such as uploading images, selecting quantities, and confirming bookings are successfully executed without any failures.

**Location-Based Functionality:** The system effectively retrieves and utilizes the user's current location, as demonstrated by the test cases related to selecting current locations and booking activities.

**Payment Method Integration:** The integration of payment methods, including Cash on Delivery and UPI, appears to be seamless, with users prompted to select their preferred payment method without encountering any issues.

**Overall System Reliability and Robustness:** The consistent passing of test cases across various functionalities suggests a high level of reliability and robustness in the system. Users can interact with different features and perform actions as intended, indicating a well-designed and implemented system.

**User Experience Considerations:** The autofill feature for user details enhances user experience by reducing manual input requirements, contributing to a smoother interaction flow. The use of increment and decrement buttons for quantity selection adds convenience and intuitiveness to the user interface.

**Data Management and Communication:** The successful uploading of booking details to the database and sending booking details to the manager indicates proper data management and communication protocols in place.

## V. CONCLUSIONS AND FUTURE WORK

### A. Conclusion

This Project conclude that most of the food wastes can be reused for providing good nutrition to plants as a natural fertilizer by composting them. But some should be disposed of in a proper way and the excess food especially the larger number of excess food can be given to the hungry-needy people or nearby orphanage homes. Thus, by using natural fertilizer to the plants, the plants give us healthy food. Therefore, this helps to be eco-friendly and to be good as humans with humanity by providing food to the needy people.

### B. Future Work

Build an agent app and send ride request to agent. Implementing this project can replace artificial manure as natural manure and major hunger can be reduced.

## REFERENCES

- [1] Divy Chhibber, Aditi Tripathi, Sandip Ray, "Do VIR: Virtualizing Food Donation Distribution through Mobile Application and Cloud-Based Supply Chain Management", 2021 IEEE International Conference on Consumer Electronics (ICCE).
- [2] Elena-Diana Ungureau-Comanit1, Ersilia Lazar Cosbuc1, Petronela Cozma1, Camelia Smaranda1, Maria Gavrilescu1,2, "Human Health Risks Concerning Food Waste Management", Grigore T. Popa University of Medicine and Pharmacy, Web Conference, Romania, October 29-30, 2020.
- [3] Swaraj Patil, Omkar Nikam, Suryakrishnan Nair, Aniket Raut, Vivian Brian Lobo, "Sustainable Food Waste Management and Tracking System Using Blockchain", 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT).
- [4] Arvind Kumar Pandey, Pratik Patel, "An Android Application Development for Food Donation: A Geographical Location Based Approach", 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).
- [5] Arfandi Andres, Edwin Setiawan Tanjaya, Ghaniyardi, Joshua Christian Sandhi, Harco Leslie Hendric Spits Warnars, "Sharing Food with FoodLifeSavr Smartphone App", 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS).
- [6] R. Kavitha, V Nuthan Prasad, Dankan Gowda V, Umayal M, Subhiksha K, Anitha K, "A Novel Approach for Identification and Donation of Surplus Food using Machine Learning-based Replate App", 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAIS).
- [7] Henry Agyemang, Sonith Riem, Samantha DiRenzo, John Ring, Jonathan T. Su, "Improved Food Waste Processing Through Water Removal in a University Dining Hall", 2023 Systems and Information Engineering Design Symposium (SIEDS).
- [8] Madushan Madhava Jayalath, R.M. Chandima Ratnayake, H. Niles Perera, Amila Thibbotuwawa, "Waste Minimization and Performance Improvement in Vegetable Supply Chains: A Case Study from a Developing Economy", 2023 8th International Conference on Business and Industrial.
- [9] Shinta Oktaviana R, Diana Ambarwati Febriani, Intan Yoshana, LR. Payanta, "FoodX, a System to Reduce Food Waste", 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE).
- [10] Katakam Sudheepa, Padmashetti Rashmitha, P. Sardar Maran, "Food Wastage Management Application using Android Studio", 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC).
- [11] Christina Varghese, Drashti Pathak, Aparna S. Varde, "SeVa : A Food Donation App for Smart Living", 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC).
- [12] Elena-Diana Ungureau-Comanit1, Ersilia Lazar Cosbuc1, Petronela Cozma1, Camelia Smaranda1, Maria Gavrilescu1, "Human Health Risks Concerning Food Waste Management".
- [13] Shinta Oktaviana R, Diana Ambarwati Febriani, Intan Yoshana, LR. Payanta, "FoodX, a System to Reduce Food Waste", 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE).

**20%**

SIMILARITY  
INDEX

**17%**

INTERNET  
SOURCES

**1%**

PUBLICATIONS

**2%**

STUDENT PAPERS

**1**

[www.thebetterindia.com](http://www.thebetterindia.com)

Internet Source

**4%**

**2**

[medium.com](http://medium.com)

Internet Source

**3%**

**3**

[www.ijirset.com](http://www.ijirset.com)

Internet Source

**2%**

**4**

[docplayer.net](http://docplayer.net)

Internet Source

**1%**

**5**

[restpublisher.com](http://restpublisher.com)

Internet Source

**1%**

**6**

[chaplin.expertlead.com](http://chaplin.expertlead.com)

Internet Source

**1%**

**7**

[developer.android.com](http://developer.android.com)

Internet Source

**1%**

**8**

Submitted to Purdue University

Student Paper

**1%**

**9**

Submitted to Debden Park High School

Student Paper

**1%**

10 fmudronebattery.shop  
Internet Source

&lt;1 %

11 www.coursehero.com  
Internet Source

&lt;1 %

12 ieeexplore.ieee.org  
Internet Source

&lt;1 %

13 visualstudio.microsoft.com  
Internet Source

&lt;1 %

14 androidsuperstar.net  
Internet Source

&lt;1 %

15 dissertations.mak.ac.ug  
Internet Source

&lt;1 %

16 gjjxczw.farmaciasantostefano.it  
Internet Source

&lt;1 %

17 habefast.ch  
Internet Source

&lt;1 %

18 Julián Alarte Aleixandre. "Information Retrieval Based on DOM Trees", Universitat Politecnica de Valencia, 2023  
Publication

&lt;1 %

19 Tanmay Thakur, Saurabh Kadam, Nikita Patil, Chinmayee Achrekar. "Machine Learning in Period, Fertility and Ovulation Tracking Application", Institute of Electrical and Electronics Engineers (IEEE), 2023  
Publication

&lt;1 %

21	Rashmi, V A Navyashree, S Sai Geetha, B K Pavan, C Deepti. "An IOTML Based Food Freshness Detection System", 2023 2nd International Conference for Innovation in Technology (INOCON), 2023 Publication	<1 %
22	dblp.dagstuhl.de Internet Source	<1 %
23	e-pres.di.uoa.gr Internet Source	<1 %
24	krishijagran.com Internet Source	<1 %
25	"Conversational Artificial Intelligence", Wiley, 2024 Publication	<1 %
26	Srinivasan Rajendran, D. Rajeswari. "CHAPTER 1 A Web-Based Platform to Reduce Food Wastage Through Women Organisation", IGI Global, 2024 Publication	<1 %
27	"Advances in Production Management Systems. Production Management Systems for Responsible Manufacturing, Service, and Logistics Futures", Springer Science and Business Media LLC, 2023 Publication	<1 %
28	Christina Varghese, Drashti Pathak, Aparna S. Varde. "SeVa: A Food Donation App for Smart Living", 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021 Publication	<1 %

# PAPER PUBLICATION

3/4/24, 5:50 PM

Gmail - Fwd: Acceptance of Paper ID # 310 for ICONIC 2K24 Presentation



ILT Dhevesh <iltdevesh@gmail.com>

## Fwd: Acceptance of Paper ID # 310 for ICONIC 2K24 Presentation

1 message

S.T.SANTHANALAKSHMI <santhanalakshmi.phd2020@gmail.com>  
To: iltdevesh@gmail.com

Mon, Mar 4, 2024 at 5:42 PM

----- Forwarded message -----

From: PECTEAM 2K24 <pecconference2k24@gmail.com>  
Date: Wed, Feb 21, 2024, 11:22 AM  
Subject: Acceptance of Paper ID # 310 for ICONIC 2K24 Presentation  
To: Lokesh Bharathwaj T <naitolokesh2272003@gmail.com>, christinak390@gmail.com <christinak390@gmail.com>, <pughazendi@gmail.com>, <santhanalakshmi.phd2020@gmail.com>

Dear Authors,

Congratulations on the acceptance of your paper ID # 310 titled " **MAKE IT NATURAL: Application for Managing Food Waste and Excess Food and Plants in Homes: A Step to Minimize Food Wastage**" for oral presentation at **ICONIC 2K24**. We appreciate your contribution to the conference. To proceed with the publication process, please carefully go through the attached reviewer comments and make necessary modifications to address the identified deficiencies in your paper. Ensure that the corrected version follows the CRP (Camera-Ready Paper) format provided in the websites.

### Submission Guidelines:

- Upload the CAMERA-READY version of your paper along with a "Response to Reviewer Comments" addressing all the comments received by the reviewers.
- Strictly adhere to the template provided on the website; no other styles are allowed.
- The plagiarism report is attached below. Maintain a similarity index of less than 15% and ensure there is no AI-generated content in the paper.
- Register for the conference before 28th Feb 2024, using the provided registration link below:  
[https://docs.google.com/forms/d/e/1FAIpQLScBClryzSulzkuAA2V4Xu8HKw53r\\_zREbWGiP5mttO34Q/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLScBClryzSulzkuAA2V4Xu8HKw53r_zREbWGiP5mttO34Q/viewform?usp=sf_link)
- For Camera Ready Paper (CRP) format, please visit <https://pecteam.co.in/>.

Please note that your registration becomes valid after your payment. View registration details and process at **7th INTERNATIONAL CONFERENCE on INTELLIGENT COMPUTING**(<https://pecteam.co.in/>)

<https://mail.google.com/mail/u/0/?ik=5f3b8e3bc&view=pt&search=all&permthid=thread-f:1792597682266746617&simpl=msg-f:1792597682266746617>

1/3

3/4/24, 5:50 PM

Gmail - Fwd: Acceptance of Paper ID # 310 for ICONIC 2K24 Presentation

### Details of the bank for registration:

Bank Name: UCO Bank  
Beneficiary Name: PEC-CONFERENCE AND RESEARCH  
Bank Account Number: 01570110103951  
Branch Name: Chetput, Chennai.  
IFSC code: UCBA0000157

Thank you for your cooperation, and we look forward to your final submission by the deadline.

### Reviewer Comment 1

1. The paper is related to the scope of the conference  
Yes
2. Does the title clearly reflect the content and outcomes in the manuscript?  
Yes
3. Research Design, Methods, Analysis of data, Interpretation of results, and conclusion are satisfactory  
Average
4. The organization of paper is satisfactory  
Yes
5. Do you recommend this paper?  
Yes
6. Overall Score  
Accept
7. Comments to Author  
Can add more references and surveys to dive into more depth of the topic.

Best regards,

### Prof.S.Vimala

Co-Convenor  
PECTEAM ICONIC 2K24  
Panimalar Engineering College  
Chennai-600 123  
Landline:7200191195  
Mobile:7395978385

paperregistration.pdf

<https://mail.google.com/mail/u/0/?ik=5f3b8e3bc&view=pt&search=all&permthid=thread-f:1792597682266746617&simpl=msg-f:1792597682266746617>

2/3

## REFERENCES

- [1] Divy Chhibber, Aditi Tripathi, Sandip Ray, "Do VIR: Virtualizing Food Donation Distribution through Mobile Application and Cloud-Based Supply Chain Management", 2021 IEEE International Conference on Consumer Electronics (ICCE).
- [2] Elena-Diana Ungureau-Comanit1, Ersilia Lazar Cosbuc1, Petronela Cozma1, Camelia Smaranda1, Maria Gavrilescu1,2, "Human Health Risks Concerning Food Waste Management", Grigore T. Popa University of Medicine and Pharmacy, Web Conference, Romania, October 29-30, 2020.
- [3] Swaraj Patil, Omkar Nikam, Suryakrishnan Nair, Aniket Raut, Vivian Brian Lobo, "Sustainable Food Waste Management and Tracking System Using Blockchain", 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT).
- [4] Arvind Kumar Pandey, Pratik Patel, "An Android Application Development for Food Donation: A Geographical Location Based Approach", 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).
- [5] Arfandi Andres, Edwin Setiawan Tanjaya, Ghaniyardi, Joshua Christian Sandhi, Harco Leslie Hendric Spits Warnars, "Sharing Food with FoodLifeSavr Smartphone App", 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS).
- [6] R. Kavitha, V Nuthan Prasad, Dankan Gowda V, Umayal M, Subhiksha K, Anitha K, "A Novel Approach for Identification and Donation of Surplus Food using Machine Learning-based Replate App", 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAIS).
- [7] Henry Agyemang, Sonith Riem, Samantha DiRenzo, John Ring, Jonathan T. Su, "Improved Food Waste Processing Through Water Removal in a University Dining Hall", 2023 Systems and Information Engineering Design Symposium (SIEDS).
- [8] Madushan Madhava Jayalath, R.M. Chandima Ratnayake, H. Niles Perera, Amila Thibbotuwawa, "Waste Minimization and Performance Improvement in Vegetable Supply Chains: A Case Study from a Developing Economy", 2023 8th International Conference on Business and Industrial.
- [9] Shinta Oktaviana R, Diana Ambarwati Febriani, Intan Yoshana, LR. Payanta, "FoodX, a System to Reduce Food Waste", 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE).
- [10] Katakan Sudheepa, Padmashetti Rashmitha, P. Sardar Maran, "Food Wastage Management Application using Android Studio", 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC).
- [11] Christina Varghese, Drashti Pathak, Aparna S. Varde, "SeVa: A Food Donation App for Smart Living", 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC).
- [12] Elena-Diana Ungureau-Comanit1, Ersilia Lazar Cosbuc1, Petronela Cozma1, Camelia Smaranda1, Maria Gavrilescu1,2, "Human Health Risks Concerning Food Waste Management".
- [13] Shinta Oktaviana R, Diana Ambarwati Febriani, Intan Yoshana, LR. Payanta, "FoodX, a System to Reduce Food Waste", 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE).
- [14] hsp.harvard, <https://www.hsp.harvard.edu/nutritionsource/sustainability/food-waste/>