**Mini Project Report**

Submitted to



**S.D.M COLLEGE (AUTONOMOUS),**

**Ujire – 574240**

**MINI PROJECT ON PASSWORD MANAGER APP**

By

Megha N

Roll No.: 221326

Under the guidance of

**Rakshitha H J**

Assistant Professor

**SRI DHARMASTHALA MANJUNATHESHWARA COLLEGE**

**(AUTONOMOUS),**

**Ujire – 574240**

**DEPARTMENT OF BVOC**

**IN**

**SOFTWARE AND APPLICATION DEVELOPMENT**

A logo with a circle and a banner

Description automatically generated with medium confidence

Certificate for the approval of project report

This is to certify that Megha N, 221326 of II B.Voc has satisfactorily completed the project

**“Mini Project Aahaar App”** for the IV Sem B.Voc in Software and

Application Development Degree

Prescribed by the college during the academic year 2023 – 2024.

**Head of the Department Project Guide Principal**

Examiners:

1.

2.

**DECLARATION**

I hereby declare that this project work entitled “Mini Project of Aahaar App” has been prepared by me during the year 2023 – 24 under the guidance of **Rakshitha H J**, Department of B.Voc (Software and Application Development), SDM College in the partial fulfilment of B.Voc Degree prescribed by the college.

I also declare that this project is the outcome of my own effort, that it has not been submitted to any other university for the award of any degree.

**Megha N**

**221326**

**ACKNOWLEGEMENT**

I would like to thank our principal Prof. Kumar Hegde, for his support. We also thank Department of B.Voc in Software and Application Development SDM College, Ujire for their valuable suggestions.

I would like to thank Mr. **Sammed Jain** HOD Department of B.Voc and **Rakshitha H J**, Assistant Professor Department of B.Voc for their help and for providing guidance in developing the project.

Megha N

**INDEX PAGE**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic Name / Chapter Name** | **Page No.** |
| 1 | Introduction |  |
| 2 | Requirement Specification |  |
| 3 | Module Description |  |
| 4 | Use Cases / DFD |  |
| 3 | Implementation |  |
| 4 | Result |  |
| 5 | Source Code |  |
| 6 | Conclusion and Scope for future work |  |
| 7 | Bibliography |  |

**Chapter 1**

**Introduction**

Welcome to Aahaar, In highly populated countries like India, food wastage is a disturbing issue. The streets, garbage bins and landfills have ample proof to prove it. Marriages, canteens, restaurants, social and family get-togethers and functions expel out so much food. Food wastage is not only an indication of hunger or pollution, but also of many economic problems. The high standard of living has resulted in the wastage of food, clothes, etc. because of quick changes in habits and lifestyle. Instead of wasting these things we can put them in use by donating them to various organizations such as orphanages, old age homes, etc. The product is an internet-based android application that basically aims at charity through donations.

The proposed application shall reduce food wastage and also fulfill other requirements like clothes, books, utensils, etc. of needy organizations and peoples.

As mentioned above in the description there is a lot of food wastage that occurs daily at restaurants and cafes. Instead of throwing away the same as trash (which usually is the scenario), it can be used to feed the homeless. Also, since the pickup is arranged for by the enterprise, the restaurants/cafes need not worry about it. Benefiters will be both the restaurants/cafés (reducing the carbon footprint and wastage), and the needy.

The ultimate objective of this project is to communicate that investments in food wastage reduction is the most logical step in the pursuit of sustainable production and consumption, including food security, climate change and other adverse environmental effects. Public awareness materials and a strategy will be developed to this effect.

**Chapter 2**

**Requirement Specification**

Requirements specification is a specification of software requirements and hardware requirements required to do the project. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating, and managing software or system requirements.

**2.1 Hardware Requirements**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Hardware / Equipment | Specification |
| 1 | Processor | - Multi-core processor (e.g., Snapdragon 400 series or MTK Dimensity series) for improved performance  - Ensure the processor is compatible with the operating system and software requirements |
| 2 | RAM | - Minimum 1GB or higher for better performance  - Recommended to use LPDDR4X or higher for faster memory speeds |

**2.2 Software Requirements**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Software / Feature | Specification |
| 1 | CPU Architecture | Universal |
| 2 | API Level | 24 or above |
| 3 | Operating System | Android, Android Emulator |
| 4 | SDK | Android |

The term "Processor" in the context of mobile devices refers to the central processing unit (CPU) of a smartphone or tablet, which is responsible for executing instructions and performing calculations. In the realm of mobile technology, processors play a crucial role in determining the device's performance, speed, and efficiency. Mobile processors are designed to handle various tasks such as running apps, browsing the internet, playing games, and multitasking.

RAM is a critical component that determines the performance and multitasking capabilities of a smartphone or tablet. It is the temporary, high-speed memory that stores the data and applications a device is currently using.

The amount of RAM in a mobile device has a significant impact on its overall performance. More RAM allows the device to keep more apps and data in memory, enabling smoother multitasking and faster app switching.

**Chapter 3**

**Modules Description**

The Android application development project described in the sources involves the utilization of highly secure Java modules, specifically the Java Cipher and AES (Advanced Encryption Standard) Encryption, to enhance the security of Android applications.

User Interface Setup:

The MainActivity class sets up the main user interface of the application using CardViews for various functionalities like donation, receiving, logging out, viewing food maps, etc.

Firebase Authentication:

The application uses Firebase Authentication for user authentication. The FirebaseAuth instance is initialized to check if a user is logged in. If not, it redirects the user to the landing page for authentication.

OnClickListener Setup:

Click listeners are set up for each CardView to handle user interaction. When a user clicks on a particular CardView, it triggers an intent to navigate to the corresponding activity.

Intent Handling:

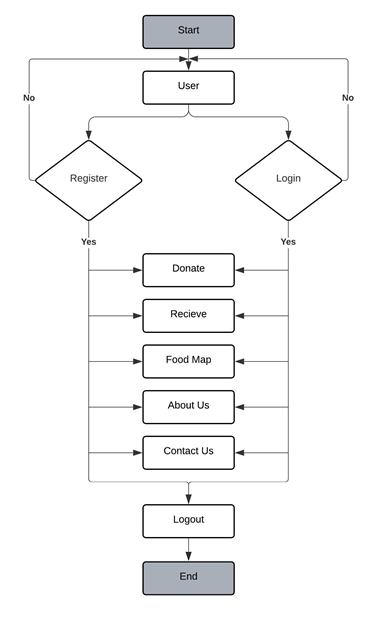
Intents are used to navigate between different activities within the application. Depending on which CardView the user clicks, an intent is created to launch the respective activity (e.g., Donate, Receive, FoodMap, etc.).

Logout Functionality:

The "Logout" CardView is provided for users to log out of their accounts. Clicking on this CardView triggers the signOut() method from Firebase Authentication, and the user is redirected to the landing page.

**Data Flow Diagram**

The data is generally stored in Android SharedPreferences using a different process diagrammatic representation of the storage process is as follows:

****

**Chapter 5**

**Implementation**

Implementation is a process of development of an application. Once the system design is completed then actual development of system will starts. The development of application using system design is called the implementation phase. In this phase, largest system is divided into small modules. For each module, algorithms are developed and each algorithm is coded using programming languages.

In this Android application project, the user interface (UI) is typically designed using XML (eXtensible Markup Language) files, which define the layout and appearance of the app's screens and components. XML provides a structured way to represent the UI elements, such as buttons, text views, and images, allowing developers to create visually appealing and responsive interfaces.

On the other hand, the coding backend of the application is often implemented using Java, a widely used programming language for Android development. Java is used to handle the logic, data processing, and interactions within the app, connecting the UI elements to the underlying functionality.

To store data persistently in Android applications, developers commonly utilize SharedPreferences, a key-value pair storage mechanism provided by the Android framework. SharedPreferences allow developers to store small amounts of data, such as user preferences, settings, or simple app state information, in a persistent manner across app sessions.

By combining XML for UI design, Java for backend logic, and SharedPreferences for data storage, developers can create efficient and user-friendly Android applications. This approach separates the presentation layer (XML) from the business logic (Java), promoting code modularity, reusability, and maintainability.

It also includes AES (Advanced Encryption Standard) encryption, which is commonly used to securely store sensitive data like passwords. By encrypting passwords before storing them, developers can enhance the security of user credentials within the app. AES encryption helps protect against unauthorized access and ensures that passwords are stored in an encrypted format, making it difficult for malicious actors to retrieve the original data. This encryption technique adds a layer of security to password storage, safeguarding user information from potential security breaches. Implementing AES encryption for password storage aligns with best practices for data security in Android applications, helping to maintain user trust and confidentiality.

Overall, this development approach leverages the strengths of each technology to create a cohesive and functional Android application, ensuring a smooth user experience and efficient data management.

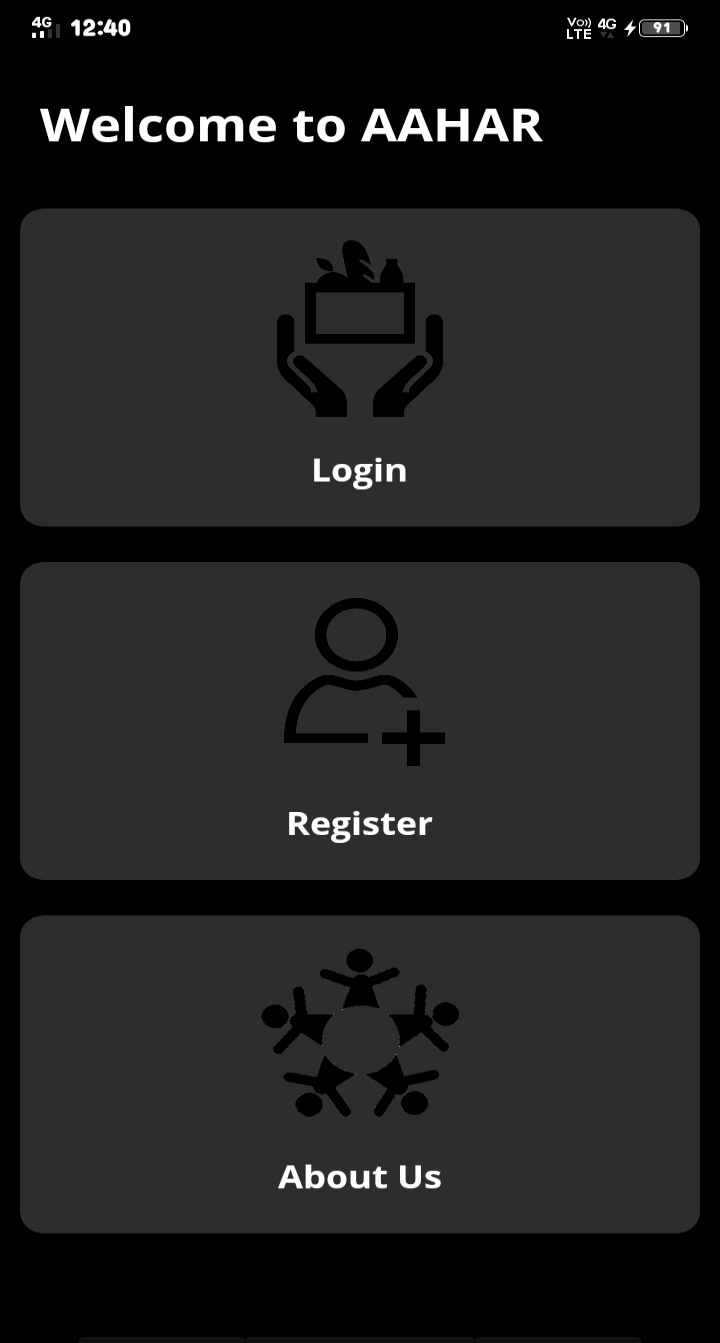
**Chapter 6**

**Results**

Results basically refer to any output that comes because of the completion of the activities that have been performed as part of the project or a particular project component. The provided screenshot appears to depict the various activities and functionalities of an Android application. Each activity, represented by a distinct screen or view, likely serves a specific purpose, and contributes to the overall user experience and functionality of the app.

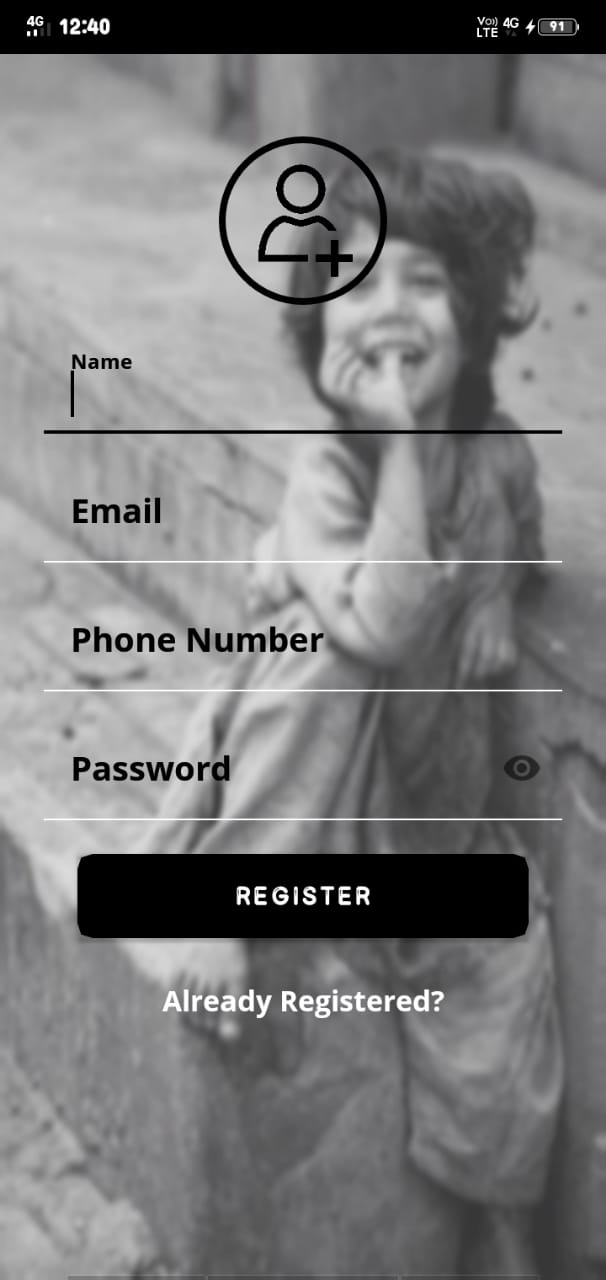


Fingerprint Authentication ActivityThis activity invokes the device in-built biometric sensor for Authentication.



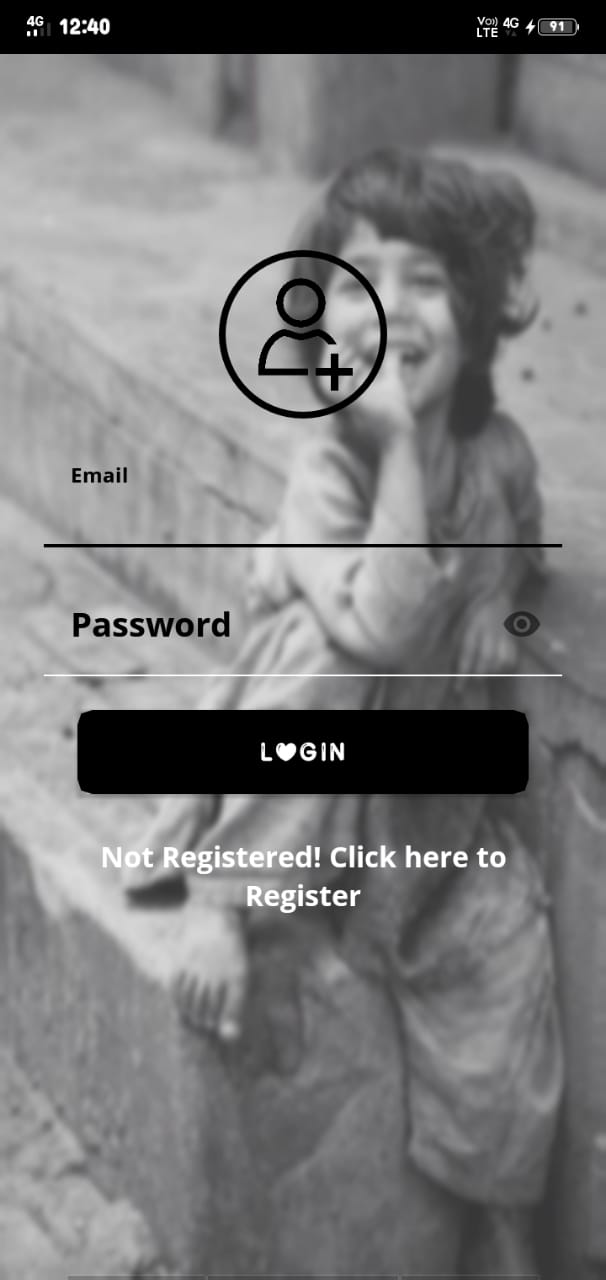
PIN-Setup Activity

This activity is used to store a PIN for Authentication of User Verification



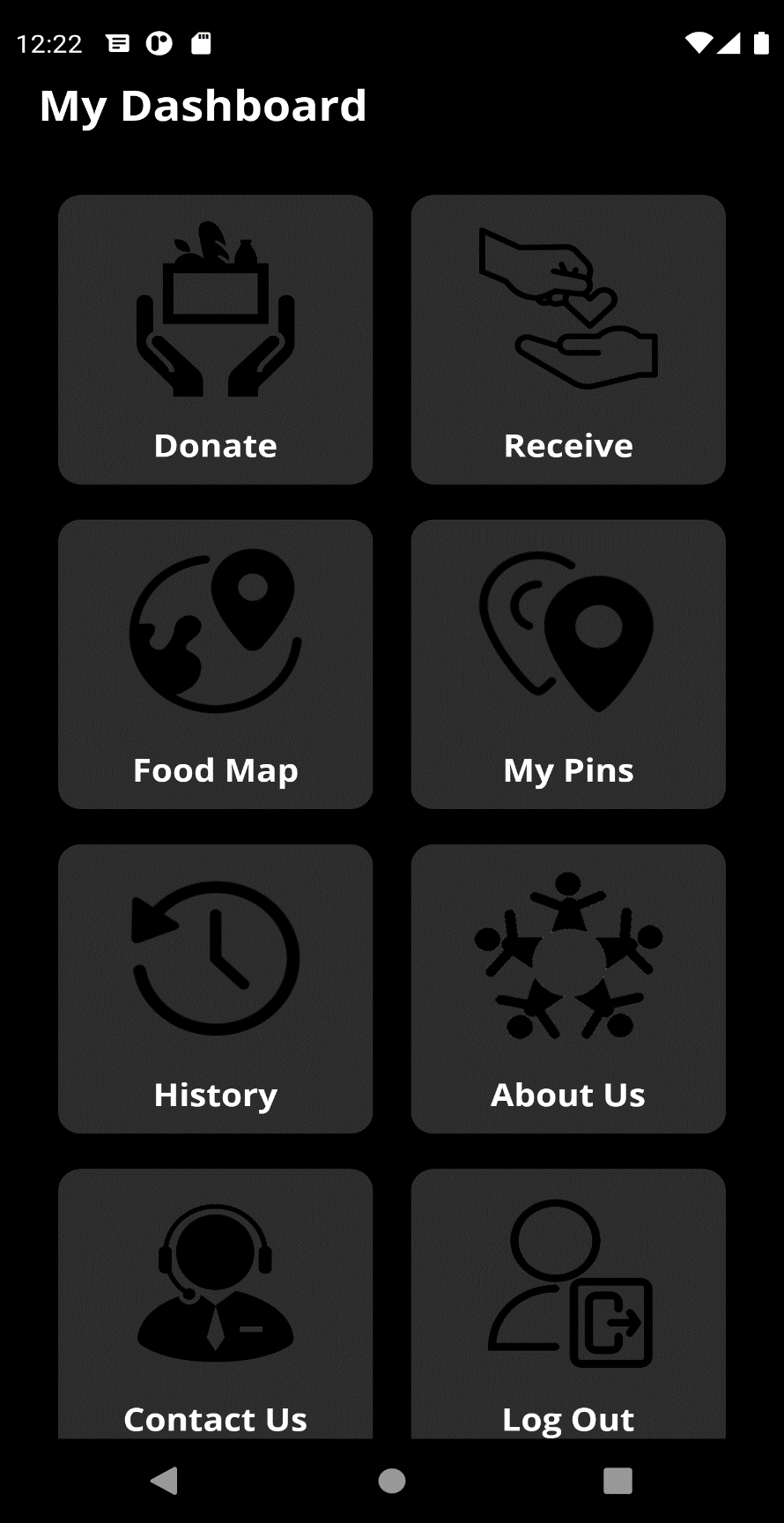
Vault Activity

This activity is used to store and display passwords and serves as homepage of the PassPod app.



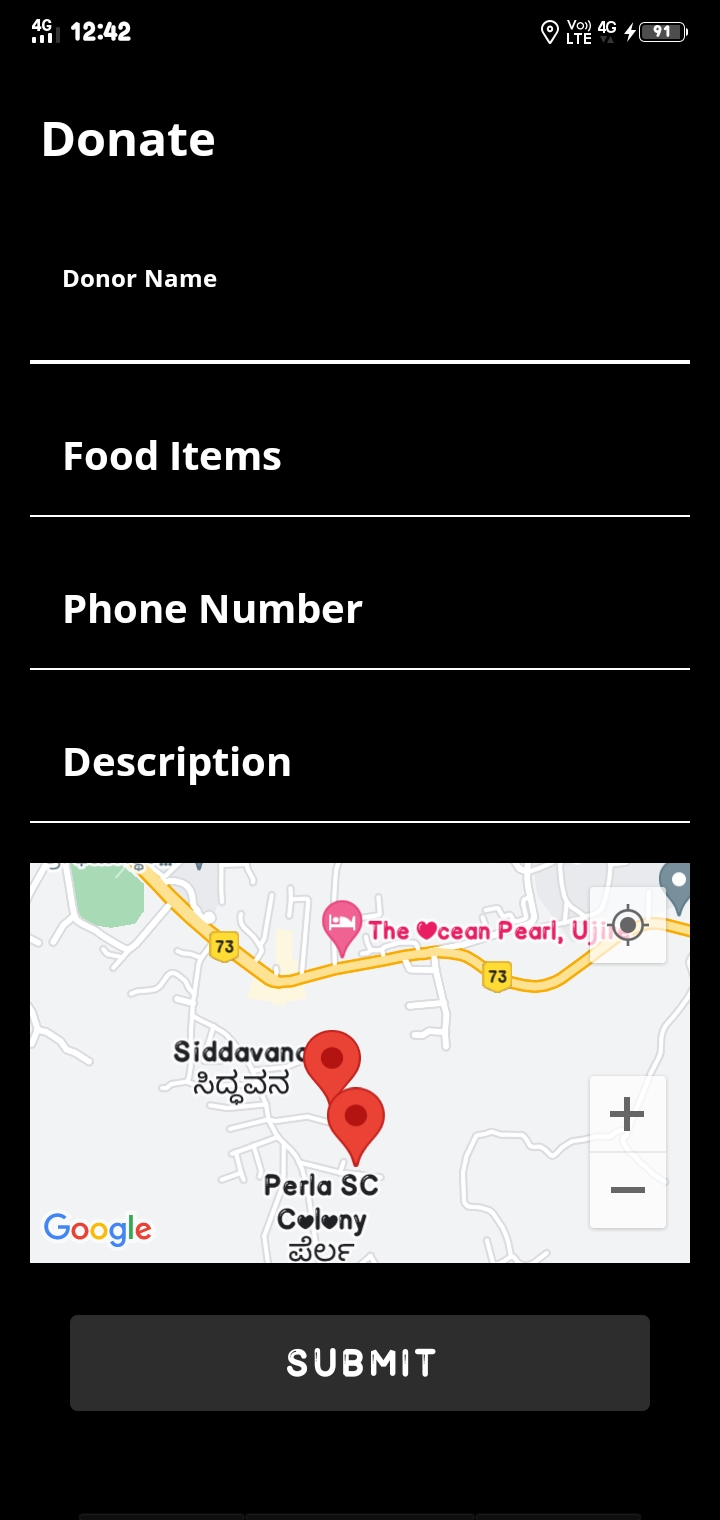
Password Store Fragment

This fragment is used to store the Password providing 3 entry fields.



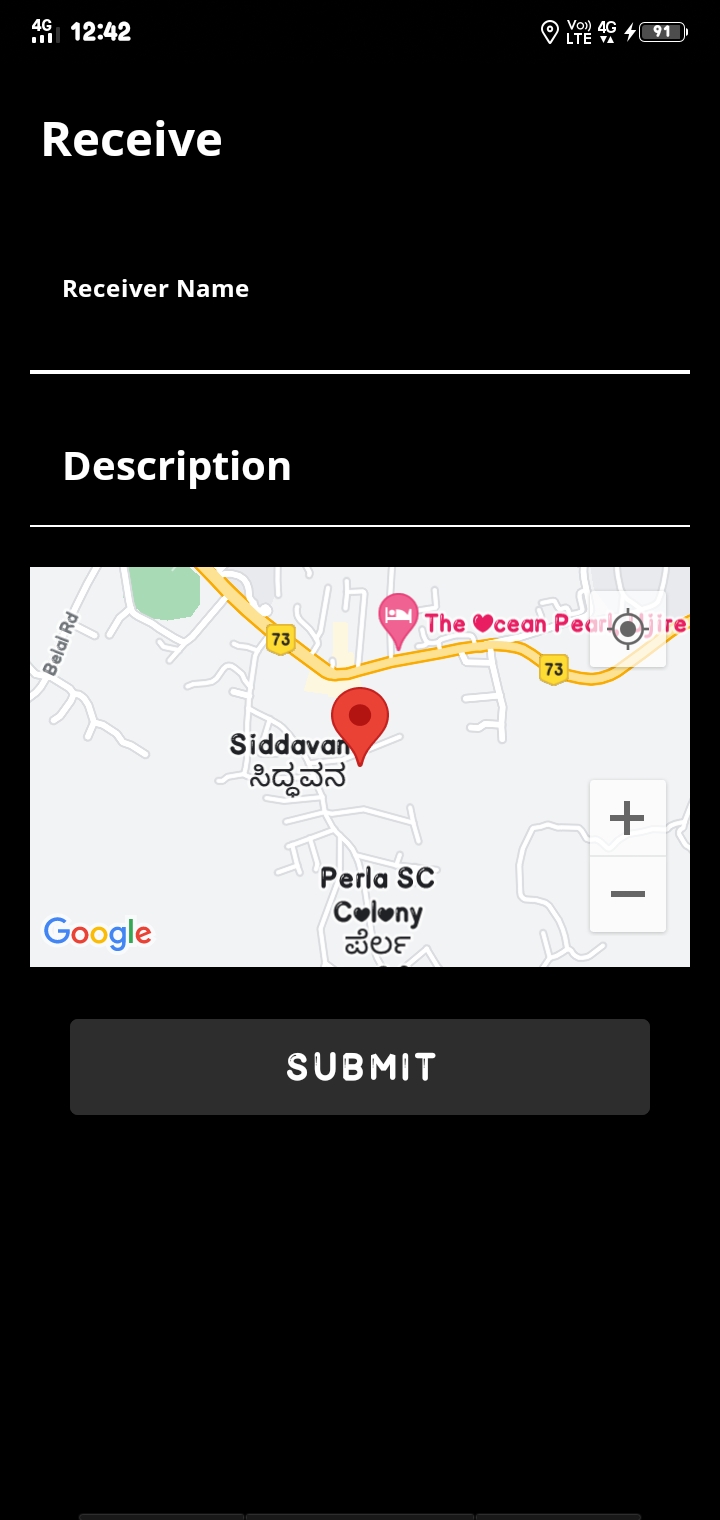
Credential Copy Action

The Copy Button invokes the Biometric. Prompt before copying the password to clipboard.



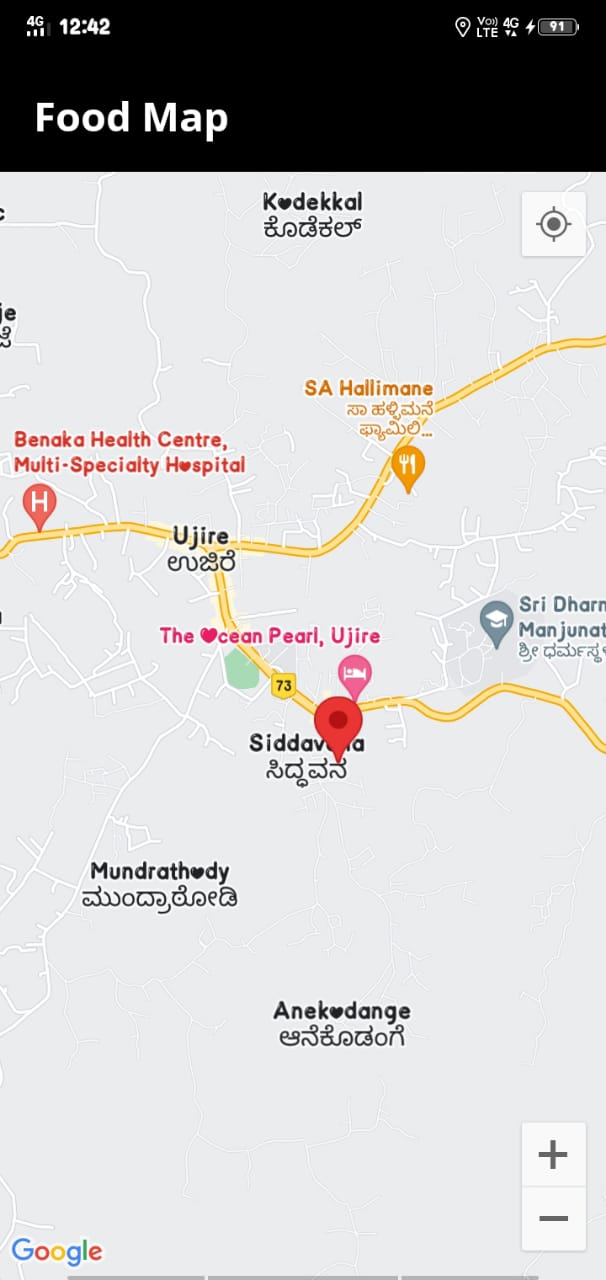
Password Update Fragment

This Fragment allows user to make changes in the saved password and update it.



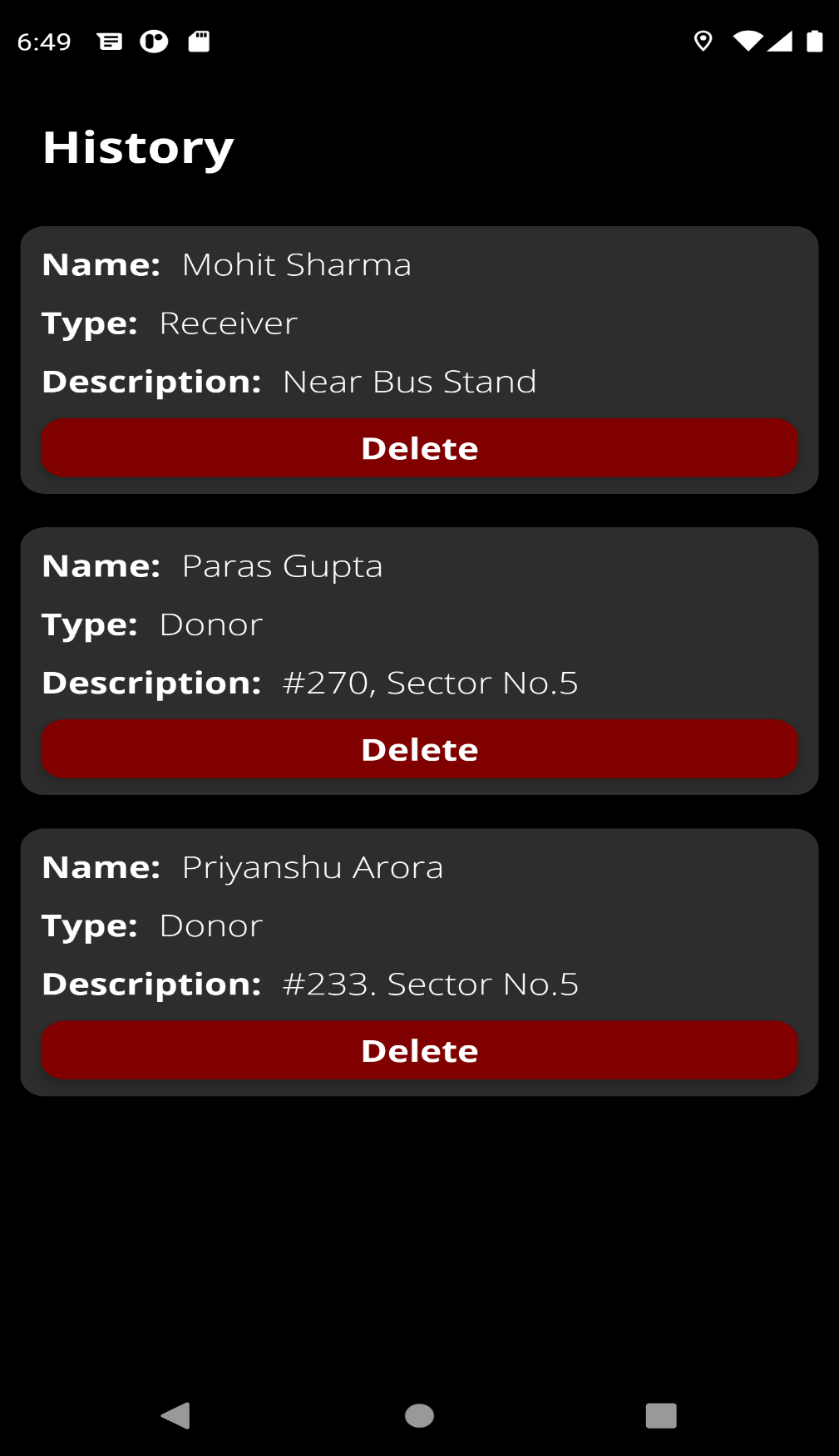
Vault Activity

This Page displays the multiple saved passwords.



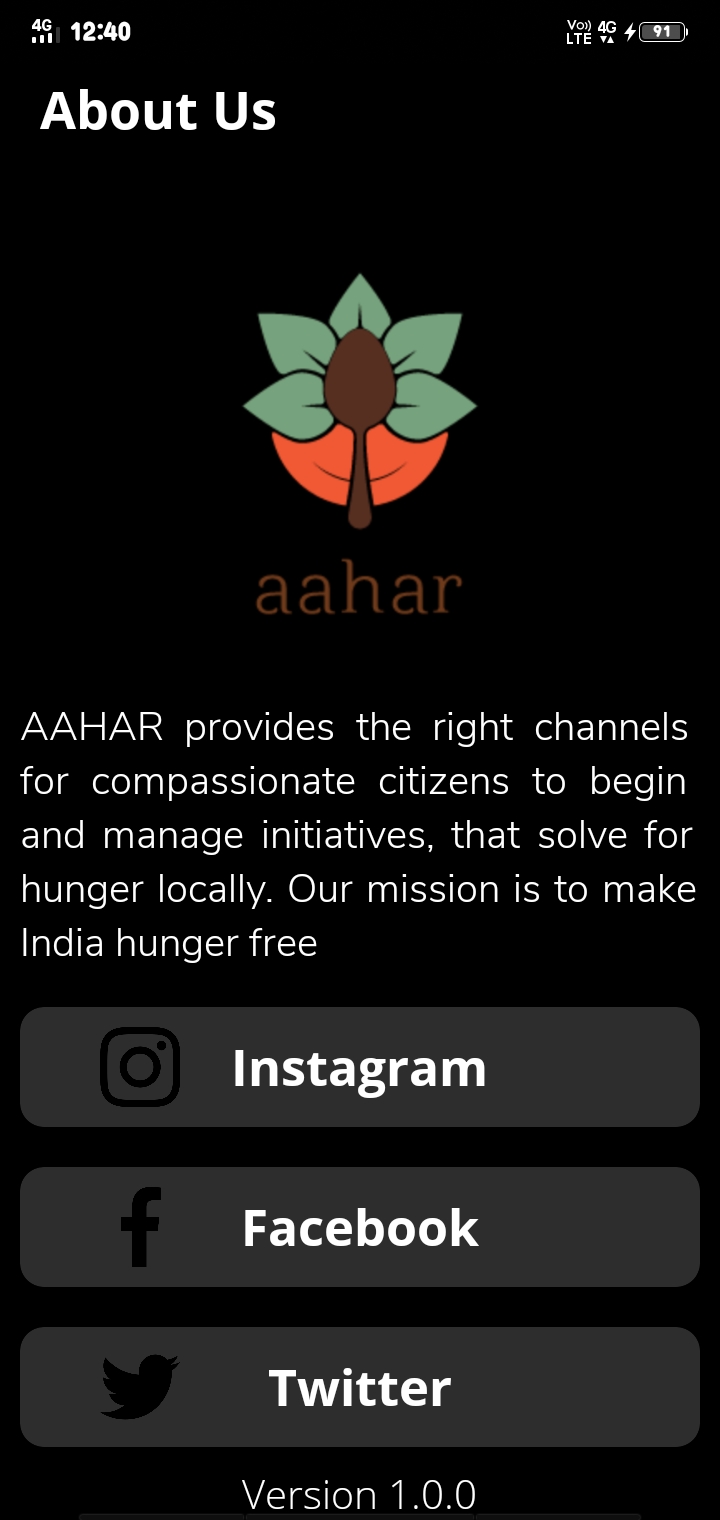
Vault Activity

This page represents multiple passwords stored using Recycler View.



Vault Activity

The Delete Button removes the stored password from the memory.



PIN - Verification Activity

This page verifies whether the stored PIN matches with the entered PIN



PIN – Verification Activity

If the entered PIN doesn’t match the stored PIN then the Submit Button gets disabled for a minute.

**Chapter 7**

**Source Code**

**activity\_landinpage.xml:**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

tools:context=".landingpage">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:ignore="ScrollViewSize">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/welcome"

android:fontFamily="@font/opensans"

android:textSize="26sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<androidx.cardview.widget.CardView

android:id="@+id/cardLogin"

android:clickable="true"

android:layout\_width="match\_parent"

android:layout\_height="180dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp"

tools:ignore="KeyboardInaccessibleWidget">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/donate"

android:layout\_marginTop="18dp"

android:layout\_gravity="center\_horizontal"

tools:ignore="ContentDescription" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/login"

android:fontFamily="@font/opensans"

android:textSize="18sp"

android:textColor="@color/white"

android:layout\_marginTop="135dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardRegister"

android:clickable="true"

android:layout\_width="match\_parent"

android:layout\_height="180dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp"

tools:ignore="KeyboardInaccessibleWidget">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/add"

android:layout\_marginTop="18dp"

tools:ignore="ContentDescription"

android:layout\_gravity="center\_horizontal"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/register"

android:fontFamily="@font/opensans"

android:textSize="18sp"

android:textColor="@color/white"

android:layout\_marginTop="135dp"

android:layout\_gravity="center\_horizontal" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardAboutus"

android:clickable="true"

android:layout\_width="match\_parent"

android:layout\_height="180dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp"

tools:ignore="KeyboardInaccessibleWidget">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/about"

android:layout\_marginTop="18dp"

tools:ignore="ContentDescription"

android:layout\_gravity="center\_horizontal"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/about"

android:fontFamily="@font/opensans"

android:textSize="18sp"

android:textColor="@color/white"

android:layout\_marginTop="135dp"

android:layout\_gravity="center\_horizontal" />

</androidx.cardview.widget.CardView>

</LinearLayout>

</ScrollView >

**activity\_signup.xml:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/bthre"

android:padding="16dp"

android:gravity="center"

tools:context=".Signup">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:gravity="center"

android:layout\_marginTop="-50dp"

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

tools:ignore="UselessParent">

<ImageView

android:id="@+id/profile"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@drawable/round\_background"

android:padding="15dp"

android:src="@drawable/add"

tools:ignore="ContentDescription" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/nameError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/name"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="textNoSuggestions"

android:background="@android:color/transparent"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:maxLines="1"

android:singleLine="true"

android:importantForAutofill="no"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/emailError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/prompt\_email"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="textEmailAddress"

android:background="@android:color/transparent"

android:maxLines="1"

android:singleLine="true"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:importantForAutofill="no"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/phoneError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/phone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/phone\_number"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:maxLength="13"

android:background="@android:color/transparent"

android:inputType="phone"

android:maxLines="1"

android:singleLine="true"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:importantForAutofill="no"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/passError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:passwordToggleEnabled="true"

android:layout\_marginTop="5dp"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/prompt\_password"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="textPassword"

android:background="@android:color/transparent"

android:maxLines="1"

android:singleLine="true"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:importantForAutofill="no"/>

</com.google.android.material.textfield.TextInputLayout>

<Button

android:id="@+id/register"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_margin="20dp"

android:background="@drawable/button\_rounded"

android:text="@string/register"

android:textColor="@android:color/white"

android:textSize="16sp"/>

<TextView

android:id="@+id/login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:layout\_marginBottom="5dp"

android:gravity="center"

android:text="@string/not\_register"

android:textColor="@color/white"

android:fontFamily="@font/opensans"

android:textSize="17sp"/>

</LinearLayout>

</LinearLayout>

**activity\_logup.xml:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/bthre"

android:padding="16dp"

android:gravity="center"

tools:context=".Logup">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:gravity="center"

android:layout\_marginTop="-50dp"

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

tools:ignore="UselessParent">

<ImageView

android:id="@+id/profile"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="@drawable/round\_background"

android:padding="15dp"

android:src="@drawable/add"

tools:ignore="ContentDescription" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/emailError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/prompt\_email"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:importantForAutofill="no"

android:inputType="textEmailAddress"

android:maxLines="1"

android:background="@android:color/transparent"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:singleLine="true"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/passError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

app:passwordToggleEnabled="true"

app:hintAnimationEnabled="true"

android:textColorHint="@color/black"

app:hintTextColor="@color/black"

app:boxStrokeColor="@color/black">

<EditText

android:id="@+id/password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/prompt\_password"

android:fontFamily="@font/opensans"

android:textSize="20sp"

android:importantForAutofill="no"

android:inputType="textPassword"

android:background="@android:color/transparent"

android:textColor="@color/black"

android:textColorHint="@color/black"

android:maxLines="1"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<Button

android:id="@+id/login"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_margin="20dp"

android:background="@drawable/button\_rounded"

android:text="@string/login"

android:textColor="@android:color/white"

android:textSize="16sp" />

<TextView

android:id="@+id/register"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:gravity="center"

android:text="@string/create\_new"

android:textColor="@color/white"

android:fontFamily="@font/opensans"

android:textSize="17sp" />

</LinearLayout>

</LinearLayout>

**activity\_main.xml:**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="My Dashboard"

android:fontFamily="@font/opensans"

android:textSize="24sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<GridLayout

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:rowCount="4"

android:columnCount="2"

android:layout\_margin="5dp"

android:alignmentMode="alignMargins"

android:layout\_gravity="center"

android:orientation="horizontal">

<androidx.cardview.widget.CardView

android:id="@+id/cardDonate"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/donate"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Donate"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardReceive"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/rcv"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Receive"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardFoodmap"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/map"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Food Map"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardMyPin"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/mypins"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="My Pins"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardHistory"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/history"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="History"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardAboutus"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/about"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="About Us"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardContact"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/contact"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Contact Us"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/cardLogout"

android:clickable="true"

android:layout\_width="165dp"

android:layout\_height="165dp"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

app:cardBackgroundColor="@color/gray"

android:layout\_margin="10dp">

<ImageView

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/lgout"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="15dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Log Out"

android:fontFamily="@font/opensans"

android:textSize="18dp"

android:textColor="@color/white"

android:layout\_marginTop="130dp"

android:layout\_gravity="center\_horizontal"/>

</androidx.cardview.widget.CardView>

</GridLayout>

</LinearLayout>

</ScrollView>

**acticty\_donate.xml**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

tools:context=".Donate">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/donate"

android:fontFamily="@font/opensans"

android:textSize="24sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center">

<androidx.cardview.widget.CardView

xmlns:Card\_View="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_margin="5dp"

android:padding="10dp"

app:cardBackgroundColor="@android:color/transparent"

Card\_View:cardCornerRadius="5dp"

Card\_View:cardElevation="0dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/nameError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/donorname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/donor\_name"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="text"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:backgroundTint="@android:color/transparent"

android:maxLines="1"

android:singleLine="true"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/itemError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/fooditem"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/food\_name"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:inputType="text"

android:backgroundTint="@android:color/transparent"

android:maxLines="2"

android:singleLine="true"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/phoneError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/phone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/phone\_number"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:backgroundTint="@android:color/transparent"

android:maxLength="12"

android:inputType="phone"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:maxLines="1"

android:singleLine="true"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

android:textSize="20sp"

android:fontFamily="@font/opensans"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/description"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/description"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="textMultiLine"

android:maxLines="10"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:backgroundTint="@android:color/transparent"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<fragment

android:id="@+id/google\_map"

xmlns:map="http://schemas.android.com/apk/res-auto"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_marginTop="20dp"

map:uiZoomControls="true"

tools:ignore="FragmentTagUsage" />

<Button

android:id="@+id/submit"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:layout\_margin="20dp"

android:backgroundTint="@color/button\_background\_color"

android:text="@string/submit"

android:textColor="@color/button\_text\_color"

android:textSize="20sp"

tools:ignore="UnusedAttribute" />

</LinearLayout>

</androidx.cardview.widget.CardView>

</RelativeLayout>

</LinearLayout>

</ScrollView>

**activity\_recieve.xml:**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

tools:context=".Receive">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Receive"

android:fontFamily="@font/opensans"

android:textSize="24sp"

android:textColor="@color/white"

android:layout\_margin="20dp"

tools:ignore="HardcodedText" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center">

<androidx.cardview.widget.CardView

xmlns:Card\_View="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

android:layout\_margin="5dp"

android:padding="10dp"

app:cardBackgroundColor="@android:color/transparent"

Card\_View:cardCornerRadius="5dp"

Card\_View:cardElevation="0dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/nameError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5dp"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/receivername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/receiver\_name"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="text"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:backgroundTint="@android:color/transparent"

android:maxLines="1"

android:singleLine="true"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/descriptionError"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:passwordToggleEnabled="true"

android:layout\_marginTop="5dp"

android:textColor="@color/white"

android:textColorHint="@color/white"

app:hintTextColor="@color/white"

android:textSize="20sp"

android:fontFamily="@font/opensans"

app:boxStrokeColor="@color/white">

<EditText

android:id="@+id/description"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:hint="@string/description"

android:textSize="20sp"

android:fontFamily="@font/opensans"

android:inputType="textMultiLine"

android:maxLines="10"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:backgroundTint="@android:color/transparent"

android:importantForAutofill="no"

tools:ignore="UnusedAttribute" />

</com.google.android.material.textfield.TextInputLayout>

<fragment

android:id="@+id/google\_map"

xmlns:map="http://schemas.android.com/apk/res-auto"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_marginTop="20dp"

map:uiZoomControls="true"

tools:ignore="FragmentTagUsage" />

<Button

android:id="@+id/submit"

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:layout\_margin="20dp"

android:backgroundTint="@color/button\_background\_color"

android:text="@string/submit"

android:textColor="@color/button\_text\_color"

android:textSize="20sp"

tools:ignore="UnusedAttribute" />

</LinearLayout>

</androidx.cardview.widget.CardView>

</RelativeLayout>

</LinearLayout>

</ScrollView>

activity\_food\_map.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"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

android:foregroundServiceType="location"

tools:context=".FoodMap">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/foodmap"

android:fontFamily="@font/opensans"

android:textSize="24sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<fragment

android:id="@+id/google\_map"

xmlns:map="http://schemas.android.com/apk/res-auto"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="70dp"

map:uiZoomControls="true"

/>

</RelativeLayout>

**activity\_my\_pin.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"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

android:foregroundServiceType="location"

tools:context=".MyPin">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/mypin"

android:fontFamily="@font/opensans"

android:textSize="24sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<fragment

android:id="@+id/google\_map"

xmlns:map="http://schemas.android.com/apk/res-auto"

android:name="com.google.android.gms.maps.SupportMapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="70dp"

map:uiZoomControls="true"

/>

</RelativeLayout>

**singlerow.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:clickable="true"

app:cardBackgroundColor="@color/gray"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

tools:ignore="KeyboardInaccessibleWidget">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:fontFamily="@font/opensans"

android:text="Name:"

android:textColor="@color/white"

android:textSize="18sp" />

<TextView

android:id="@+id/name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginRight="10dp"

android:fontFamily="@font/opensans\_light"

android:textColor="@color/white"

android:textSize="18sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:fontFamily="@font/opensans"

android:text="Type:"

android:textColor="@color/white"

android:textSize="18sp" />

<TextView

android:id="@+id/type"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginRight="10dp"

android:fontFamily="@font/opensans\_light"

android:textColor="@color/white"

android:textSize="18sp"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:fontFamily="@font/opensans"

android:text="Description:"

android:textColor="@color/white"

android:textSize="18sp" />

<TextView

android:id="@+id/description"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginRight="10dp"

android:fontFamily="@font/opensans\_light"

android:textColor="@color/white"

android:textSize="18sp" />

</LinearLayout>

<androidx.cardview.widget.CardView

android:id="@+id/delete"

android:layout\_width="match\_parent"

android:layout\_height="35dp"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:clickable="true"

app:cardBackgroundColor="@color/red"

app:cardCornerRadius="12dp"

app:cardElevation="5dp"

tools:ignore="KeyboardInaccessibleWidget">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:fontFamily="@font/opensans"

android:text="@string/delete"

android:textColor="@color/white"

android:textSize="18sp" />

</androidx.cardview.widget.CardView>

</LinearLayout>

</androidx.cardview.widget.CardView>

**userdata.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.appcompat.widget.LinearLayoutCompat

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

android:orientation="vertical"

tools:context=".UserdataActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/history"

android:fontFamily="@font/opensans"

android:textSize="26sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<androidx.recyclerview.widget.RecyclerView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/rec\_view"/>

</androidx.appcompat.widget.LinearLayoutCompat>

**activity\_history.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.appcompat.widget.LinearLayoutCompat

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/black"

android:orientation="vertical"

tools:context=".UserdataActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/history"

android:fontFamily="@font/opensans"

android:textSize="26sp"

android:textColor="@color/white"

android:layout\_margin="20dp"/>

<androidx.recyclerview.widget.RecyclerView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/rec\_view"/>

</androidx.appcompat.widget.LinearLayoutCompat>

**SplashScreen.java:**

package com.example.aahaarapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

public class SplashScreen extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

startActivity(new Intent(this,landingpage.class));

finish();

}

}

**MainActivity.java:**

package com.example.aahaarapp;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {

CardView donate, receive, logout, foodmap, about, contact, mypin, history;

FirebaseAuth fAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

donate = findViewById(R.id.cardDonate);

receive = findViewById(R.id.cardReceive);

logout = findViewById(R.id.cardLogout);

foodmap = findViewById(R.id.cardFoodmap);

mypin = findViewById(R.id.cardMyPin);

history = findViewById(R.id.cardHistory);

about = findViewById(R.id.cardAboutus);

contact = findViewById(R.id.cardContact);

fAuth= FirebaseAuth.getInstance();

if(fAuth.getCurrentUser() ==null){

Intent intent = new Intent(MainActivity.this, landingpage.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

donate.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), Donate.class);

startActivity(intent);

}

});

receive.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), Receive.class);

startActivity(intent);

}

});

foodmap.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), FoodMap.class);

startActivity(intent);

}

});

about.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), About.class);

startActivity(intent);

}

});

mypin.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), MyPin.class);

startActivity(intent);

}

});

history.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), UserdataActivity.class);

startActivity(intent);

}

});

contact.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent intent = new Intent(getApplicationContext(), Contact.class);

startActivity(intent);

}

});

logout.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

FirebaseAuth.getInstance().signOut();

Intent intent = new Intent(MainActivity.this, landingpage.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

});

}

}

**Signup.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;

import java.util.Map;

public class Signup extends AppCompatActivity {

public static final String TAG = "TAG";

EditText mFullName,mEmail,mPassword,mPhone;

Button mRegisterBtn;

TextView mLoginBtn;

FirebaseAuth fAuth;

FirebaseFirestore fStore;

String userID;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_signup);

mFullName = findViewById(R.id.name);

mEmail = findViewById(R.id.email);

mPassword = findViewById(R.id.password);

mPhone = findViewById(R.id.phone);

mRegisterBtn=findViewById(R.id.register);

mLoginBtn = findViewById(R.id.login);

fAuth=FirebaseAuth.getInstance();

fStore=FirebaseFirestore.getInstance();

if(fAuth.getCurrentUser() !=null){

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

//finish();

Intent intent = new Intent(Signup.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

mRegisterBtn.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v)

{

String email = mEmail.getText().toString().trim();

String password= mPassword.getText().toString().trim();

String name= mFullName.getText().toString().trim();

String phone= mPhone.getText().toString().trim();

if(TextUtils.isEmpty(email))

{

mEmail.setError("Email is Required.");

return;

}

if(TextUtils.isEmpty(password))

{

mPassword.setError("Password is Required.");

return;

}

if(password.length() < 6)

{

mPassword.setError("Password Must be >=6 Characters");

return;

}

fAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful()){

Toast.makeText(Signup.this, "User Created.", Toast.LENGTH\_SHORT) .show();

userID = fAuth.getCurrentUser().getUid();

DocumentReference documentReference = fStore.collection("users").document(userID);

Map<String,Object> user = new HashMap<>();

user.put("name",name);

user.put("email",email);

user.put("phone",phone);

documentReference.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Log.d(TAG,"onSuccess: user Profile is created for "+ userID);

Toast.makeText(Signup.this, "Registered Successfully.", Toast.LENGTH\_SHORT) .show();

}

});

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

Intent intent = new Intent(Signup.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

else{

Toast.makeText(Signup.this, "Error!" + task.getException().getMessage(),Toast.LENGTH\_SHORT).show();

}

}

});

}

});

mLoginBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), Logup.class));

}

});

}

}

**Logup.java:**

package com.example.aahaarapp;

import android.content.Intent;

import android.text.TextUtils;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import android.widget.EditText;

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 Logup extends AppCompatActivity {

EditText mEmail,mPassword;

Button mLoginBtn;

TextView mRegisterBtn;

FirebaseAuth fAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_logup);

mEmail = findViewById(R.id.email);

mPassword = findViewById(R.id.password);

mRegisterBtn = findViewById(R.id.register);

mLoginBtn = findViewById(R.id.login);

fAuth=FirebaseAuth.getInstance();

if(fAuth.getCurrentUser() !=null){

Intent intent = new Intent(Logup.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

mLoginBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String email = mEmail.getText().toString().trim();

String password= mPassword.getText().toString().trim();

if(TextUtils.isEmpty(email))

{

mEmail.setError("Email is Required.");

return;

}

if(TextUtils.isEmpty(password))

{

mPassword.setError("Password is Required.");

return;

}

if(password.length() < 6)

{

mPassword.setError("Password Must be >=6 Characters");

return;

}

//authenticate the user

fAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful()){

Toast.makeText(Logup.this, "Logged in Successfully.", Toast.LENGTH\_SHORT) .show();

Intent intent = new Intent(Logup.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}else{

Toast.makeText(Logup.this, "Error! " + task.getException().getMessage(),Toast.LENGTH\_SHORT).show();

}

}

});

}

});

mRegisterBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// redirect to RegisterActivity

Intent intent = new Intent(getApplicationContext(), Signup.class);

startActivity(intent);

}

});

}

}

**Donate.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.GeoPoint;

import com.google.firebase.firestore.ServerTimestamp;

import java.util.HashMap;

import java.util.Map;

public class Donate extends AppCompatActivity implements OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener, com.google.android.gms.location.LocationListener {

private GoogleMap mMap;

GoogleApiClient mGoogleApiClient;

Location mLastLocation;

LocationRequest mLocationRequest;

private int REQUEST\_CODE = 11;

SupportMapFragment mapFragment;

EditText mFullName,mFoodItem,mDescription,mPhone;

Button mSubmitBtn;

FirebaseAuth fAuth;

FirebaseFirestore fStore;

String userID;

public static final String TAG = "TAG";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_donate);

mFullName = findViewById(R.id.donorname);

mFoodItem = findViewById(R.id.fooditem);

mPhone = findViewById(R.id.phone);

mDescription = findViewById(R.id.description);

mSubmitBtn=findViewById(R.id.submit);

fAuth=FirebaseAuth.getInstance();

fStore= FirebaseFirestore.getInstance();

mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.google\_map);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

mapFragment.getMapAsync(this);

} else {

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_CODE);

}

}

@Override

public void onMapReady(GoogleMap googleMap) {

mMap = googleMap;

buildGoogleApiClient();

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

mMap.setMyLocationEnabled(true);

}

protected synchronized void buildGoogleApiClient(){

mGoogleApiClient = new GoogleApiClient.Builder(this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

mGoogleApiClient.connect();

}

@Override

public void onLocationChanged(@NonNull Location location) {

mLastLocation = location;

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

//MarkerOptions markerOptions1 = new MarkerOptions().position(latLng).title("You are here");

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

//mMap.animateCamera(CameraUpdateFactory.zoomTo(15));

//mMap.addMarker(markerOptions1).showInfoWindow();

MarkerOptions markerOptions = new MarkerOptions().position(latLng).title("You are here");

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,15));

mMap.addMarker(markerOptions).showInfoWindow();

mSubmitBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String fullname = mFullName.getText().toString().trim();

String fooditem= mFoodItem.getText().toString().trim();

String description= mDescription.getText().toString().trim();

String phone= mPhone.getText().toString().trim();

String type= "Donor";

if(TextUtils.isEmpty(fullname))

{

mFullName.setError("Name is Required.");

return;

}

if(TextUtils.isEmpty(fooditem))

{

mFoodItem.setError("Required.");

return;

}

if(phone.length() < 10)

{

mPhone.setError("Phone Number Must be >=10 Characters");

return;

}

userID = fAuth.getCurrentUser().getUid();

//DocumentReference documentReference = fStore.collection("donate").document(userID);

CollectionReference collectionReference = fStore.collection("user data");

GeoPoint geoPoint = new GeoPoint(location.getLatitude(),location.getLongitude());

Map<String,Object> user = new HashMap<>();

user.put("timestamp", FieldValue.serverTimestamp());

user.put("name",fullname);

user.put("food item",fooditem);

user.put("phone",phone);

user.put("description",description);

user.put("location",geoPoint);

user.put("userid",userID);

user.put("type",type);

collectionReference.add(user)

.addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH\_SHORT).show();

Log.d(TAG,"Success!");

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

Intent intent = new Intent(Donate.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(getApplicationContext(),"Error!",Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error!", e);

}

});

}

});

}

@Override

public void onConnected(@Nullable Bundle bundle) {

mLocationRequest = new LocationRequest();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);

}

@Override

public void onConnectionSuspended(int i) {

}

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

if (requestCode == REQUEST\_CODE){

if(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){

mapFragment.getMapAsync(this);

}else{

Toast.makeText(this,"Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

}

**Receive.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.GeoPoint;

import com.google.firebase.firestore.ServerTimestamp;

import java.util.HashMap;

import java.util.Map;

public class Receive extends AppCompatActivity implements OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener, com.google.android.gms.location.LocationListener {

private GoogleMap mMap;

GoogleApiClient mGoogleApiClient;

Location mLastLocation;

LocationRequest mLocationRequest;

private int REQUEST\_CODE = 11;

SupportMapFragment mapFragment;

EditText mFullName,mDescription;

Button mSubmitBtn;

FirebaseAuth fAuth;

FirebaseFirestore fStore;

String userID;

public static final String TAG = "TAG";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_receive);

mFullName = findViewById(R.id.receivername);

mDescription = findViewById(R.id.description);

mSubmitBtn=findViewById(R.id.submit);

fAuth=FirebaseAuth.getInstance();

fStore= FirebaseFirestore.getInstance();

mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.google\_map);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

mapFragment.getMapAsync(this);

} else {

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_CODE);

}

}

@Override

public void onMapReady(GoogleMap googleMap) {

mMap = googleMap;

buildGoogleApiClient();

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

mMap.setMyLocationEnabled(true);

}

protected synchronized void buildGoogleApiClient(){

mGoogleApiClient = new GoogleApiClient.Builder(this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

mGoogleApiClient.connect();

}

@Override

public void onLocationChanged(@NonNull Location location) {

mLastLocation = location;

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

//MarkerOptions markerOptions1 = new MarkerOptions().position(latLng).title("You are here");

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

//mMap.animateCamera(CameraUpdateFactory.zoomTo(15));

//mMap.addMarker(markerOptions1).showInfoWindow();

MarkerOptions markerOptions = new MarkerOptions().position(latLng).title("You are here");

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,15));

mMap.addMarker(markerOptions).showInfoWindow();

mSubmitBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String fullname = mFullName.getText().toString().trim();

String description= mDescription.getText().toString().trim();

String type= "Receiver";

if(TextUtils.isEmpty(fullname))

{

mFullName.setError("Name is Required.");

return;

}

if(TextUtils.isEmpty(description))

{

mFullName.setError("Description is Required.");

return;

}

userID = fAuth.getCurrentUser().getUid();

//DocumentReference documentReference = fStore.collection("receiver").document(userID);

CollectionReference collectionReference = fStore.collection("user data");

GeoPoint geoPoint = new GeoPoint(location.getLatitude(),location.getLongitude());

Map<String,Object> user = new HashMap<>();

user.put("timestamp", FieldValue.serverTimestamp());

user.put("name",fullname);

user.put("description",description);

user.put("location",geoPoint);

user.put("userid",userID);

user.put("type",type);

collectionReference.add(user)

.addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH\_SHORT).show();

Log.d(TAG,"Success!");

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

Intent intent = new Intent(Receive.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(getApplicationContext(),"Error!",Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error!", e);

}

});

}

});

}

@Override

public void onConnected(@Nullable Bundle bundle) {

mLocationRequest = new LocationRequest();

//mLocationRequest.setInterval(1000);

//mLocationRequest.setFastestInterval(1000);

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);

}

@Override

public void onConnectionSuspended(int i) {

}

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

if (requestCode == REQUEST\_CODE){

if(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){

mapFragment.getMapAsync(this);

}else{

Toast.makeText(this,"Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

}

**FoodMap.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.provider.ContactsContract;

import android.util.Log;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.Marker;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.EventListener;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.FirebaseFirestoreException;

import com.google.firebase.firestore.GeoPoint;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import static android.util.Base64.CRLF;

public class FoodMap extends AppCompatActivity implements OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener, com.google.android.gms.location.LocationListener {

private GoogleMap mMap;

GoogleApiClient mGoogleApiClient;

Location mLastLocation;

LocationRequest mLocationRequest;

SupportMapFragment mapFragment;

private int REQUEST\_CODE = 11;

FirebaseFirestore fStore;

public static final String TAG = "TAG";

private FirebaseFirestore cloudstorage;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_food\_map);

mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.google\_map);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

mapFragment.getMapAsync(this);

} else {

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_CODE);

}

}

@Override

public void onMapReady(GoogleMap googleMap) {

mMap = googleMap;

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

buildGoogleApiClient();

mMap.setMyLocationEnabled(true);

}

protected synchronized void buildGoogleApiClient(){

mGoogleApiClient = new GoogleApiClient.Builder(this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

mGoogleApiClient.connect();

}

@Override

public void onLocationChanged(@NonNull Location location) {

mLastLocation = location;

showLocation();

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

MarkerOptions markerOptions1 = new MarkerOptions().position(latLng).title("You are here").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_RED));

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

//mMap.animateCamera(CameraUpdateFactory.zoomTo(15));

//mMap.addMarker(markerOptions1).showInfoWindow();

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,15));

mMap.addMarker(markerOptions1).showInfoWindow();

}

public void showLocation() {

this.cloudstorage = FirebaseFirestore.getInstance();

cloudstorage.collection("user data")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

for (QueryDocumentSnapshot document : task.getResult()) {

Log.d(TAG, document.getId() + " => " + document.getData());//

if (document.contains("location") && document.contains("name") && document.contains("description")) {

GeoPoint location = (GeoPoint) document.get("location");

String title = (String) document.get("name");

String type = (String) document.get("type");

String description = (String) document.get("description");

if(type.equals("Donor")) {

Log.d(TAG, String.valueOf(location) + " Success " + title);

LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

mMap.addMarker(new MarkerOptions().position(latLng).title(title+"("+type+")").snippet(description).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_GREEN)));

}

else if(type.equals("Receiver")){

Log.d(TAG, String.valueOf(location) + " Success " + title);

LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

mMap.addMarker(new MarkerOptions().position(latLng).title(title+"("+type+")").snippet(description).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_BLUE)));

}

}

}

} else {

Log.d(TAG, "Error fetching data: ", task.getException());

}

}

});

}

@Override

public void onConnected(@Nullable Bundle bundle) {

mLocationRequest = new LocationRequest();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;+

}

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);

}

@Override

public void onConnectionSuspended(int i) {

}

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

if (requestCode == REQUEST\_CODE){

if(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){

mapFragment.getMapAsync(this);

}else{

Toast.makeText(this,"Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

}

**MyPin.java**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.util.Log;

import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.api.GoogleApiClient;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.BitmapDescriptorFactory;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.GeoPoint;

import com.google.firebase.firestore.QueryDocumentSnapshot;

import com.google.firebase.firestore.QuerySnapshot;

public class MyPin extends AppCompatActivity implements OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener, com.google.android.gms.location.LocationListener {

private GoogleMap mMap;

GoogleApiClient mGoogleApiClient;

Location mLastLocation;

LocationRequest mLocationRequest;

SupportMapFragment mapFragment;

private int REQUEST\_CODE = 11;

FirebaseFirestore fStore;

FirebaseAuth fAuth;

public static final String TAG = "TAG";

private FirebaseFirestore cloudstorage;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_my\_pin);

fAuth= FirebaseAuth.getInstance();

mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.google\_map);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

mapFragment.getMapAsync(this);

} else {

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_CODE);

}

}

@Override

public void onMapReady(GoogleMap googleMap) {

mMap = googleMap;

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

buildGoogleApiClient();

mMap.setMyLocationEnabled(true);

}

protected synchronized void buildGoogleApiClient(){

mGoogleApiClient = new GoogleApiClient.Builder(this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API)

.build();

mGoogleApiClient.connect();

}

@Override

public void onLocationChanged(@NonNull Location location) {

mLastLocation = location;

showLocation();

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

MarkerOptions markerOptions1 = new MarkerOptions().position(latLng).title("You are here").icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_RED));

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

//mMap.animateCamera(CameraUpdateFactory.zoomTo(15));

//mMap.addMarker(markerOptions1).showInfoWindow();

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,15));

mMap.addMarker(markerOptions1).showInfoWindow();

}

public void showLocation() {

this.cloudstorage = FirebaseFirestore.getInstance();

cloudstorage.collection("user data")

.get()

.addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {

@Override

public void onComplete(@NonNull Task<QuerySnapshot> task) {

if (task.isSuccessful()) {

for (QueryDocumentSnapshot document : task.getResult()) {

Log.d(TAG, document.getId() + " => " + document.getData());//

if (document.contains("location") && document.contains("name") && document.contains("description") && document.contains("userid")) {

GeoPoint location = (GeoPoint) document.get("location");

String title = (String) document.get("name");

String type = (String) document.get("type");

String description = (String) document.get("description");

String Userid = (String) document.get("userid");

String userID = fAuth.getCurrentUser().getUid();

if(type.equals("Donor") & Userid.equals(userID)) {

Log.d(TAG, userID + " Success " + title);

LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

mMap.addMarker(new MarkerOptions().position(latLng).title(title+"("+type+")").snippet(description).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_GREEN)));

}

else if(type.equals("Receiver") & Userid.equals(userID)){

Log.d(TAG, String.valueOf(location) + " Success " + title);

LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());

//mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

mMap.addMarker(new MarkerOptions().position(latLng).title(title+"("+type+")").snippet(description).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_BLUE)));

}

}

}

} else {

Log.d(TAG, "Error fetching data: ", task.getException());

}

}

});

}

@Override

public void onConnected(@Nullable Bundle bundle) {

mLocationRequest = new LocationRequest();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

return;

}

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);

}

@Override

public void onConnectionSuspended(int i) {

}

@Override

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

if (requestCode == REQUEST\_CODE){

if(grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED){

mapFragment.getMapAsync(this);

}else{

Toast.makeText(this,"Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

}

**landinpage.java:**

package com.example.aahaarapp;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import com.google.firebase.auth.FirebaseAuth;

public class landingpage extends AppCompatActivity {

CardView login,register,about;

FirebaseAuth fAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_landingpage);

login = findViewById(R.id.cardLogin);

register = findViewById(R.id.cardRegister);

about = findViewById(R.id.cardAboutus);

fAuth= FirebaseAuth.getInstance();

if(fAuth.getCurrentUser() !=null){

Intent intent = new Intent(landingpage.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

login.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), Logup.class));

}

});

register.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), Signup.class));

}

});

about.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

startActivity(new Intent(getApplicationContext(), About.class));

}

});

}

}

**About.java:**

package com.example.aahaarapp;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

public class About extends AppCompatActivity {

CardView instagram,facebook,twitter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about);

instagram = findViewById(R.id.instagram);

facebook = findViewById(R.id.facebook);

twitter = findViewById(R.id.twitter);

instagram.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent myWebLink = new Intent(android.content.Intent.ACTION\_VIEW);

myWebLink.setData(Uri.parse("http://www.instagram.com"));

startActivity(myWebLink);

}

});

facebook.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent myWebLink = new Intent(android.content.Intent.ACTION\_VIEW);

myWebLink.setData(Uri.parse("http://www.facebook.com"));

startActivity(myWebLink);

}

});

twitter.setOnClickListener(new View.OnClickListener ()

{

@Override

public void onClick(View v) {

Intent myWebLink = new Intent(android.content.Intent.ACTION\_VIEW);

myWebLink.setData(Uri.parse("http://www.twitter.com"));

startActivity(myWebLink);

}

});

}

}

**Contact.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.material.textfield.TextInputLayout;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.GeoPoint;

import java.util.HashMap;

import java.util.Map;

public class Contact extends AppCompatActivity {

EditText name, email, message;

Button submit;

boolean isNameValid, isEmailValid, isMessageValid;

FirebaseAuth fAuth;

FirebaseFirestore fStore;

String userID;

public static final String TAG = "TAG";

TextInputLayout nameError, emailError, messageError;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_contact);

name = (EditText) findViewById(R.id.name);

email = (EditText) findViewById(R.id.email);

message = (EditText) findViewById(R.id.message);

submit = (Button) findViewById(R.id.submit);

nameError = (TextInputLayout) findViewById(R.id.nameError);

emailError = (TextInputLayout) findViewById(R.id.emailError);

messageError = (TextInputLayout) findViewById(R.id.messageError);

fAuth=FirebaseAuth.getInstance();

fStore= FirebaseFirestore.getInstance();

submit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

SetValidation();

}

});

}

public void SetValidation() {

// Check for a valid name.

if (name.getText().toString().isEmpty()) {

nameError.setError(getResources().getString(R.string.name\_error));

isNameValid = false;

} else {

isNameValid = true;

nameError.setErrorEnabled(false);

}

// Check for a valid email address.

if (email.getText().toString().isEmpty()) {

emailError.setError(getResources().getString(R.string.email\_error));

isEmailValid = false;

} else if (!Patterns.EMAIL\_ADDRESS.matcher(email.getText().toString()).matches()) {

emailError.setError(getResources().getString(R.string.error\_invalid\_email));

isEmailValid = false;

} else {

isEmailValid = true;

emailError.setErrorEnabled(false);

}

// Check for a valid phone number.

if (message.getText().toString().isEmpty()) {

messageError.setError(getResources().getString(R.string.phone\_error));

isMessageValid = false;

} else {

isMessageValid = true;

messageError.setErrorEnabled(false);

}

if (isNameValid && isEmailValid && isMessageValid ) {

String Name = name.getText().toString().trim();

String Email= email.getText().toString().trim();

String Message= message.getText().toString().trim();

userID = fAuth.getCurrentUser().getUid();

//DocumentReference documentReference = fStore.collection("donate").document(userID);

CollectionReference collectionReference = fStore.collection("contact data");

Map<String,Object> user = new HashMap<>();

user.put("timestamp", FieldValue.serverTimestamp());

user.put("name",Name);

user.put("email",Email);

user.put("message",Message);

user.put("userid",userID);

collectionReference.add(user)

.addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH\_SHORT).show();

Log.d(TAG,"Successfully! We will shortly revert you back.");

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

Intent intent = new Intent(Contact.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(getApplicationContext(),"Error!",Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error!", e);

}

});

}

}

}

**History.java:**

package com.example.aahaarapp;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.util.Patterns;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.material.textfield.TextInputLayout;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.CollectionReference;

import com.google.firebase.firestore.DocumentReference;

import com.google.firebase.firestore.FieldValue;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.GeoPoint;

import java.util.HashMap;

import java.util.Map;

public class Contact extends AppCompatActivity {

EditText name, email, message;

Button submit;

boolean isNameValid, isEmailValid, isMessageValid;

FirebaseAuth fAuth;

FirebaseFirestore fStore;

String userID;

public static final String TAG = "TAG";

TextInputLayout nameError, emailError, messageError;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_contact);

name = (EditText) findViewById(R.id.name);

email = (EditText) findViewById(R.id.email);

message = (EditText) findViewById(R.id.message);

submit = (Button) findViewById(R.id.submit);

nameError = (TextInputLayout) findViewById(R.id.nameError);

emailError = (TextInputLayout) findViewById(R.id.emailError);

messageError = (TextInputLayout) findViewById(R.id.messageError);

fAuth=FirebaseAuth.getInstance();

fStore= FirebaseFirestore.getInstance();

submit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

SetValidation();

}

});

}

public void SetValidation() {

// Check for a valid name.

if (name.getText().toString().isEmpty()) {

nameError.setError(getResources().getString(R.string.name\_error));

isNameValid = false;

} else {

isNameValid = true;

nameError.setErrorEnabled(false);

}

// Check for a valid email address.

if (email.getText().toString().isEmpty()) {

emailError.setError(getResources().getString(R.string.email\_error));

isEmailValid = false;

} else if (!Patterns.EMAIL\_ADDRESS.matcher(email.getText().toString()).matches()) {

emailError.setError(getResources().getString(R.string.error\_invalid\_email));

isEmailValid = false;

} else {

isEmailValid = true;

emailError.setErrorEnabled(false);

}

// Check for a valid phone number.

if (message.getText().toString().isEmpty()) {

messageError.setError(getResources().getString(R.string.phone\_error));

isMessageValid = false;

} else {

isMessageValid = true;

messageError.setErrorEnabled(false);

}

if (isNameValid && isEmailValid && isMessageValid ) {

String Name = name.getText().toString().trim();

String Email= email.getText().toString().trim();

String Message= message.getText().toString().trim();

userID = fAuth.getCurrentUser().getUid();

//DocumentReference documentReference = fStore.collection("donate").document(userID);

CollectionReference collectionReference = fStore.collection("contact data");

Map<String,Object> user = new HashMap<>();

user.put("timestamp", FieldValue.serverTimestamp());

user.put("name",Name);

user.put("email",Email);

user.put("message",Message);

user.put("userid",userID);

collectionReference.add(user)

.addOnSuccessListener(new OnSuccessListener<DocumentReference>() {

@Override

public void onSuccess(DocumentReference documentReference) {

Toast.makeText(getApplicationContext(),"Success!",Toast.LENGTH\_SHORT).show();

Log.d(TAG,"Successfully! We will shortly revert you back.");

//startActivity(new Intent(getApplicationContext(),MainActivity.class));

Intent intent = new Intent(Contact.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(getApplicationContext(),"Error!",Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error!", e);

}

});

}

}

}

**model.java:**

package com.example.aahaarapp;

public class model {

String name,type,description,userid;

public model() {

}

public model(String name, String type, String description, String userid) {

this.name = name;

this.type = type;

this.description = description;

this.userid = userid;

}

public String getUserid() {

return userid;

}

public void setUserid(String userid) {

this.userid = userid;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

**myadapter.java:**

package com.example.aahaarapp;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.auth.FirebaseAuth;

import java.util.ArrayList;

public class myadapter extends RecyclerView.Adapter<myadapter.myviewholder>

{

ArrayList<model> datalist;

FirebaseAuth fAuth= FirebaseAuth.getInstance();

public String userID = fAuth.getCurrentUser().getUid();

public String uid;

public myadapter(ArrayList<model> datalist) {

this.datalist = datalist;

}

@NonNull

@Override

public myviewholder onCreateViewHolder (@NonNull ViewGroup parent,int viewType){

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.singlerow, parent, false);

return new myviewholder(view);

}

@Override

public void onBindViewHolder(@NonNull myviewholder holder, int position) {

holder.tname.setText(datalist.get(position).getName());

holder.ttype.setText(datalist.get(position).getType());

holder.tdescription.setText(datalist.get(position).getDescription());

}

public void deleteItem(int position){

//getSnapshots().getSnapshot(position).getReference().delete();

//notifyDataSetChanged();

}

@Override

public int getItemCount() {

return datalist.size();

}

class myviewholder extends RecyclerView.ViewHolder

{

TextView tname,ttype,tdescription;

public myviewholder(@NonNull View itemView) {

super(itemView);

tname = itemView.findViewById(R.id.name);

ttype = itemView.findViewById(R.id.type);

tdescription = itemView.findViewById(R.id.description);

}

}

}

UserdataActivity.java:

package com.example.aahaarapp;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.DocumentSnapshot;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.Query;

import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;

import java.util.List;

public class UserdataActivity extends AppCompatActivity {

RecyclerView recyclerView;

ArrayList<model> datalist;

FirebaseFirestore db;

myadapter adapter;

FirebaseAuth fAuth= FirebaseAuth.getInstance();

public String userID = fAuth.getCurrentUser().getUid();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.userdata);

recyclerView=(RecyclerView)findViewById(R.id.rec\_view);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

datalist=new ArrayList<>();

adapter=new myadapter(datalist);

recyclerView.setAdapter(adapter);

db=FirebaseFirestore.getInstance();

db.collection("user data").orderBy("timestamp", Query.Direction.DESCENDING).get()

.addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {

@Override

public void onSuccess(QuerySnapshot queryDocumentSnapshots) {

List<DocumentSnapshot> list=queryDocumentSnapshots.getDocuments();

for(DocumentSnapshot d:list)

{

model obj=d.toObject(model.class);

//datalist.add(obj);

String Userid = (String) d.get("userid");

if(Userid.equals(userID)) {

datalist.add(obj);

}

}

adapter.notifyDataSetChanged();

}

});

}

}

**Chapter 8**

**Conclusions and scope for future work**

User Profile Management:

Implement a module to allow users to create and manage their profiles within the app. Users should be able to update their personal information, view their donation/receipt history, and track their contributions over time.

Notification System:

Integrate a notification system to alert users about important events such as donation pickups, delivery confirmations, and new messages. This enhances user engagement and keeps them informed about the status of their donations.

Real-Time Tracking:

Enhance the logistics module to include real-time tracking of donation pickups and deliveries. This provides transparency to donors and recipients by allowing them to track the location and status of their donations in real-time.

Social Sharing:

Implement social sharing functionality to allow users to share their donation activities on social media platforms. This not only promotes the app but also encourages others to participate in charitable activities.

Volunteer Management:

Create a module for users to sign up as volunteers to assist with donation pickups, deliveries, and other tasks. Volunteers can indicate their availability and preferences, and the app can match them with relevant opportunities.Feedback and Rating

System:

Enhance the feedback and rating system to allow users to provide detailed feedback on their donation/receipt experience. This helps improve the quality of service and fosters trust among users.

Localization and Multi-language Support:

Implement localization features to support multiple languages and cater to users from different regions. This improves accessibility and usability for a diverse user base.

**Chapter 9**

**Bibliography**

1. <https://developer.android.com/guide>

2. <https://android.stackexchange.com/>

3. <https://lottiefiles.com/>

4. <https://fonts.google.com/>

5. <https://m3.material.io/develop/android/mdc-android>