WORK AT YOUR WILL

A

Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

M. AMRUTH SAI(1602-19-737-066)

K. BHARATH (1602-19-737-069)

B. JOSEPH(1602-19-737-074)

Under the guidance of

KEZIA RANI

HASEEBA

MUKESH TRIPATHI

Professor

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to OsmaniaUniversity)IbrahimBagh,

Hyderabad-31

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

DECLARATION BY THE CANDIDATES

We, **M. AMRUTH SAI, K. BHARATH, B. JOSEPH** bearing hall ticket number, **1602-19-737-066, 1602-19-737-069, 1602-19-737-074** hereby declare that the project report entitled **“WORK AT YOUR WILL”** under the guidance of **KEZIA RANI (**Professor), Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of MINI PROJECT of VI semester of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

M. AMRUTH SAI (1602-19-737-066)

K. BHARATH (1602-19-737-069)

B. JOSEPH (1602-19-737-07

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled “**WORK AT YOUR WILL**” being submitted by **M. AMRUTH SAI**, **K. BHARATH**, **B. JOSEPH** bearing hall ticket numbers 1602-19-737-066, 1602-19-737-069, 1602-19-737-074 in partial fulfilment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under guidance.

DR. K. Ram Mohan Rao
HOD, IT

KEZIA RANI

HASEEBA

MUKESH TRIPATHI

Professor

Internal Guide

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to **KEZIA RANI (Professor)** under whom we executed this project. We would also use this opportunity to thank our Head of Department **Dr. K. Ram Mohan Rao**. We would also like to thank all faculty members and staff of the Department of Information Technology for their generous help in various ways for the completion of this project.

Finally, we would like to express our heartfelt thanks to our senior
M W Farooqui Junaid (1602-17-737-020).

We are grateful to his guidance, and constructive suggestions that helped us in the preparation

of this project. His constant guidance and willingness to share his vast knowledge made us

understand this project and its manifestations in great depths and helped us to complete the

assigned tasks.

ABSTRACT

Our Work India application is a one-stop solution for all the recruitment needs aims at creating a system which allows a user to finish their work in a easy way and acts as a perfect job search portal for all the skill-full workers who are driven and best at providing their services considering user's comfort as their priority. Our application creates a collaborative environment for job seekers where all they must do is simply post their information regarding their respective job expertise on our app! and the user can get to choose a cook, driver, accountant, hairdresser, sales executive, tailor, photographer etc. depending upon their needs from the available options and hire them temporarily. The main objective of our application is, it allows the users to find people nearer to them and get the work done from people having knowledge with the concerned problem and provides a great user interface, where the user can easily hire people with relevant industry experience to complete their work.

Contents

CHAPTER 1

TITLE AND DESCRIPTION: -

“**WORK AT YOUR WILL**” is a Android-based application where people who want to hire workers for their household works like Electricals , Carpentry can find a worker for the need. Here, the user who want to hire can book worker services with the workers at their own time by signing up to WORK AT YOUR WILL using their email id/mobile no. In return the worker will verify their service requests and approves for services . Then the user needs to complete the payment and can get the service. WORK AT YOUR WILL provides affordable and reliable services for their need.

1.1 Motivation:

Many people can’t go outside to complete their work . The user can book the services from our android application. So, our Android application will help them to get proper affordable and reliable services for their need.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATIONS:

2.1 Introduction

2.1.1 Description of Problem Statement**:**

In the current competitive world, the people wander around cities for various reasons like higher education, jobs and livelihood. So, to them searching for a services for their household work is a time dependent task. And also, to have them in desired style of preferences. Developing a Android application which will be helpful for the people in search of services with their preferences is encouraged.

Our application has a solution for it in keeping in view of the socio-economic conditions of the users we help them by providing services for their work according to their preferences.

2.1.2 Scope:

This Android application will help people to sort out their works based on their requirements and specifications from a thousand of workes in their locality. The scope of this project is to enable the users to search for workers online. The motive of developing this application is to design a feature rich search engine which can make the search of wokers is an easy task.

2.1.3 Definitions, Acronyms and Abbreviations:

XML - Extensible Markup Language .

2.1.4 Overview:

This document includes a brief description of our project. This chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Description of the modules are also included.

The third chapter provides information about the work related to our project with examples.

The fourth chapter has the information of the technologies used and UML diagrams.

The fifth chapter includes implementation part with the coding part

The sixth chapter is included with the result screenshots of our project.

2.2 General description

2.2.1 Product Perspective:

This system is an Android application which will be used to find wokers and view information about them and also to manage the new information.

This application needs the details of the workers for the user like service type, service details and location. For location the user should give the location link from google maps while updating the details of service they need to complete.

Since this is a data centric product it will need somewhere to store the data. For that a Firestore is used. The Android application will use the Firestore to get data and also to modify the data. But, the modifications of the data by the user are restricted only to their details. All the Firestore communication will go over the Internet.

2.2.2 Product Functions:

With the Android application, the users will be able to search the workers according to their preferences. The results will be based on their criteria.

If a user is searching for the worker on what criteria included in the search a list of services will be displayed. Each worker will have the detailed information. The user will also be provided with the location . The location will be redirected to google maps.

Any kind of modifications in the Android app are done only by the admin and some modification permissions are also given to the wokers restricted only for his/her property.

2.2.3 User Characteristics:

There are two types of users that interact with this system: users i.e., users and wokers.

Users:

- Users have to register into the Android Application and login if account already exists and have to sign up if they are new user.
- The User1 will choose from the list of available service providers and requests for a service.
- The workers will verify the pending requests and confirms the services slot.
- The User1 then needs to finish the payment and join the service.

Workers:

- The workers have to login to the Android Application.
- The workers will verify the details of the services they provide.
- Add the verified services into the Firestore by which it will get available on the Android Application.
- Has access to Firestore and makes necessary modifications as requested by the user.

2.3 Modules description:

- **Registration Module:** This module will help the user to register to the Android Application.
- **Login Module:** The users can login to the Android Application once they have successfully registered.
- **Service Module:** The service details like service type, area, price ,location and photos are shown.
- **Search Module:** The specifications applied while searching a services by the tenant will be applied and respective results are shown in a list.

2.4 System Requirements

2.4.1 Hardware Requirements:

- Mobile with 1 GB RAM is required.
- A Mobile with an internet connection and any internet browser is required for the client to run the Android application.

2.4.2 Software Requirement Specifications:

- FLUTTER/ANDROID, XML (FRONT END)
- JAVA, FIRESTORE(BACK END)

Interpreter:

Android Studio : It features a lightning-fast source code editor, perfect for day to day use with support of hundreds of languages. It also helps to be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-section, snippets etc.

Model:

Model is a single, definitive data source which contains the essential field and behaviour of the data. Usually, one model is one table in the Firestore. Each attribute in the model represents a field of a table in the Firestore. Android studio provides a set of automatically- generated Firestore application programming interfaces (APIs) for the convenience of users.

View:

View is a short form of view file. It is a file containing a XML function which takes Android requests and returns Android responses. A response can be XML content or XML documents or a “404 error” and so on. The logic inside the view function can be arbitrary as long as it returns the desired response. To link the view function with a particular URL we need to use a structure called URL conf which maps URLs to view functions.

Template:

Template XML’s template is a simple text file which can generate a text-based format like HTML and XML. The template contains variables and tags. Variables will be replaced by the result when the template is evaluated. Tags control the logic of the template. We also can modify the variables by using filters. For example, a lowercase filter can convert the variable from uppercase into lowercase.

2.5 Design Constraints:

Software Constraints: User can run the application either on Android with an internet connection and any internet browser.

Hardware Constraints: This system will run on a core processor with minimum of 1GB RAM.

CPU – Intel Core i3/i5 3.60 Ghz • RAM – 4/8 Gb • GPU – NVidia/AMD/Intel Integrated

Acceptance criteria: Before accepting, the developer must check whether the application is running properly or not.

CHAPTER 3

MARKET RESEARCH

Urbancompany.com :

Urbancompany.com is an Android application which is an online home service platform . The platform helps customer book reliable and high quality services – home cleaning ,beauty treatment , massage ,haircuts and more -delivered by trained professionals conveniently at home .Urban Company’s vison is to empower millions of professionals worldwide to deliver services at home like never experienced before .

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture and Technologies:

Front-end:

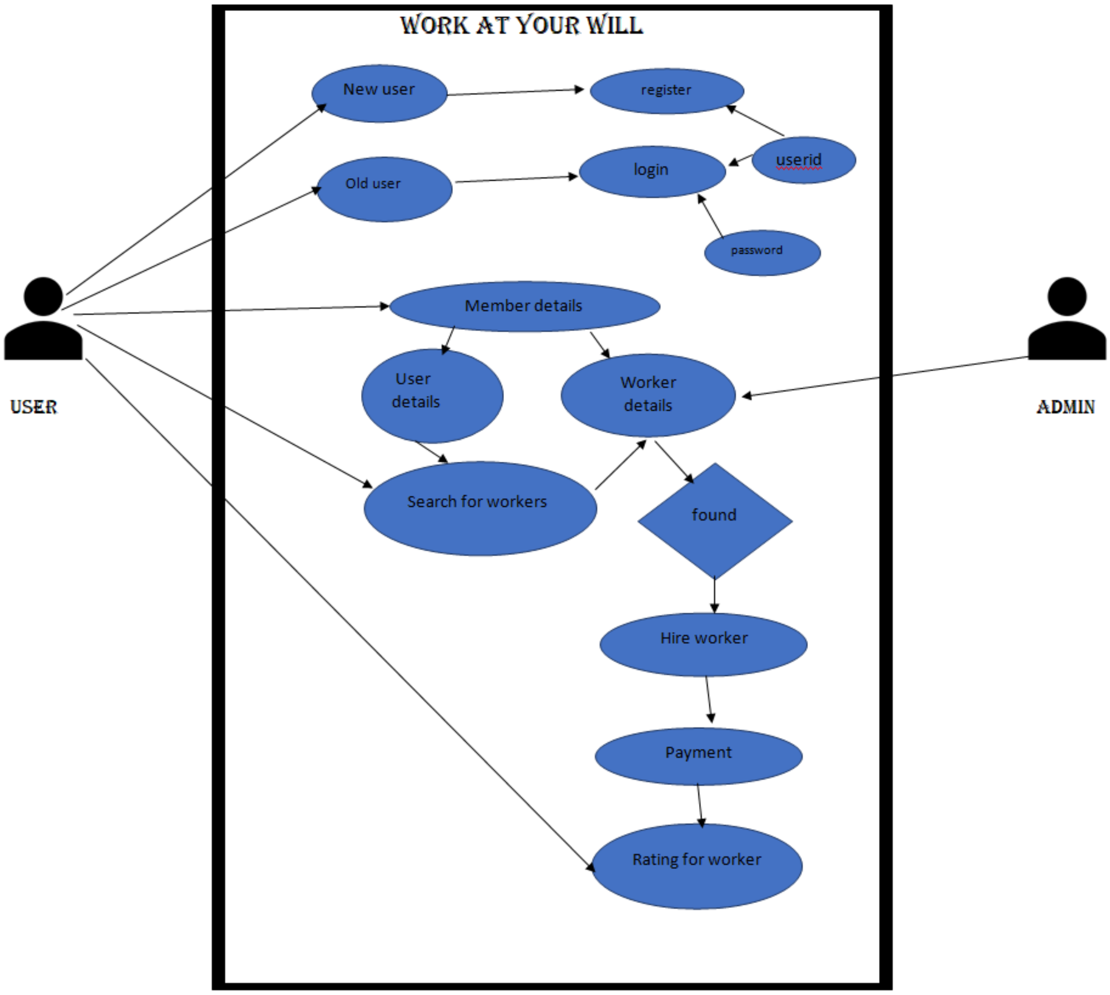
1. XML
2. FLUTTER/ANDROID

Back-end:

1. Google Cloud store(firebase)

4.2 UML Diagrams:

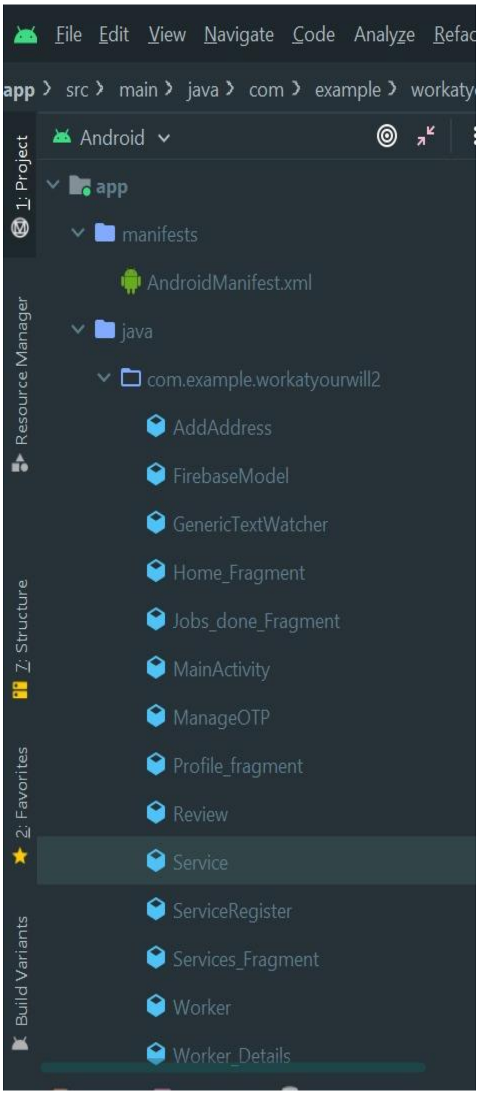
USE CASE DIAGRAM:



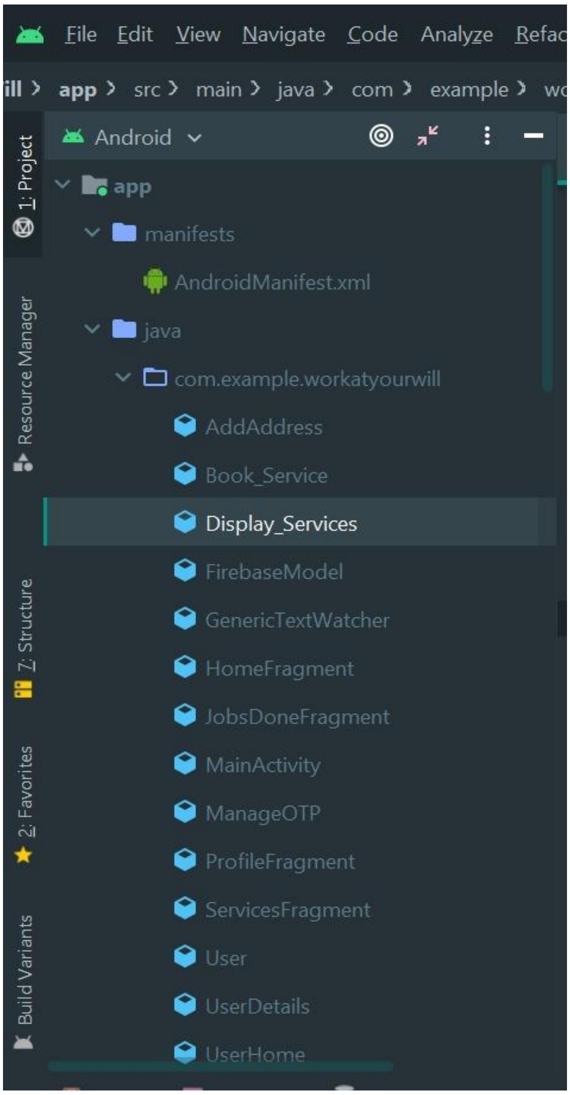
CHAPTER 5

IMPLEMENTATION CODE

5.1 SYSTEM ARCHITECTURE(DESIGN) (User Android)

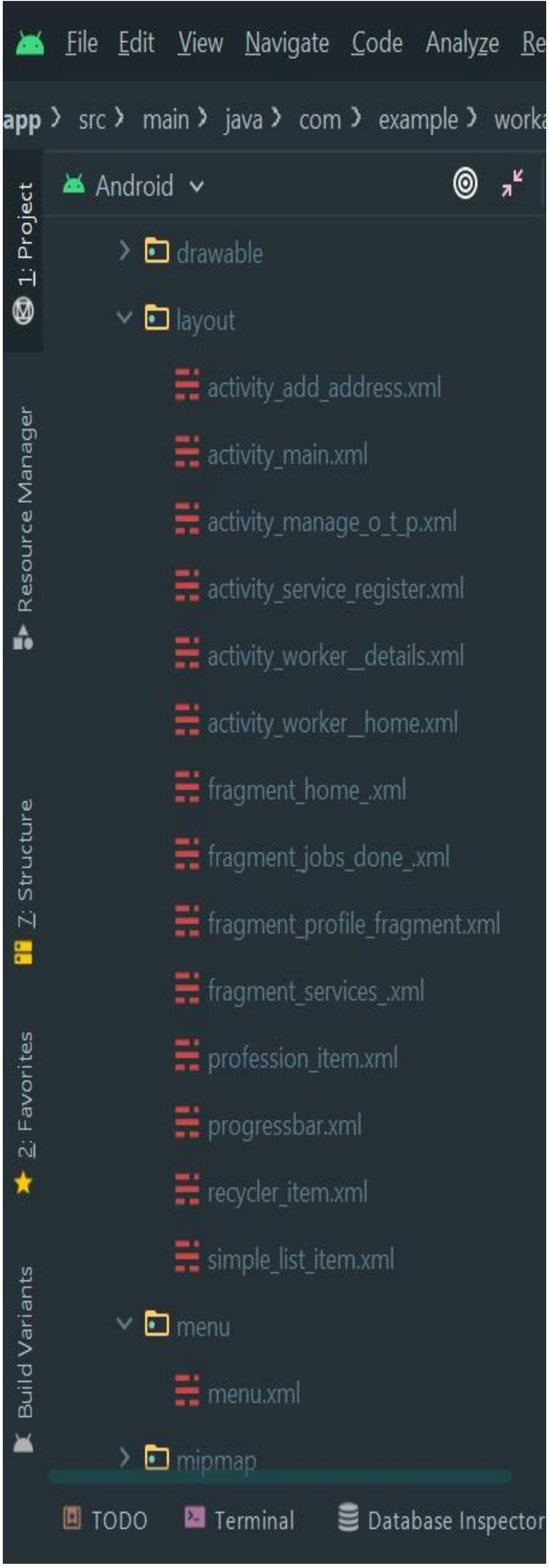

File Edit View Navigate Code Analyze Refac
app › src › main › java › com › example › workaty
Android
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants

```
app
  manifests
    AndroidManifest.xml
  java
    com.example.workatyourwill2
      AddAddress
      FirebaseModel
      GenericTextWatcher
      Home_Fragment
      Jobs_done_Fragment
      MainActivity
      ManageOTP
      Profile_fragment
      Review
      Service
      ServiceRegister
      Services_Fragment
      Worker
      Worker_Details
```

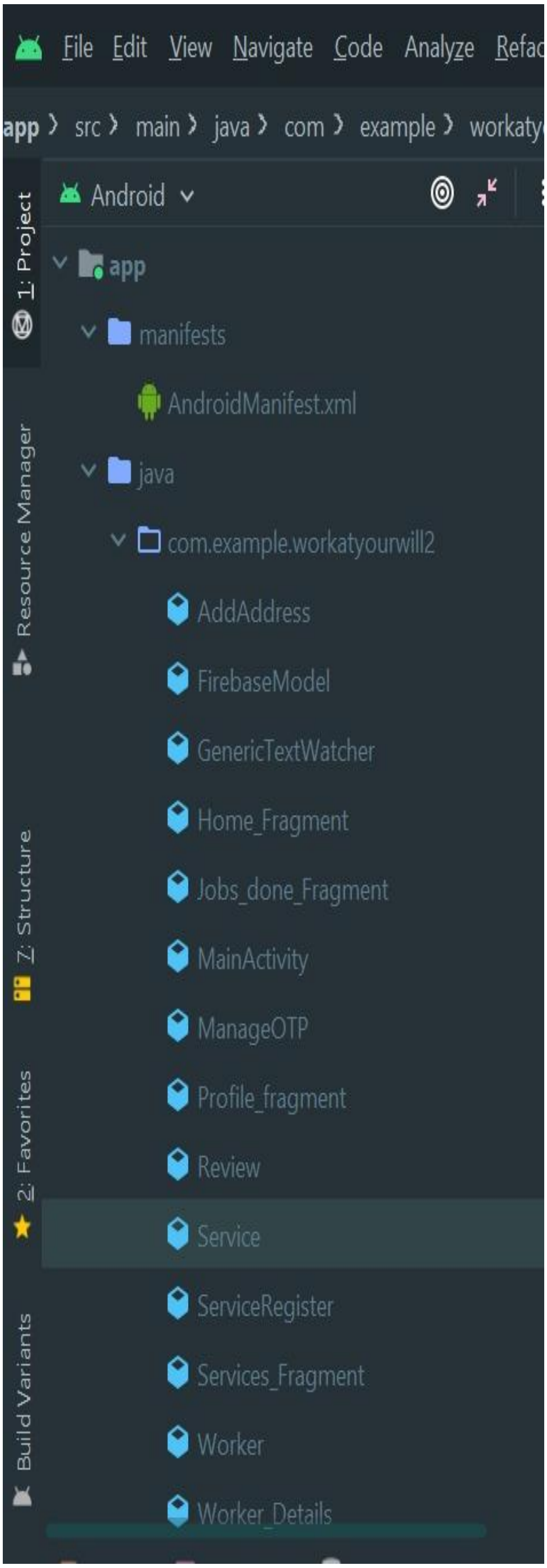

File Edit View Navigate Code Analyze Refac
ill › app › src › main › java › com › example › wo
Android
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants

```
app
  manifests
    AndroidManifest.xml
  java
    com.example.workatyourwill
      AddAddress
      Book_Service
      Display_Services
      FirebaseModel
      GenericTextWatcher
      HomeFragment
      JobsDoneFragment
      MainActivity
      ManageOTP
      ProfileFragment
      ServicesFragment
      User
      UserDetails
      UserHome
```

(Worker Android):



```
› drawable
⌄ layout
    activity_add_address.xml
    activity_main.xml
    activity_manage_o_t_p.xml
    activity_service_register.xml
    activity_worker_details.xml
    activity_worker_home.xml
    fragment_home_.xml
    fragment_jobs_done_.xml
    fragment_profile_fragment.xml
    fragment_services_.xml
    profession_item.xml
    progressbar.xml
    recycler_item.xml
    simple_list_item.xml
⌄ menu
    menu.xml
› mipmap
```


File Edit View Navigate Code Analyze Refac
app › src › main › java › com › example › workaty
Android
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants

```
⌄ app
  ⌄ manifests
      AndroidManifest.xml
  ⌄ java
    ⌄ com.example.workatyourwill2
        AddAddress
        FirebaseModel
        GenericTextWatcher
        Home_Fragment
        Jobs_done_Fragment
        MainActivity
        ManageOTP
        Profile_fragment
        Review
        Service
        ServiceRegister
        Services_Fragment
        Worker
        Worker_Details
```

5.2 IMPLEMENTATION AND CODE

(Worker Code)

(i) **Workers_Home.java**

```
package com.example.workatyourwill2;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.navigation.NavigationBarView;
import com.google.firebase.auth.FirebaseAuth;

public class Worker_Home extends AppCompatActivity {
BottomNavigationView bottomNavigationView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_worker__home);
        bottomNavigationView=findViewById(R.id.bottom_nav);
        getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,new
Home_Fragment()).commit();
        bottomNavigationView.setOnItemSelectedListener(new
NavigationBarView.OnItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                Fragment fragment=null;
                switch (item.getItemId()){
                    case R.id.home:
                        fragment=new Home_Fragment();
                        break;
                    case R.id.profile:
                        fragment=new Profile_fragment();
                        break;
                    case R.id.Services:
                        fragment=new Services_Fragment();
                        break;
                    case R.id.Jobs_done:
                        fragment=new Jobs_done_Fragment();
                        break;
            }
getSupportFragmentManager().beginTransaction().replace(R.id.fragment_container,fragment).commit();
            return true;
          }
        });

    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
    }
}
```

(ii) **Workers_Details .java**

```
package com.example.workatyourwill2;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Settings;
import androidx.annotation.NonNull;

import com.example.workatyourwill2.BuildConfig;
import com.example.workatyourwill2.R;
import com.google.android.gms.common.GooglePlayServicesNotAvailableException;
import com.google.android.gms.common.GooglePlayServicesRepairableException;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.ui.PlacePicker;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.snackbar.Snackbar;

import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.GeoPoint;
import com.google.firebase.firestore.QuerySnapshot;

import org.imperiumlabs.geofirestore.GeoFirestore;

import java.util.Locale;

/**
 * Location sample.
 * <p>
 * Demonstrates use of the Location API to retrieve the last known location for a device.
 */
public class Worker_Details extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();

    private static final int REQUEST_PERMISSIONS_REQUEST_CODE = 34;

    /**
     * Provides the entry point to the Fused Location Provider API.
     */
    private FusedLocationProviderClient mFusedLocationClient;
    AutoCompleteTextView autoCompleteTextView;

Spinner spinner;

    /**
     * Represents a geographical location.
     */
    protected Location mLastLocation;
    private  Button addAddress;
    private String mLatitudeLabel;
    private String mLongitudeLabel;
    private TextView mLatitudeText;
    private TextView mLongitudeText;
    Button signOut;
    FirebaseFirestore firebaseFirestore;
    FirebaseAuth firebaseAuth;
    FirebaseUser firebaseUser;
    GeoFirestore geoFirestore;
    double Latitude;
    double Longitude;
    int PLACE_PICKER_REQUEST=1;
    // Details
    String user_Phone;
    String user_FullName;
    String user_business;
    double user_Loc_Lat;
    double user_Loc_Long;
    String user_Profession;
    EditText editText_F_name;
    EditText editText_Bus_name;
    Button bt_Submit;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_worker__details);


        firebaseFirestore = FirebaseFirestore.getInstance();
        firebaseAuth = FirebaseAuth.getInstance();
        firebaseUser = firebaseAuth.getCurrentUser();
        addAddress=findViewById(R.id.addAddress);
        editText_F_name=findViewById(R.id.et_worker_name);
        editText_Bus_name=findViewById(R.id.et_worker_Org);
        autoCompleteTextView=findViewById(R.id.autoCompleteTextView);
        bt_Submit=findViewById(R.id.bt_submit);
        user_Loc_Long=Double.MIN_VALUE;
        user_Loc_Lat=Double.MIN_VALUE;
        if (firebaseUser == null) {
            startActivity(new Intent(Worker_Details.this, MainActivity.class));
            finish();
        }
        DocumentReference
collectionReference=firebaseFirestore.collection("workers").document(firebaseUser.getUid());
        collectionReference.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {

                if(documentSnapshot.getData()!=null){
//                    Toast.makeText(Worker_Details.this,"Already
present",Toast.LENGTH_SHORT).show();
                    CollectionReference
coll_refer=firebaseFirestore.collection("workers").document(firebaseUser.getUid()).collection("
services");
                coll_refer.get().addOnSuccessListener(new OnSuccessListener<QuerySnapshot>()
{
                    @Override
                    public void onSuccess(QuerySnapshot queryDocumentSnapshots) {
                        int count =queryDocumentSnapshots.getDocuments().size();
                        if(count!=0){
//                  progressDialog.dismiss();

                            startActivity(new Intent(Worker_Details.this,Worker_Home.class));

                            finish();




                        }
                        else{
                            startActivity(new Intent(Worker_Details.this,ServiceRegister.class));
                            finish();
                        }
//              progressDialog.dismiss();


                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
//              progressDialog.dismiss();

Toast.makeText(Worker_Details.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
                });


            }

        }
      }).addOnFailureListener(new OnFailureListener() {
          @Override
          public void onFailure(@NonNull Exception e) {

Toast.makeText(Worker_Details.this,e.getMessage(),Toast.LENGTH_SHORT).show();
          }
      });



      addAddress.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
            if (!checkPermissions()) {
                requestPermissions();
            } else {
                getLastLocation();
                Intent intent=new Intent(Worker_Details.this,AddAddress.class);
                intent.putExtra("latitude",Latitude);
                intent.putExtra("longitude",Longitude);
                startActivityForResult(intent,100);
            }


//              PlacePicker.IntentBuilder builder=new PlacePicker.IntentBuilder();
//              try {
//
startActivityForResult(builder.build(Worker_Details.this),PLACE_PICKER_REQUEST);
//              } catch (GooglePlayServicesRepairableException e) {
//                  e.printStackTrace();
//              } catch (GooglePlayServicesNotAvailableException e) {
//                  e.printStackTrace();
//              }
        }
    });


    user_Phone=firebaseUser.getPhoneNumber();



    String[] professions=getApplication().getResources().getStringArray(R.array.professions);
    ArrayAdapter arrayAdapter=new ArrayAdapter(this,R.layout.profession_item,professions);
    autoCompleteTextView.setAdapter(arrayAdapter);
    user_Profession="";
    autoCompleteTextView.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
            if(i!=0)
            user_Profession=professions[i];
        }
    });




    CollectionReference documentReference =
firebaseFirestore.collection("worker_locations");
    geoFirestore = new GeoFirestore(documentReference);
//      signOut.setOnClickListener(new View.OnClickListener() {
//          @Override
//          public void onClick(View view) {
//
```

```
//                  FirebaseAuth.getInstance().signOut();
//                  startActivity(new Intent(Worker_Details.this, MainActivity.class));
//                  finish();
//              }
//          });
        bt_Submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                user_FullName=editText_F_name.getText().toString();
                user_business=editText_Bus_name.getText().toString();
                user_Phone=firebaseUser.getPhoneNumber();




if(!user_FullName.isEmpty()&&!user_business.isEmpty()&&!user_Phone.isEmpty()&&!user_
Profession.equals("")&&user_Loc_Lat!=Double.MIN_VALUE&&user_Loc_Long!=Double.MI
N_VALUE){
                    DocumentReference
documentReference1=firebaseFirestore.collection("workers").document(firebaseUser.getUid());
                    Worker worker=new
Worker(firebaseUser.getUid(),user_FullName,user_business,user_Phone,new
GeoPoint(user_Loc_Lat,user_Loc_Long),user_Profession);
                    documentReference1.set(worker).addOnSuccessListener(new
OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            Toast.makeText(Worker_Details.this,"Submitted
Succesfully.",Toast.LENGTH_SHORT).show();
                         startActivity(new Intent(Worker_Details.this,ServiceRegister.class));

                        }
                    });

                }
                else{
                    Toast.makeText(Worker_Details.this,"Fields are
emtpy!",Toast.LENGTH_LONG).show();
                }
            }
        });

    }

    @Override
    public void onStart() {
        super.onStart();

        if (!checkPermissions()) {
            requestPermissions();
        } else {
            getLastLocation();
        }
    }

    /**
     * Provides a simple way of getting a device's location and is well suited for
     * applications that do not require a fine-grained location and that do not need location
     * updates. Gets the best and most recent location currently available, which may be null
     * in rare cases when a location is not available.
     * <p>
     * Note: this method should be called after location permission has been granted.
     */
    @SuppressWarnings("MissingPermission")
    private void getLastLocation() {
        mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
        mFusedLocationClient.getLastLocation()
                .addOnCompleteListener(this, new OnCompleteListener<Location>() {
                    @Override
                    public void onComplete(@NonNull Task<Location> task) {
                        if (task.isSuccessful() && task.getResult() != null) {
                            mLastLocation = task.getResult();
                            geoFirestore.setLocation(firebaseUser.getUid(), new
GeoPoint(mLastLocation.getLatitude(), mLastLocation.getLongitude()));
                            Latitude=mLastLocation.getLatitude();
                            Longitude=mLastLocation.getLongitude();

Toast.makeText(Worker_Details.this,mLastLocation.getLatitude()+","+mLastLocation.getLongi
tude(),Toast.LENGTH_SHORT).show();

                        } else {
                            Log.d(TAG, "getLastLocation:exception", task.getException());
                            showSnackbar(getString(R.string.no_location_detected));
                        }
                    }
                });
    }

    /**
     * Shows a {@link Snackbar} using {@code text}.
     *
     * @param text The Snackbar text.
     */
    private void showSnackbar(final String text) {
        View container = findViewById(R.id.main_activity_container);
        if (container != null) {
            Snackbar.make(container, text, Snackbar.LENGTH_LONG).show();
        }
    }

    /**
     * Shows a {@link Snackbar}.
     *
     * @param mainTextStringId The id for the string resource for the Snackbar text.
     * @param actionStringId   The text of the action item.
     * @param listener         The listener associated with the Snackbar action.
     */
    private void showSnackbar(final int mainTextStringId, final int actionStringId,
                              View.OnClickListener listener) {
        Snackbar.make(findViewById(android.R.id.content),
                getString(mainTextStringId),
                Snackbar.LENGTH_INDEFINITE)
                .setAction(getString(actionStringId), listener).show();
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==PLACE_PICKER_REQUEST){
            if(resultCode==RESULT_OK){
                Place place=PlacePicker.getPlace(data,this);
                StringBuilder stringBuilder=new StringBuilder();
                String latitude=String.valueOf(place.getLatLng().latitude);
                String longitude=String.valueOf(place.getLatLng().longitude);
                stringBuilder.append(latitude+","+longitude);

Toast.makeText(Worker_Details.this,stringBuilder.toString(),Toast.LENGTH_SHORT).show();


            }
        }
        else if(requestCode==100){
            if(resultCode==RESULT_OK){
                user_Loc_Lat=data.getDoubleExtra("latitude",0);
                user_Loc_Long=data.getDoubleExtra("longitude",0);

Toast.makeText(Worker_Details.this,user_Loc_Lat+","+user_Loc_Long,Toast.LENGTH_SHO
RT).show();
            }
        }
    }

    /**
     * Return the current state of the permissions needed.
     */

    private boolean checkPermissions() {
        int permissionState = ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_COARSE_LOCATION);
        return permissionState == PackageManager.PERMISSION_GRANTED;
    }

    private void startLocationPermissionRequest() {
        ActivityCompat.requestPermissions(Worker_Details.this,
                new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
                REQUEST_PERMISSIONS_REQUEST_CODE);
    }

    private void requestPermissions() {
        boolean shouldProvideRationale =
                ActivityCompat.shouldShowRequestPermissionRationale(this,
                        Manifest.permission.ACCESS_COARSE_LOCATION);

        // Provide an additional rationale to the user. This would happen if the user denied the
        // request previously, but didn't check the "Don't ask again" checkbox.
        if (shouldProvideRationale) {
            Log.i(TAG, "Displaying permission rationale to provide additional context.");

            showSnackbar(R.string.permission_rationale, android.R.string.ok,
                    new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            // Request permission
                            startLocationPermissionRequest();
                        }
                    });

        } else {
            Log.i(TAG, "Requesting permission");
            // Request permission. It's possible this can be auto answered if device policy
            // sets the permission in a given state or the user denied the permission
            // previously and checked "Never ask again".
            startLocationPermissionRequest();
        }
    }

    /**
     * Callback received when a permissions request has been completed.
     */
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                           @NonNull int[] grantResults) {
        Log.i(TAG, "onRequestPermissionResult");
        if (requestCode == REQUEST_PERMISSIONS_REQUEST_CODE) {
            if (grantResults.length <= 0) {
                // If user interaction was interrupted, the permission request is cancelled and you
                // receive empty arrays.
                Log.i(TAG, "User interaction was cancelled.");
            } else if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // Permission granted.
                getLastLocation();
            } else {
```

```
            showSnackbar(R.string.permission_denied_explanation, R.string.settings,
                    new View.OnClickListener() {
                        @Override
                        public void onClick(View view) {
                            // Build intent that displays the App settings screen.
                            Intent intent = new Intent();
                            intent.setAction(
                                    Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
                            Uri uri = Uri.fromParts("package",
                                    BuildConfig.APPLICATION_ID, null);
                            intent.setData(uri);
                            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                            startActivity(intent);
                        }
                    });
        }
    }
  }
}
```

(iii) **Workers.java** :

```
package com.example.workatyourwill2;

import com.google.android.gms.maps.model.LatLng;
import com.google.firebase.firestore.GeoPoint;

public class Worker {
    private String worker_id;
    private String name;
    private String business_name;
    private String Ph_No;
    private GeoPoint latLng;
    private String profession;


    public Worker(String worker_id,String name, String business_name,String ph_No, GeoPoint
latLng, String profession) {
        this.worker_id=worker_id;
        this.name = name;
        this.business_name=business_name;
        Ph_No = ph_No;
        this.latLng = latLng;
        this.profession = profession;
    }

    public String getWorker_id() {
        return worker_id;
    }
    public void setWorker_id(String worker_id) {
        this.worker_id = worker_id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPh_No() {
        return Ph_No;
    }

    public void setPh_No(String ph_No) {
        Ph_No = ph_No;
    }

    public GeoPoint getLatLng() {
        return latLng;
    }

    public void setLatLng(GeoPoint latLng) {
        this.latLng = latLng;
    }

    public String getProfession() {
        return profession;
    }

    public String getBusiness_name() {
        return business_name;
    }

    public void setBusiness_name(String business_name) {
        this.business_name = business_name;
    }

    public void setProfession(String profession) {
        this.profession = profession;
    }
}
```

(iv) **Services_Fragments.java**

```
package com.example.workatyourwill2;

import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;

import android.provider.ContactsContract;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.firebase.ui.firestore.FirestoreRecyclerAdapter;
import com.firebase.ui.firestore.FirestoreRecyclerOptions;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.Query;

import java.text.DateFormat;
import java.util.Date;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link Services_Fragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class Services_Fragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
    FirebaseAuth firebaseAuth;
    FirebaseUser firebaseUser;
    FirebaseFirestore firebaseFirestore;
    RecyclerView recyclerView;
    FloatingActionButton fab;
    FirestoreRecyclerAdapter<FirebaseModel,NoteViewHolder> serviceAdapter;
    StaggeredGridLayoutManager staggeredGridLayoutManager;
    public Services_Fragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment Services_Fragment.
     */
    // TODO: Rename and change types and number of parameters
    public static Services_Fragment newInstance(String param1, String param2) {
        Services_Fragment fragment = new Services_Fragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view=inflater.inflate(R.layout.fragment_services_, container, false);
        fab=view.findViewById(R.id.fab);
        recyclerView=view.findViewById(R.id.recyclerView);
        firebaseAuth=FirebaseAuth.getInstance();
        firebaseUser=firebaseAuth.getCurrentUser();
        firebaseFirestore=FirebaseFirestore.getInstance();

        staggeredGridLayoutManager=new StaggeredGridLayoutManager(1,
StaggeredGridLayoutManager.VERTICAL);
        Query
query=firebaseFirestore.collection("workers").document(firebaseUser.getUid()).collection("serv
ices");

        FirestoreRecyclerOptions<FirebaseModel> allServices=new
FirestoreRecyclerOptions.Builder<FirebaseModel>().setQuery(query,FirebaseModel.class).buil
d();
        serviceAdapter=new FirestoreRecyclerAdapter<FirebaseModel,
NoteViewHolder>(allServices) {
```

```
            @Override
            protected void onBindViewHolder(@NonNull NoteViewHolder holder, int position,
@NonNull FirebaseModel model) {
                String serv_title=model.getS_name();
                String serv_desc=model.getS_desc();
                String data="jflsjfse";
                double cost=model.getS_cost();
                Date date_added=model.getDate_added();
                String rating=model.getS_rating()+"";
                String bookings_month=model.getS_bookings_per_month()+"";
                data= DateFormat.getDateInstance().format(date_added);
                holder.ServiceTitle.setText(serv_title);
                holder.ServiceDesc.setText(serv_desc);
                holder.dateAdded.setText(data);
                holder.cost.setText(cost+"");
                holder.rating.setText(rating);
                holder.bookings.setText(bookings_month);



                String docId=serviceAdapter.getSnapshots().getSnapshot(position).getId();

            }

            @NonNull
            @Override
            public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
                View
view1=LayoutInflater.from(parent.getContext()).inflate(R.layout.recycler_item,parent,false);
                return new NoteViewHolder(view1);
            }
        };
        recyclerView.setHasFixedSize(true);

        recyclerView.setLayoutManager(staggeredGridLayoutManager);
        recyclerView.setAdapter(serviceAdapter);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(container.getContext(),ServiceRegister.class));

            }
        });
        return view;


    }
    public class NoteViewHolder extends RecyclerView.ViewHolder{
        public TextView ServiceTitle;
        public TextView ServiceDesc;
        LinearLayout mNoteLayout;
        CardView cardView;
        ImageView popupButton;
        TextView dateAdded;
        TextView cost;
        TextView rating;
        TextView bookings;

        public NoteViewHolder(@NonNull View itemView) {
            super(itemView);
            ServiceTitle=itemView.findViewById(R.id.noteTitle);
            ServiceDesc=itemView.findViewById(R.id.noteContent);
            mNoteLayout=itemView.findViewById(R.id.noteLayout);
            cardView=itemView.findViewById(R.id.notecard);
            dateAdded=itemView.findViewById(R.id.notelastedited);
            cost=itemView.findViewById(R.id.note_cost);
            popupButton=itemView.findViewById(R.id.menupopupButton);
            rating=itemView.findViewById(R.id.rating);
            bookings=itemView.findViewById(R.id.booking_month);


        }


    }

    @Override
    public void onStart() {
        super.onStart();
        serviceAdapter.startListening();
    }
}
```

(v) **ServiceRegister.java**

```
package com.example.workatyourwill2;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
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.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.Date;

public class ServiceRegister extends AppCompatActivity {

    EditText ser_name;
    EditText ser_desc;
    EditText ser_cost;
    Button add_service;
    TextView skip_add;
    FirebaseAuth firebaseAuth;
    FirebaseUser firebaseUser;
    FirebaseFirestore firebaseFirestore;
    ProgressDialog progressDialog;
    TextView progresstext;
    String occupation="";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_service_register);
        ser_name=findViewById(R.id.et_service_name);
        ser_desc=findViewById(R.id.et_service_desc);
        ser_cost=findViewById(R.id.et_service_cost);
        add_service=findViewById(R.id.addService);
        skip_add=findViewById(R.id.skipAddService);
        firebaseAuth=FirebaseAuth.getInstance();
        firebaseUser=firebaseAuth.getCurrentUser();

        firebaseFirestore=FirebaseFirestore.getInstance();
        if(firebaseUser==null){
            startActivity(new Intent(ServiceRegister.this,MainActivity.class));
            finish();
        }

        String[] profession =new String[1];
        add_service.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String name=ser_name.getText().toString().trim();
                String desc=ser_desc.getText().toString().trim();
                int cost=Integer.parseInt(ser_cost.getText().toString());
                if(name.isEmpty()||desc.isEmpty()||ser_cost.getText().toString().isEmpty()){
                   Toast.makeText(ServiceRegister.this,"Fields are
empty!",Toast.LENGTH_SHORT).show();
                   return;
                }
                DocumentReference
docRef=firebaseFirestore.collection("workers").document(firebaseUser.getUid());
                docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
                   @Override
                   public void onSuccess(DocumentSnapshot documentSnapshot) {
                       String prof =(String)documentSnapshot.get("profession");

                       Toast.makeText(ServiceRegister.this,prof,Toast.LENGTH_SHORT).show();
                      DocumentReference
collectionReference=firebaseFirestore.collection("workers").document(firebaseUser.getUid()).c
ollection("services").document();
                      Date date_added=new Date();
                      Service service=new
Service(prof,date_added,name,desc,cost,firebaseUser.getUid());
                      collectionReference.set(service).addOnSuccessListener(new
OnSuccessListener<Void>() {
                     @Override
                     public void onSuccess(Void aVoid) {
                        Toast.makeText(ServiceRegister.this,"Added Service
Successfully.",Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(ServiceRegister.this,Worker_Home.class));
                        finish();
                     }
                      }).addOnFailureListener(new OnFailureListener() {
                     @Override
                     public void onFailure(@NonNull Exception e) {
                        Toast.makeText(ServiceRegister.this,"Couldn't add
Service!.",Toast.LENGTH_SHORT).show();
                     }
                      });

                   }
                }).addOnFailureListener(new OnFailureListener() {
                   @Override
                   public void onFailure(@NonNull Exception e) {
                      Toast.makeText(ServiceRegister.this,"Couldn't get
professtion!",Toast.LENGTH_SHORT).show();
                   }
                });



            }
        });
        skip_add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(ServiceRegister.this,Worker_Home.class));
                finish();
            }
        });


    }

    @Override
    protected void onStart() {
        super.onStart();
progressDialog.getWindow().setBackgroundDrawableResource(android.R.color.transparent);

    }
}
```

(vi) **Service.java**

```
package com.example.workatyourwill2;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Date;

public class Service {
    public Date date_added;
    public String s_name;
    public String s_desc;
    public String category;
    public int s_cost;
    public double s_rating=0;
    public String worker_id;
    public int s_bookings_per_month=0;
    public ArrayList<Review> s_reviews;

    public Date getDate_added() {
        return date_added;
    }

    public void setDate_added(Date date_added) {
        this.date_added = date_added;
    }
    public Service(){

    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    public Service(String category, Date date_added, String s_name, String s_desc, int s_cost,
String worker_id) {
        this.category=category;
        this.date_added=date_added;
        this.s_name = s_name;
        this.s_desc = s_desc;
        this.s_cost = s_cost;

        this.worker_id = worker_id;
    }

    public String getS_name() {
        return s_name;
    }

    public void setS_name(String s_name) {
        this.s_name = s_name;
    }

    public String getS_desc() {
        return s_desc;
    }

    public void setS_desc(String s_desc) {
        this.s_desc = s_desc;
    }

    public double getS_cost() {
        return s_cost;
    }

    public void setS_cost(int s_cost) {
        this.s_cost = s_cost;
    }

    public double getS_rating() {
        return s_rating;
    }

    public void setS_rating(double s_rating) {
        this.s_rating = s_rating;
    }

    public String getWorker_id() {
        return worker_id;
    }

    public void setWorker_id(String worker_id) {
        this.worker_id = worker_id;
    }

    public int getS_bookings_per_month() {
        return s_bookings_per_month;
    }

    public void setS_bookings_per_month(int s_bookings_per_month) {
        this.s_bookings_per_month = s_bookings_per_month;
    }

    public ArrayList<Review> getS_reviews() {
        return s_reviews;
    }

    public void setS_reviews(ArrayList<Review> s_reviews) {
        this.s_reviews = s_reviews;
    }
}
```

(vii)**Review.java**

```
package com.example.workatyourwill2;

import java.util.Date;

public class Review {
    double rating;
    Date date;
    String user_id;

    public Review(double rating, Date date, String user_id) {
        this.rating = rating;
        this.date = date;
        this.user_id = user_id;
    }

    public double getRating() {
        return rating;
    }

    public void setRating(double rating) {
        this.rating = rating;
    }

    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }

    public String getUser_id() {
        return user_id;
    }

    public void setUser_id(String user_id) {
        this.user_id = user_id;
    }
}
```

(viii) **ManageOTP.java**

```
package com.example.workatyourwill2;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
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.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthOptions;
import com.google.firebase.auth.PhoneAuthProvider;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.concurrent.TimeUnit;

import static android.content.ContentValues.TAG;
```

```
public class ManageOTP extends AppCompatActivity {
    EditText otp_textbox_one, otp_textbox_two, otp_textbox_three,
otp_textbox_four,otp_textbox_five,otp_textbox_six;
Button verifyOTP;
String PhNo;
FirebaseAuth mAuth;
FirebaseFirestore firebaseFirestore;
FirebaseUser firebaseUser;

String otpID;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manage_o_t_p);
        otp_textbox_one=findViewById(R.id.otp_edit_box1);
        otp_textbox_two=findViewById(R.id.otp_edit_box2);
        otp_textbox_three=findViewById(R.id.otp_edit_box3);
        otp_textbox_four=findViewById(R.id.otp_edit_box4);
        otp_textbox_five=findViewById(R.id.otp_edit_box5);
        otp_textbox_six=findViewById(R.id.otp_edit_box6);
        verifyOTP=findViewById(R.id.verifyOTP);
        EditText[] edit = {otp_textbox_one, otp_textbox_two, otp_textbox_three,
otp_textbox_four,otp_textbox_five,otp_textbox_six};

        otp_textbox_one.addTextChangedListener(new GenericTextWatcher(otp_textbox_one,
edit));
        otp_textbox_two.addTextChangedListener(new GenericTextWatcher(otp_textbox_two,
edit));
        otp_textbox_three.addTextChangedListener(new GenericTextWatcher(otp_textbox_three,
edit));
        otp_textbox_four.addTextChangedListener(new GenericTextWatcher(otp_textbox_four,
edit));
        otp_textbox_five.addTextChangedListener(new GenericTextWatcher(otp_textbox_five,
edit));
        otp_textbox_six.addTextChangedListener(new GenericTextWatcher(otp_textbox_six,
edit));
        Intent intent=getIntent();
        mAuth=FirebaseAuth.getInstance();
       PhNo= intent.getStringExtra("mobile").toString();
        initiateOTP();
        verifyOTP.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String c1=otp_textbox_one.getText().toString();
                String c2=otp_textbox_two.getText().toString();
                String c3=otp_textbox_three.getText().toString();
                String c4=otp_textbox_four.getText().toString();
                String c5=otp_textbox_five.getText().toString();
                String c6=otp_textbox_six.getText().toString();
                String enteredOTP=c1+c2+c3+c4+c5+c6;
                if(c1.isEmpty()||c2.isEmpty()||c3.isEmpty()||c4.isEmpty()||c5.isEmpty()||c6.isEmpty())
                {
                    Toast.makeText(getApplicationContext(),"Please Enter
OTP",Toast.LENGTH_SHORT).show();
                }
                else
                {
                    PhoneAuthCredential
credential=PhoneAuthProvider.getCredential(otpID,enteredOTP);
                    signInWithPhoneAuthCredential(credential);
                }
            }
        });



    }
    private void initiateOTP(){
        PhoneAuthProvider.getInstance().verifyPhoneNumber(
                PhNo,        // Phone number to verify
                60,                 // Timeout duration
                TimeUnit.SECONDS,   // Unit of timeout
                this,               // Activity (for callback binding)
                new PhoneAuthProvider.OnVerificationStateChangedCallbacks()
                {
                    @Override
                    public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken
forceResendingToken)
                    {
                        otpID=s;
                    }

                    @Override
                    public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential)
                    {
                        signInWithPhoneAuthCredential(phoneAuthCredential);
                    }

                    @Override
                    public void onVerificationFailed(FirebaseException e) {

Toast.makeText(getApplicationContext(),e.getMessage(),Toast.LENGTH_LONG).show();
                    }
                });        // OnVerificationStateChangedCallbacks

    }
    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
```

```
            if (task.isSuccessful()) {
                firebaseUser=mAuth.getCurrentUser();
                firebaseFirestore=FirebaseFirestore.getInstance();
                DocumentReference
collectionReference=firebaseFirestore.collection("workers").document(firebaseUser.getUid());
                collectionReference.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
                    @Override
                    public void onSuccess(DocumentSnapshot documentSnapshot) {

                        if(documentSnapshot.getData()!=null){
//                    Toast.makeText(Worker_Details.this,"Already
present",Toast.LENGTH_SHORT).show();
                            CollectionReference
coll_refer=firebaseFirestore.collection("workers").document(firebaseUser.getUid()).collection("
services");
                            coll_refer.get().addOnSuccessListener(new
OnSuccessListener<QuerySnapshot>() {
                                @Override
                                public void onSuccess(QuerySnapshot queryDocumentSnapshots) {
                                    int count =queryDocumentSnapshots.getDocuments().size();
                                    if(count!=0){
//                    progressDialog.dismiss();

                                        startActivity(new
Intent(ManageOTP.this,Worker_Home.class));

                                        finish();




                                    }
                                    else{
                                        startActivity(new
Intent(ManageOTP.this,ServiceRegister.class));
                                        finish();
                                    }
//                progressDialog.dismiss();


                                }
                            }).addOnFailureListener(new OnFailureListener() {
                                @Override
                                public void onFailure(@NonNull Exception e) {
//                progressDialog.dismiss();

Toast.makeText(ManageOTP.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                                }
                            });
                        }

                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

Toast.makeText(ManageOTP.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
                });


                startActivity(new Intent(ManageOTP.this,Worker_Details.class));
                finish();
            } else {
                Toast.makeText(getApplicationContext(),"SignIn Code
error",Toast.LENGTH_SHORT).show();
            }
        }
    });
}
}
```

(ix) **Profile_fragment.java**

```
package com.example.workatyourwill2;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.ImageView;
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.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link Profile_fragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class Profile_fragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
    FirebaseUser firebaseUser;
    FirebaseAuth firebaseAuth;
    FirebaseFirestore firebaseFirestore;
    ImageButton imageButton;
    TextView userName;
    TextView userPhn;

    public Profile_fragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment Profile_fragment.
     */
    // TODO: Rename and change types and number of parameters
    public static Profile_fragment newInstance(String param1, String param2) {
        Profile_fragment fragment = new Profile_fragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view =inflater.inflate(R.layout.fragment_profile_fragment, container, false);
        imageButton=view.findViewById(R.id.user_profile_photo);
        userName=view.findViewById(R.id.user_profile_name);
        userPhn=view.findViewById(R.id.user_profile_short_bio);
        firebaseAuth=FirebaseAuth.getInstance();
        firebaseUser=firebaseAuth.getCurrentUser();
        firebaseFirestore=FirebaseFirestore.getInstance();
        String userNam=firebaseUser.getDisplayName();
        Uri userimg=firebaseUser.getPhotoUrl();
        DocumentReference
documentReference=firebaseFirestore.collection("workers").document(firebaseUser.getUid());

        documentReference.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                 String name=(String)documentSnapshot.get("name");
                 userName.setText(name);
                Toast.makeText(container.getContext(),"Success",Toast.LENGTH_SHORT).show();

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

            }
        });
        userName.setText(userNam);
        userPhn.setText(firebaseUser.getPhoneNumber());
        return view;

    }
}
```

(x**) MainActivity.java :**

```
package com.example.workatyourwill2;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
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.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;
import com.hbb20.CountryCodePicker;

public class MainActivity extends AppCompatActivity {
CountryCodePicker ccp;
EditText PhNo;
Button getOTP;
FirebaseAuth firebaseAuth;
FirebaseUser firebaseUser;
FirebaseFirestore firebaseFirestore;
ProgressDialog progressDialog;
TextView progresstext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ccp=(CountryCodePicker)findViewById(R.id.ccp);
        PhNo=findViewById(R.id.PhNo);
        getOTP=findViewById(R.id.getOTP);
        ccp.registerCarrierNumberEditText(PhNo);
        firebaseAuth=FirebaseAuth.getInstance();
        firebaseUser=firebaseAuth.getCurrentUser();

        if(firebaseUser!=null){
            progressDialog=new ProgressDialog(MainActivity.this);

            progressDialog.show();

            progressDialog.setContentView(R.layout.progressbar);


progressDialog.getWindow().setBackgroundDrawableResource(android.R.color.transparent);
            progresstext=progressDialog.findViewById(R.id.progress_text);
```

```
          progresstext.setText("Signing in..");
          firebaseFirestore= FirebaseFirestore.getInstance();
          DocumentReference
collectionReference=firebaseFirestore.collection("workers").document(firebaseUser.getUid());
          collectionReference.get().addOnSuccessListener(new
OnSuccessListener<DocumentSnapshot>() {
             @Override
             public void onSuccess(DocumentSnapshot documentSnapshot) {

                if(documentSnapshot.getData()!=null){
//                Toast.makeText(Worker_Details.this,"Already
present",Toast.LENGTH_SHORT).show();
                   CollectionReference
coll_refer=firebaseFirestore.collection("workers").document(firebaseUser.getUid()).collection("
services");
                   coll_refer.get().addOnSuccessListener(new
OnSuccessListener<QuerySnapshot>() {
                       @Override
                       public void onSuccess(QuerySnapshot queryDocumentSnapshots) {
                          int count =queryDocumentSnapshots.getDocuments().size();
                          if(count!=0){
//                progressDialog.dismiss();
                             progressDialog.dismiss();
                             startActivity(new Intent(MainActivity.this,Worker_Home.class));

                             finish();




                          }
                          else{
                             progressDialog.dismiss();
                             startActivity(new Intent(MainActivity.this,ServiceRegister.class));
                             finish();
                          }
//             progressDialog.dismiss();


                       }
                   }).addOnFailureListener(new OnFailureListener() {
                       @Override
                       public void onFailure(@NonNull Exception e) {
//             progressDialog.dismiss();
                          progressDialog.dismiss();

Toast.makeText(MainActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
                       }
                   });
                }
                else{
                    progressDialog.dismiss();
                    startActivity(new Intent(MainActivity.this,Worker_Details.class));
                    finish();
                }


            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                progressDialog.dismiss();

Toast.makeText(MainActivity.this,e.getMessage(),Toast.LENGTH_SHORT).show();
            }
        });



    }
    getOTP.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent=new Intent(MainActivity.this,ManageOTP.class);
            intent.putExtra("mobile",ccp.getFullNumberWithPlus().replace(" ",""));
            startActivity(intent);
        }
    });
  }
}
```

(xi) **Jobs_done_Fragment.java**

```
package com.example.workatyourwill2;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link Jobs_done_Fragment#newInstance} factory method to
 * create an instance of this fragment.
 */
```

```
public class Jobs_done_Fragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    public Jobs_done_Fragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment Jobs_done_Fragment.
     */
    // TODO: Rename and change types and number of parameters
    public static Jobs_done_Fragment newInstance(String param1, String param2) {
        Jobs_done_Fragment fragment = new Jobs_done_Fragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_jobs_done_, container, false);
    }
}
```

(xii) **Home_Fragment.java**

```
package com.example.workatyourwill2;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link Home_Fragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class Home_Fragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    public Home_Fragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment Home_Fragment.
     */
    // TODO: Rename and change types and number of parameters
    public static Home_Fragment newInstance(String param1, String param2) {
        Home_Fragment fragment = new Home_Fragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home_, container, false);
    }
}
```

(xiii) **GenericTextWatcher.java**

```
package com.example.workatyourwill2;

import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.EditText;

public class GenericTextWatcher implements TextWatcher {
    private final EditText[] editText;
    private View view;
    public GenericTextWatcher(View view, EditText editText[])
    {
        this.editText = editText;
        this.view = view;
    }

    @Override
    public void afterTextChanged(Editable editable) {
        String text = editable.toString();
        switch (view.getId()) {

            case R.id.otp_edit_box1:
                if (text.length() == 1)
                    editText[1].requestFocus();
                break;
            case R.id.otp_edit_box2:

                if (text.length() == 1)
                    editText[2].requestFocus();
                else if (text.length() == 0)
                    editText[0].requestFocus();
                break;
            case R.id.otp_edit_box3:
                if (text.length() == 1)
                    editText[3].requestFocus();
                else if (text.length() == 0)
```

```
                    editText[1].requestFocus();
                break;
            case R.id.otp_edit_box4:
                if (text.length() == 1)
                    editText[4].requestFocus();
                else if (text.length() == 0)
                    editText[2].requestFocus();
                break;
            case R.id.otp_edit_box5:
                if (text.length() == 1)
                    editText[5].requestFocus();
                else if (text.length() == 0)
                    editText[3].requestFocus();
                break;
            case R.id.otp_edit_box6:
                if (text.length() == 0)
                    editText[4].requestFocus();
                break;
        }
    }

    @Override
    public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
    }

    @Override
    public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3) {
    }
}
```

(xiv) **FirebaseModel.java**

```
package com.example.workatyourwill2;

import java.util.ArrayList;
import java.util.Date;

public class FirebaseModel {
    public Date date_added;
    public String s_name;
    public String s_desc;
    public int s_cost;
    public double s_rating;
    public String worker_id;
    public int s_bookings_per_month;
    public ArrayList<Review> s_reviews;

    public Date getDate_added() {
        return date_added;
    }

    public void setDate_added(Date date_added) {
        this.date_added = date_added;
    }
    public FirebaseModel(){

    }
    public FirebaseModel(Date date_added, String s_name, String s_desc, int s_cost, String
worker_id) {
        this.date_added=date_added;
        this.s_name = s_name;
        this.s_desc = s_desc;
        this.s_cost = s_cost;

        this.worker_id = worker_id;
    }

    public String getS_name() {
        return s_name;
    }

    public void setS_name(String s_name) {
        this.s_name = s_name;
    }

    public String getS_desc() {
        return s_desc;
    }

    public void setS_desc(String s_desc) {
        this.s_desc = s_desc;
    }

    public double getS_cost() {
        return s_cost;
    }

    public void setS_cost(int s_cost) {
        this.s_cost = s_cost;
    }

    public double getS_rating() {
        return s_rating;
    }

    public void setS_rating(double s_rating) {
        this.s_rating = s_rating;
    }

    public String getWorker_id() {
        return worker_id;
    }
    public void setWorker_id(String worker_id) {
        this.worker_id = worker_id;
    }

    public int getS_bookings_per_month() {
        return s_bookings_per_month;
    }

    public void setS_bookings_per_month(int s_bookings_per_month) {
        this.s_bookings_per_month = s_bookings_per_month;
    }

    public ArrayList<Review> getS_reviews() {
        return s_reviews;
    }

    public void setS_reviews(ArrayList<Review> s_reviews) {
        this.s_reviews = s_reviews;
    }
}
```

(xv) **AddAddress.java**

```
package com.example.workatyourwill2;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.net.Uri;
import android.os.Bundle;
//import android.view.View;
//import android.widget.AdapterView;
//import android.widget.ArrayAdapter;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.Toast;
//import android.widget.SearchView;
//import android.widget.Toast;
//
//import com.google.android.gms.common.api.Status;
//import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.location.FusedLocationProviderClient;
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.Marker;
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.android.material.snackbar.Snackbar;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.GeoPoint;

import org.imperiumlabs.geofirestore.GeoFirestore;

import java.io.IOException;
import java.util.List;
//import com.google.android.gms.maps.model.LatLng;
//import com.google.android.gms.maps.model.MarkerOptions;
//import com.google.android.libraries.places.api.Places;
//import com.google.android.libraries.places.api.model.Place;
//import com.google.android.libraries.places.widget.Autocomplete;
//import com.google.android.libraries.places.widget.AutocompleteActivity;
//import com.google.android.libraries.places.widget.model.AutocompleteActivityMode;

//import java.io.IOException;
//import java.util.ArrayList;
//import java.util.Arrays;
//import java.util.List;

public class AddAddress extends FragmentActivity implements OnMapReadyCallback{
    private static final String TAG = MainActivity.class.getSimpleName();
    protected Location mLastLocation;
    private static final int REQUEST_PERMISSIONS_REQUEST_CODE = 34;

    /**
     * Provides the entry point to the Fused Location Provider API.
     */
    private FusedLocationProviderClient mFusedLocationClient;
GoogleMap map;
Button confirm_location;
SupportMapFragment mapFragment;
SearchView searchView;
```

```
ListView listView;
String[] list;
StringBuilder sb;
Marker nm;
boolean map_Ready_flag=false;
double latitude;
double longitude;
    Intent intent;
    MarkerOptions marker;
//    Geocoder geocoder=new Geocoder(AddAddress.this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_address);
        searchView=findViewById(R.id.search_location);
        listView=findViewById(R.id.listView);
        confirm_location=findViewById(R.id.confirm_location);

        searchView.setFocusable(true);
        mapFragment=(SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.google_map);
        intent=getIntent();
        latitude=intent.getDoubleExtra("latitude",0);
        longitude=intent.getDoubleExtra("longitude",0);
//        mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);


//        listView.setVisibility(View.INVISIBLE);

        searchView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                listView.setVisibility(View.VISIBLE);
            }
        });

        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String s) {
                String location=searchView.getQuery().toString();
                List<Address> addresses=null;
                map.clear();

                if(location!=null&&!location.equals("")){
                    Geocoder geocoder=new Geocoder(AddAddress.this);
                    try {
                        addresses=geocoder.getFromLocationName(location,1);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    if(addresses.size()>=1) {
                        Address address = addresses.get(0);
                        LatLng latLng=new LatLng(address.getLatitude(),address.getLongitude());
                        if(nm!=null){
                            nm.remove();
                        }
                        marker =new MarkerOptions().position(latLng).title(location).draggable(true);

                        nm=map.addMarker(marker);

                        map.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));

                        listView.setVisibility(View.INVISIBLE);
                    }
                    else{
                        Toast.makeText(AddAddress.this,"HI",Toast.LENGTH_SHORT).show();
                    }
//                          addresses = geocoder.getFromLocation(address.getLatitude(),
address.getLongitude(), 1); // Here 1 represent max location result to returned, by documents it
recommended 1 to 5

//                          String addressline = addresses.get(0).getAddressLine(0); // If any additional
address line present than only, check with max available address lines by
getMaxAddressLineIndex()
//                          String city = addresses.get(0).getLocality();
//                          String state = addresses.get(0).getAdminArea();
//                          String country = addresses.get(0).getCountryName();
//                          String postalCode = addresses.get(0).getPostalCode();
//                          String knownName = addresses.get(0).getFeatureName();


                }
                return false;
            }

            @Override
            public boolean onQueryTextChange(String s) {
                String location=s;
                List<Address> addresses=null;
                Geocoder geocoder=new Geocoder(AddAddress.this);
                listView.setVisibility(View.VISIBLE);
                if(location!=null&&!location.equals("")){

                    try {
                        addresses=geocoder.getFromLocationName(location,5);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    int n=0;
                    if(addresses.size()<5){
                        n=addresses.size();
                    }
                    else{
                        n=5;
                    }

                    list=new String[n];
                    sb=new StringBuilder();
//                      addresses = geocoder.getFromLocation(address.getLatitude(), 
address.getLongitude(), 1); // Here 1 represent max location result to returned, by documents it 
recommended 1 to 5

                    for(int i=0;i<n;i++) {
                        sb.append(addresses.get(i).getAddressLine(0)); // If any additional address line 
present than only, check with max available address lines by getMaxAddressLineIndex()

                        sb.append(","+addresses.get(i).getAddressLine(0));


                        sb.append(","+addresses.get(i).getLocality());
                        sb.append(","+addresses.get(i).getAdminArea());
                        sb.append(","+addresses.get(i).getCountryName());
                        sb.append(","+addresses.get(i).getPostalCode());
                        sb.append(","+addresses.get(i).getFeatureName());
                        list[i]=sb.toString();
                        sb.setLength(0);


                    }
                    ArrayAdapter<String> arrayAdapter=new ArrayAdapter<>(AddAddress.this, 
R.layout.simple_list_item,list);
//                  arrayAdapter.notifyDataSetChanged();
                    listView.setAdapter(arrayAdapter);
                    List<Address> finalAddresses = addresses;
                    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                        @Override
                        public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) 
{

//                      addresses = geocoder.getFromLocation(address.getLatitude(), 
address.getLongitude(), 1); // Here 1 represent max location result to returned, by documents it 
recommended 1 to 5

                            if(finalAddresses!=null&&finalAddresses.size()>i) {
                                Address address = finalAddresses.get(i);
                                LatLng latLng = new LatLng(address.getLatitude(), 
address.getLongitude());
                                if(nm!=null){
                                    nm.remove();
                                }
                                marker =new 
MarkerOptions().position(latLng).title(location).draggable(true).title(address.getLocality());
                                nm=map.addMarker(marker);
```

```
                                map.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));

                                listView.setVisibility(View.GONE);

                            }
                        }
                    });


                }
                return false;
            }
        });
        mapFragment.getMapAsync(this);
        confirm_location.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                LatLng latLng=nm.getPosition();
                intent.putExtra("latitude",latLng.latitude);
                intent.putExtra("longitude",latLng.longitude);
                setResult(RESULT_OK,intent);
                finish();

            }
        });
    }

//    @Override
//    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
//        super.onActivityResult(requestCode, resultCode, data);
////        if(requestCode==100&&resultCode==RESULT_OK){
////            Place place=Autocomplete.getPlaceFromIntent(data);
////            searchView.setText(place.getAddress());
////            LatLng latLng=place.getLatLng();
////            map.addMarker(new
MarkerOptions().position(latLng).title(searchView.getText().toString()));
////            map.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));
////
////        }
////        else if(resultCode== AutocompleteActivity.RESULT_ERROR){
////            Status status=Autocomplete.getStatusFromIntent(data);
////
Toast.makeText(getApplicationContext(),status.getStatusMessage(),Toast.LENGTH_SHORT).s
how();
////        }
//    }

//    @Override
//    public void onStart() {
//        super.onStart();
//
```

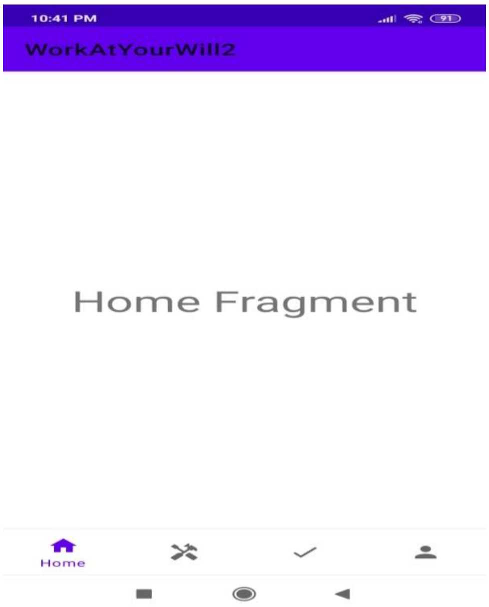
```
//
//    }
//    private boolean checkPermissions() {
//        int permissionState = ActivityCompat.checkSelfPermission(this,
//                Manifest.permission.ACCESS_COARSE_LOCATION);
//        return permissionState == PackageManager.PERMISSION_GRANTED;
//    }
//
//    private void startLocationPermissionRequest() {
//        ActivityCompat.requestPermissions(AddAddress.this,
//                new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
//                REQUEST_PERMISSIONS_REQUEST_CODE);
//    }
//
//    private void requestPermissions() {
//        boolean shouldProvideRationale =
//                ActivityCompat.shouldShowRequestPermissionRationale(this,
//                        Manifest.permission.ACCESS_COARSE_LOCATION);
//
//        // Provide an additional rationale to the user. This would happen if the user denied the
//        // request previously, but didn't check the "Don't ask again" checkbox.
//        if (shouldProvideRationale) {
//            Log.i(TAG, "Displaying permission rationale to provide additional context.");
//
//            showSnackbar(R.string.permission_rationale, android.R.string.ok,
//                    new View.OnClickListener() {
//                        @Override
//                        public void onClick(View view) {
//                            // Request permission
//                            startLocationPermissionRequest();
//                        }
//                    });
//
//        } else {
//            Log.i(TAG, "Requesting permission");
//            // Request permission. It's possible this can be auto answered if device policy
//            // sets the permission in a given state or the user denied the permission
//            // previously and checked "Never ask again".
//            startLocationPermissionRequest();
//        }
//    }

    /**
     * Callback received when a permissions request has been completed.
     */
//    @Override
//    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
//                                           @NonNull int[] grantResults) {
//        Log.i(TAG, "onRequestPermissionResult");
//        if (requestCode == REQUEST_PERMISSIONS_REQUEST_CODE) {
//            if (grantResults.length <= 0) {
```

```
//                // If user interaction was interrupted, the permission request is cancelled and you
//                // receive empty arrays.
//                Log.i(TAG, "User interaction was cancelled.");
//            } else if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
//                // Permission granted.
//                getLastLocation();
//            } else {
//                // Permission denied.
//
//                // Notify the user via a SnackBar that they have rejected a core permission for the
//                // app, which makes the Activity useless. In a real app, core permissions would
//                // typically be best requested during a welcome-screen flow.
//
//                // Additionally, it is important to remember that a permission might have been
//                // rejected without asking the user for permission (device policy or "Never ask
//                // again" prompts). Therefore, a user interface affordance is typically implemented
//                // when permissions are denied. Otherwise, your app could appear unresponsive to
//                // touches or interactions which have required permissions.
//                showSnackbar(R.string.permission_denied_explanation, R.string.settings,
//                        new View.OnClickListener() {
//                            @Override
//                            public void onClick(View view) {
//                                // Build intent that displays the App settings screen.
//                                Intent intent = new Intent();
//                                intent.setAction(
//                                        Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
//                                Uri uri = Uri.fromParts("package",
//                                        BuildConfig.APPLICATION_ID, null);
//                                intent.setData(uri);
//                                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
//                                startActivity(intent);
//                            }
//                        });
//            }
//        }
//    }
//    /**
//     * Shows a {@link Snackbar}.
//     *
//     * @param mainTextStringId The id for the string resource for the Snackbar text.
//     * @param actionStringId   The text of the action item.
//     * @param listener         The listener associated with the Snackbar action.
//     */
//    private void showSnackbar(final int mainTextStringId, final int actionStringId,
//                              View.OnClickListener listener) {
//        Snackbar.make(findViewById(android.R.id.content),
//                getString(mainTextStringId),
//                Snackbar.LENGTH_INDEFINITE)
//                .setAction(getString(actionStringId), listener).show();
//    }
//    @SuppressWarnings("MissingPermission")
//    private void getLastLocation() {
//        mFusedLocationClient.getLastLocation()
//                .addOnCompleteListener(this, new OnCompleteListener<Location>() {
//                    @Override
//                    public void onComplete(@NonNull Task<Location> task) {
//                        if (task.isSuccessful() && task.getResult() != null) {
//                            mLastLocation = task.getResult();
//                            latitude=mLastLocation.getLatitude();
//                            longitude=mLastLocation.getLongitude();
////                            geoFirestore.setLocation(firebaseUser.getUid(), new
GeoPoint(mLastLocation.getLatitude(), mLastLocation.getLongitude()));
//
Toast.makeText(AddAddress.this,mLastLocation.getLatitude()+","+mLastLocation.getLongitud
e(),Toast.LENGTH_SHORT).show();
//
//                        } else {
//                            Log.w(TAG, "getLastLocation:exception",task.getException());
//                            showSnackbar(getString(R.string.no_location_detected));
//                            setResult(RESULT_CANCELED,intent);
//                        }
//                    }
//                });
//    }
//
//    /**
//     * Shows a {@link Snackbar} using {@code text}.
//     *
//     * @param text The Snackbar text.
//     */
//    private void showSnackbar(final String text) {
//        View container = findViewById(R.id.main_activity_container);
//        if (container != null) {
//            Snackbar.make(container, text, Snackbar.LENGTH_LONG).show();
//        }
//    }
    @Override
    public void onMapReady(@NonNull GoogleMap googleMap) {
        googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

        map=googleMap;
        map.setOnMarkerDragListener(new GoogleMap.OnMarkerDragListener() {
            @Override
            public void onMarkerDragStart(Marker marker) {

            }

            @Override
            public void onMarkerDrag(Marker marker) {

            }
            @Override
            public void onMarkerDragEnd(Marker marker) {
                nm=marker;
            }
        });
        map_Ready_flag=true;
//        getLastLocation();
        LatLng latLng = new LatLng(latitude, longitude);
        if(nm!=null){
            nm.remove();
        }

        marker =new MarkerOptions().position(latLng).title("Your current
location").draggable(true);

        nm = map.addMarker(marker);
        map.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));

    }


}
```

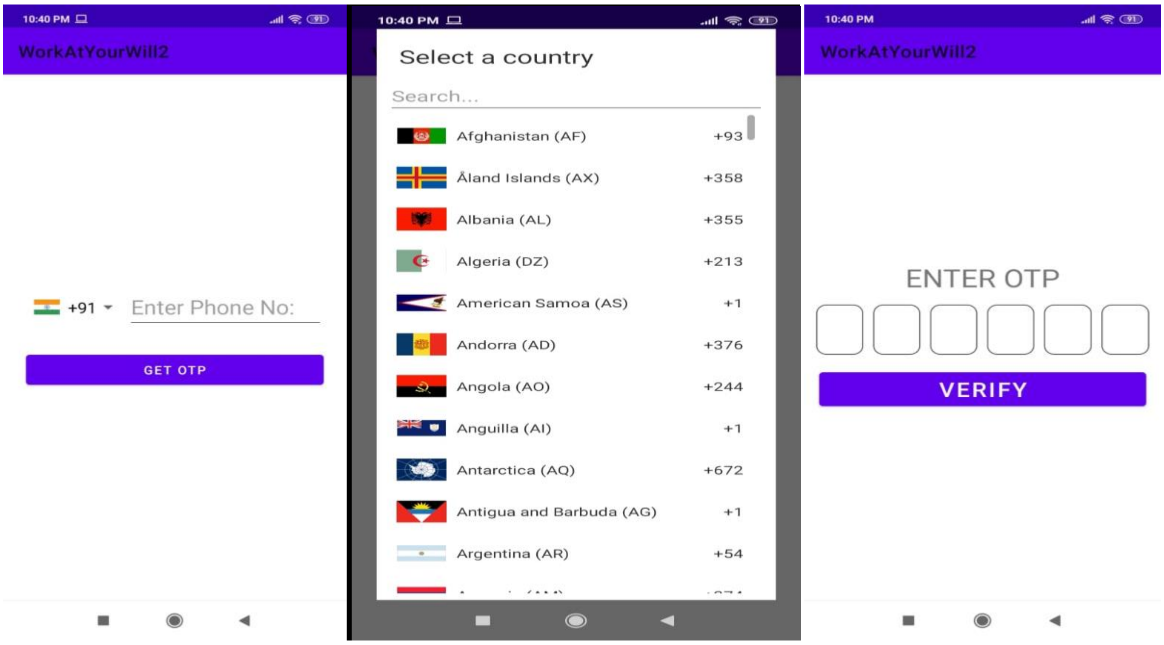
CHAPTER 6

RESULTS

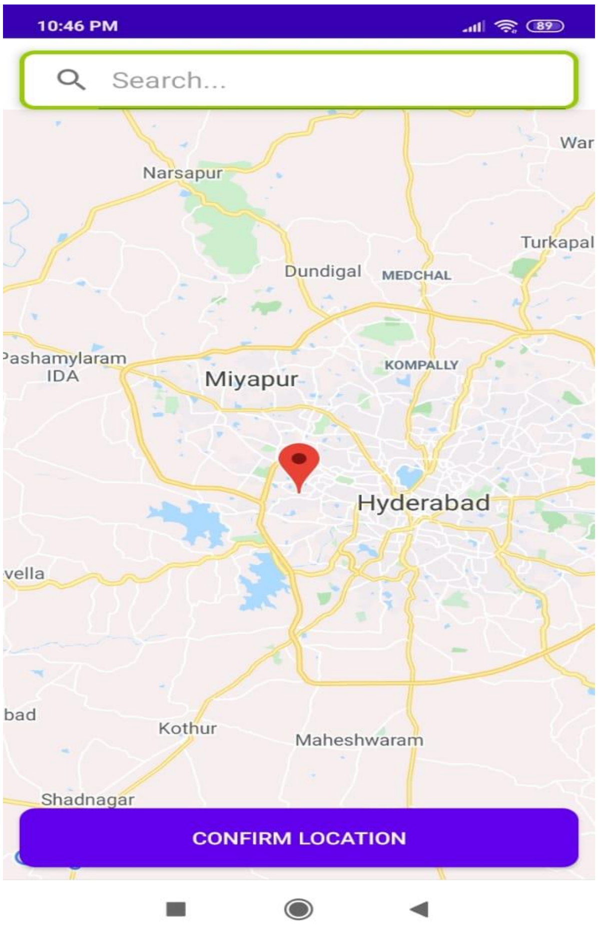
Home Page:



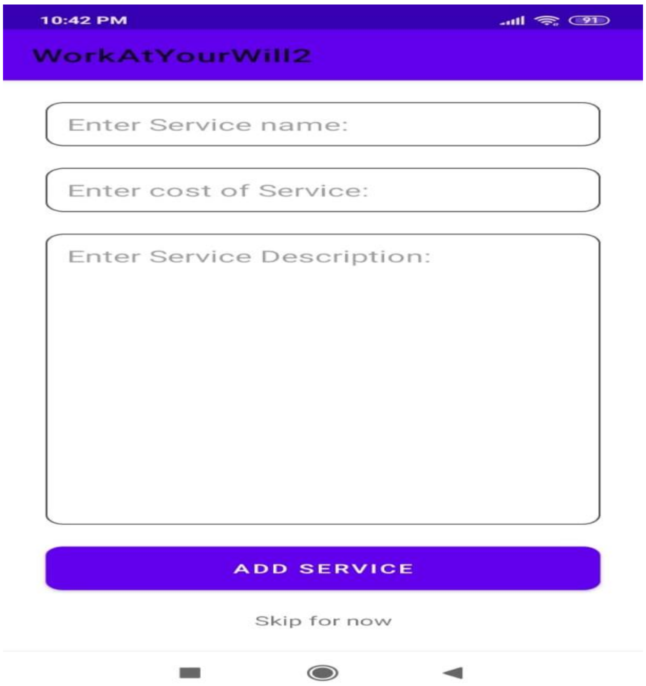
Login Page:



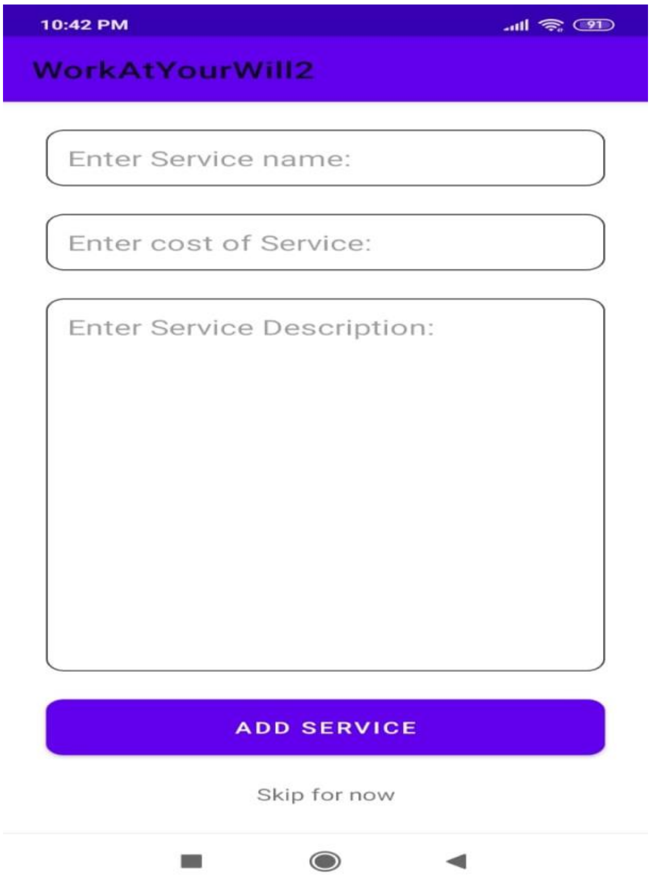
Location Page:



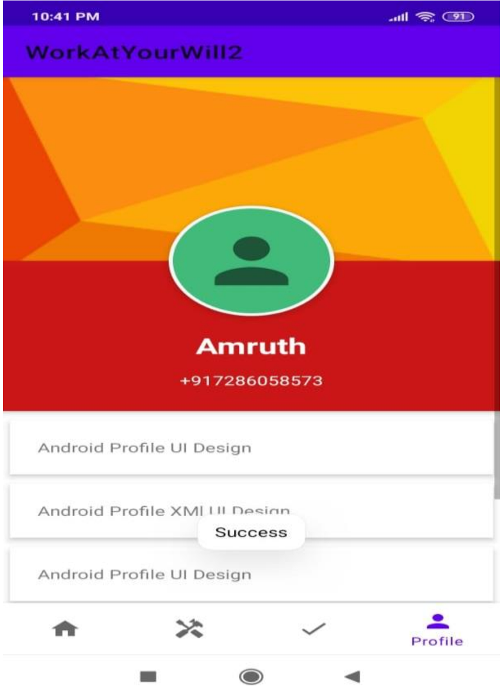
Service detail page :



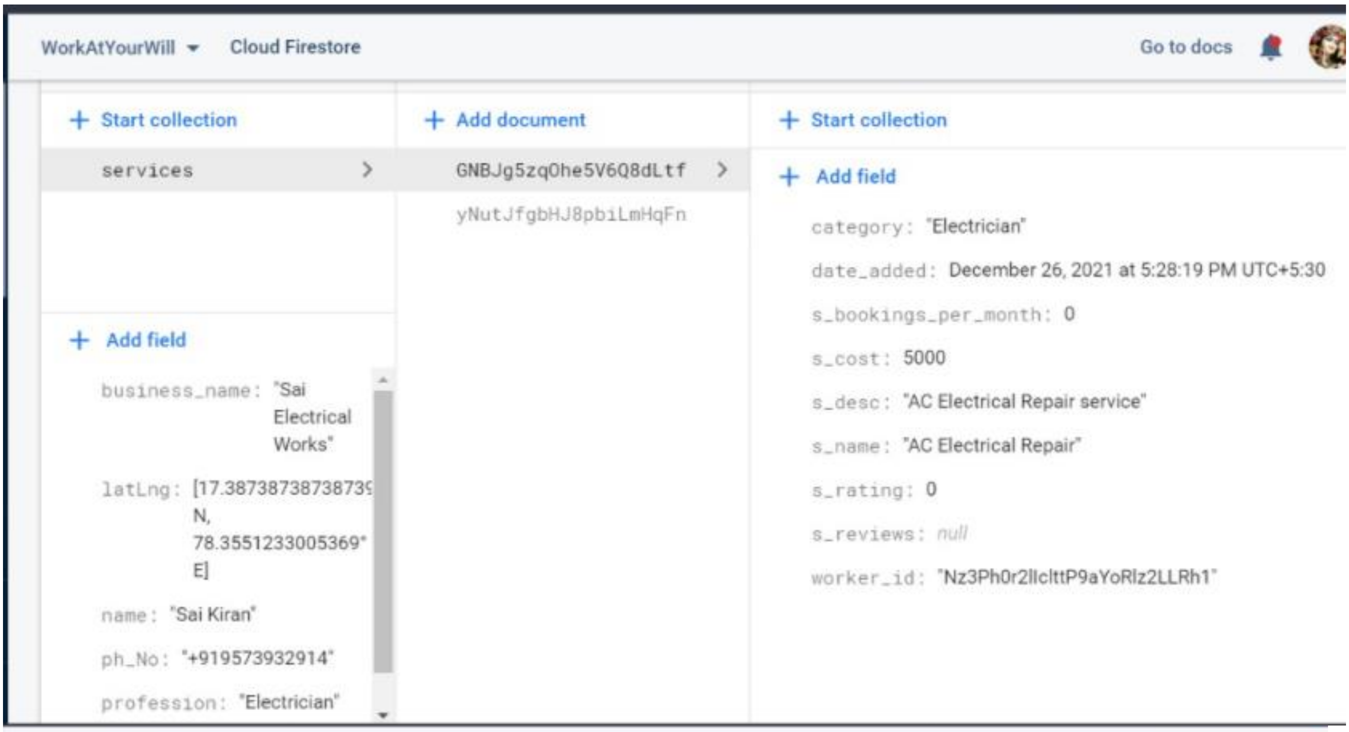
(vi) Add New Service Details :



(viii) **Profile Details :**


10:41 PM
WorkAtYourWill2
Amruth
+917286058573
Android Profile UI Design
Android Profile XML UI Design
Success
Android Profile UI Design
Profile

6.1 FIRESTORE DESIGN:



services

GNBJg5zqOhe5V6Q8dLtf

yNutJfgbHJ8pbiLmHqFn

business_name: "Sai Electrical Works"

latLng: [17.38738738738739 N, 78.3551233005369° E]

name: "Sai Kiran"

ph_No: "+919573932914"

profession: "Electrician"

category: "Electrician"

date_added: December 26, 2021 at 5:28:19 PM UTC+5:30

s_bookings_per_month: 0

s_cost: 5000

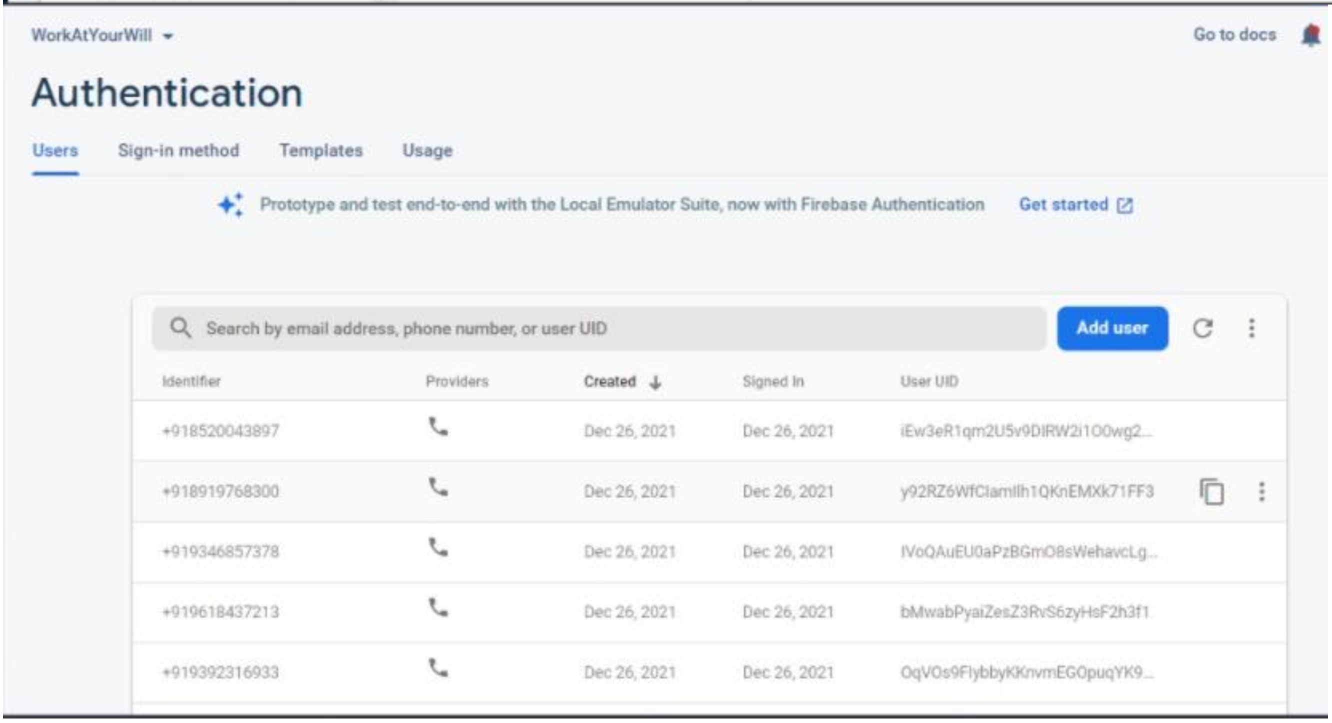
s_desc: "AC Electrical Repair service"

s_name: "AC Electrical Repair"

s_rating: 0

s_reviews: null

worker_id: "Nz3Ph0r2llcltP9aYoRlz2LLRh1"

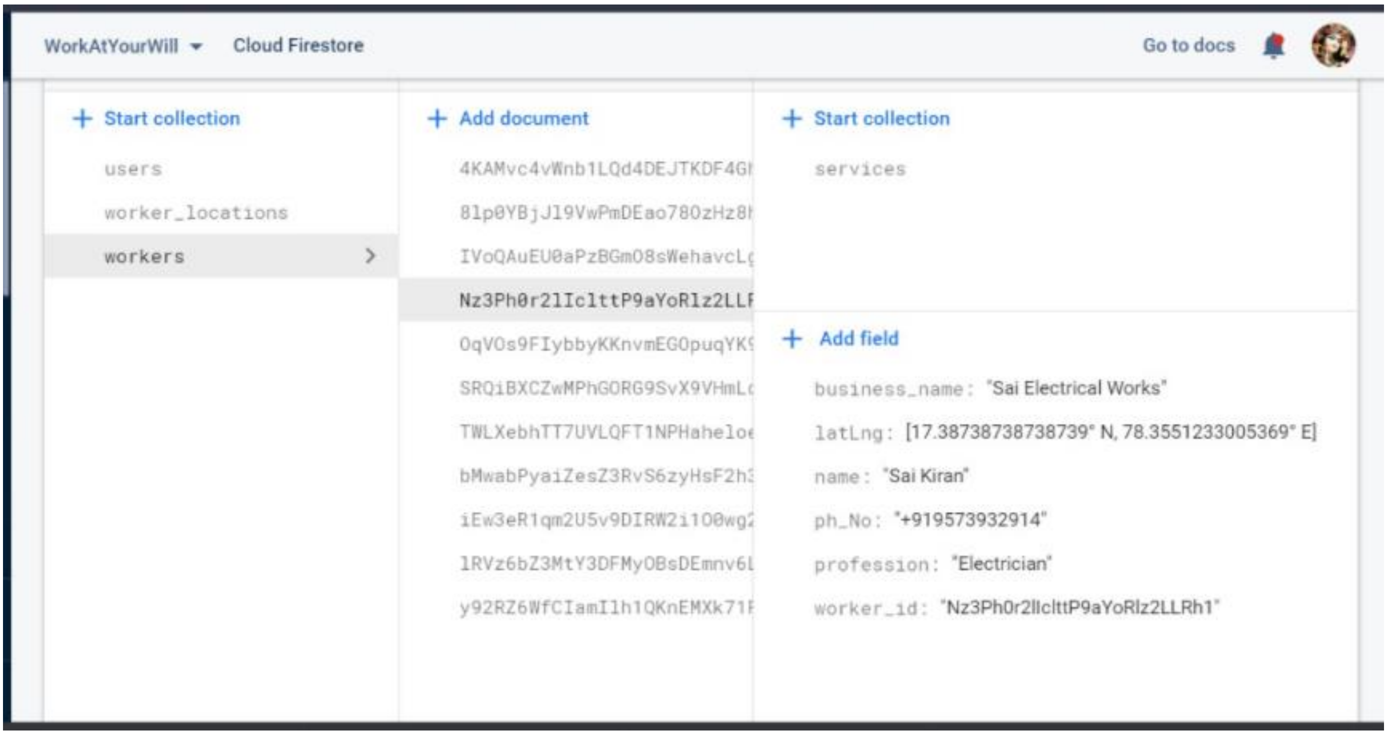


Authentication

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Search by email address, phone number, or user UID

Identifier	Providers	Created ↓	Signed In	User UID
+918520043897	☎	Dec 26, 2021	Dec 26, 2021	iEw3eR1qm2U5v9DIRW2i1O0wg2...
+918919768300	☎	Dec 26, 2021	Dec 26, 2021	y92RZ6WfClamllh1QKnEMXk71FF3
+919346857378	☎	Dec 26, 2021	Dec 26, 2021	lVoQAuEU0aPzBGmO8sWehavcLg...
+919618437213	☎	Dec 26, 2021	Dec 26, 2021	bMwabPyaiZesZ3RvS6zyHsF2h3f1
+919392316933	☎	Dec 26, 2021	Dec 26, 2021	OqVOs9FlybbyKKnvmEGOpuqYK9...



users
worker_locations
workers

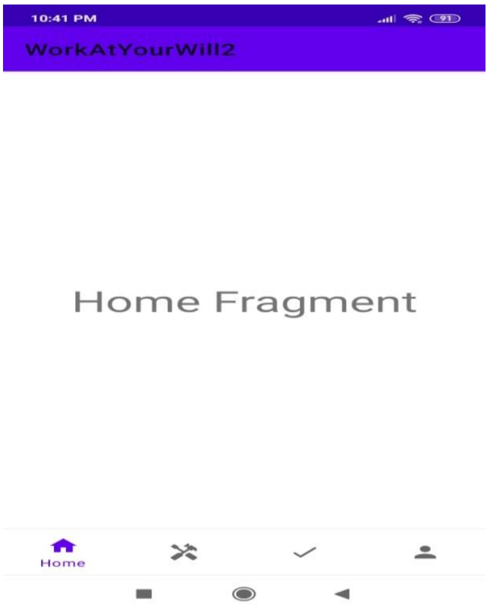
4KAMvc4vWnb1LQd4DEJTKDF4G
81p0YBjJ19VwPmDEao780zHz8
IVoQAuEU0aPzBGm08sWehavcL
Nz3Ph0r2lIclttP9aYoRlz2LL
OqVOs9FIybbyKKnvmEGOpuqYK
SRQiBXCZwMPhGORG9SvX9VHmL
TWLXebhTT7UVLQFT1NPHahelo
bMwabPyaiZesZ3RvS6zyHsF2h
iEw3eR1qm2U5v9DIRW2i1O0wg
lRVz6bZ3MtY3DFMyOBsDEmnv6
y92RZ6WfCIamIlh1QKnEMXk71

services

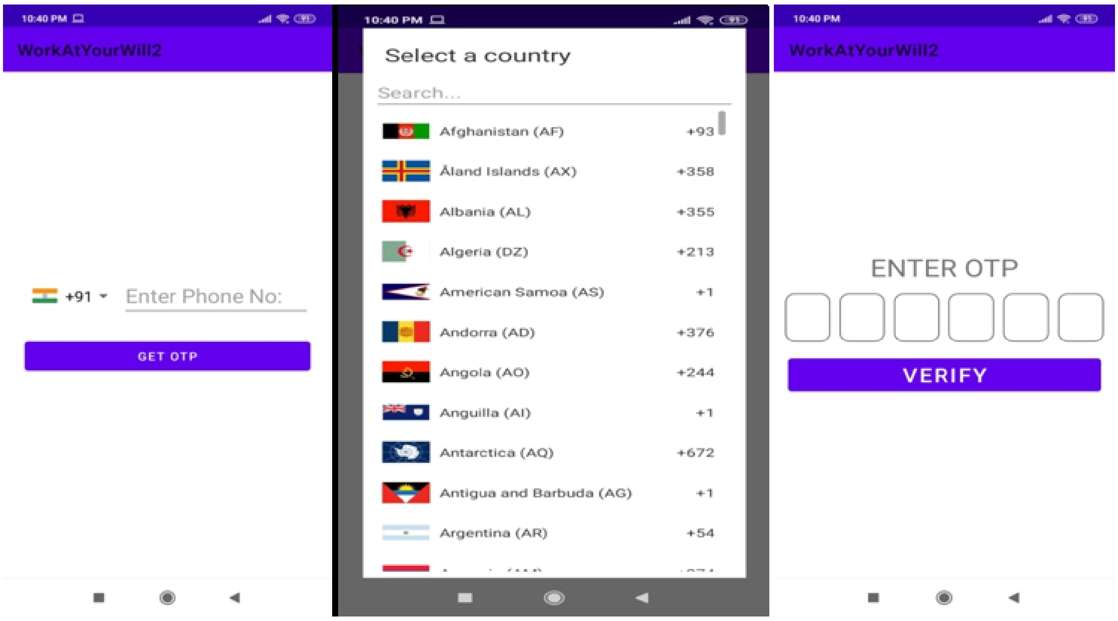
```
business_name: "Sai Electrical Works"
latLng: [17.38738738738739° N, 78.3551233005369° E]
name: "Sai Kiran"
ph_No: "+919573932914"
profession: "Electrician"
worker_id: "Nz3Ph0r2lIclttP9aYoRlz2LLRh1"
```

(User Android Application)

(i) Home Page

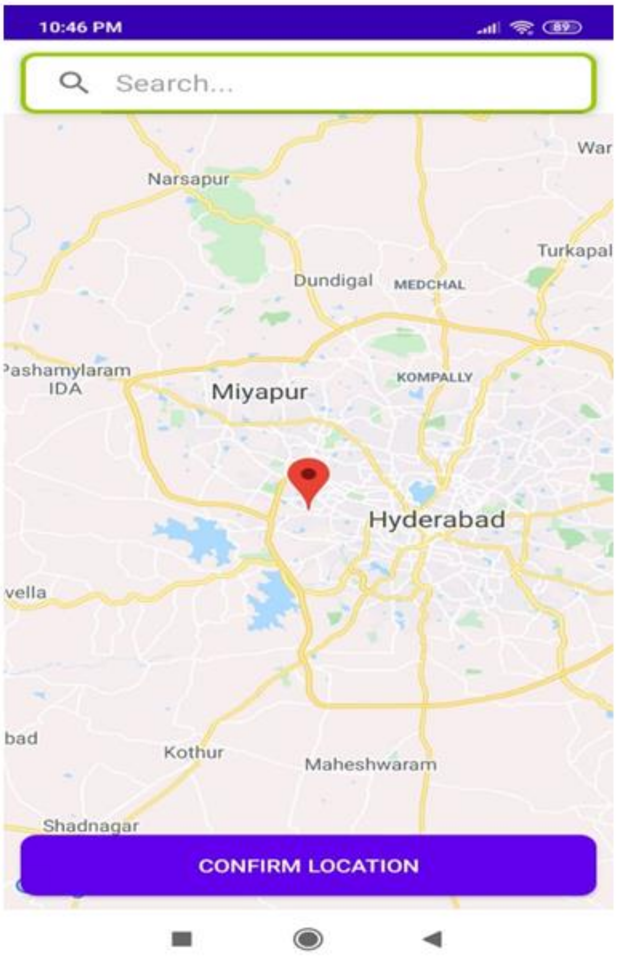


Login Page:

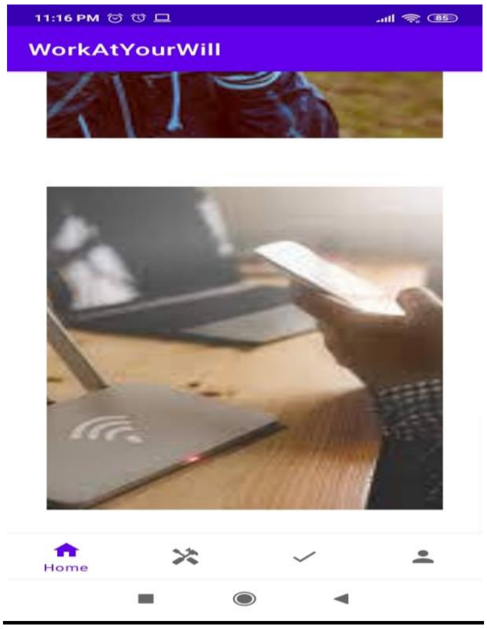


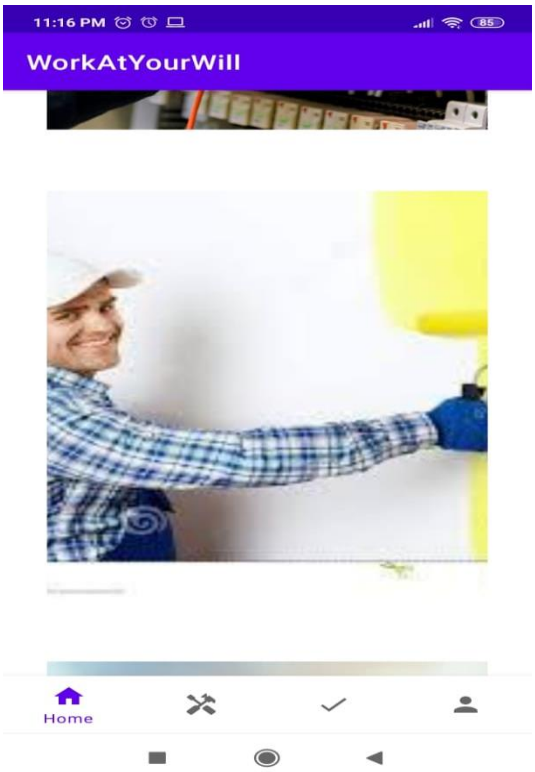
Afghanistan (AF)	+93
Åland Islands (AX)	+358
Albania (AL)	+355
Algeria (DZ)	+213
American Samoa (AS)	+1
Andorra (AD)	+376
Angola (AO)	+244
Anguilla (AI)	+1
Antarctica (AQ)	+672
Antigua and Barbuda (AG)	+1
Argentina (AR)	+54

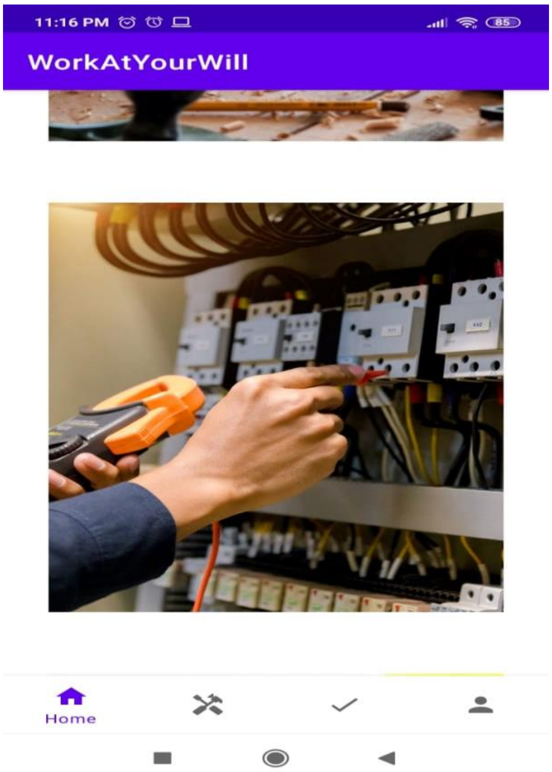
Location Page:

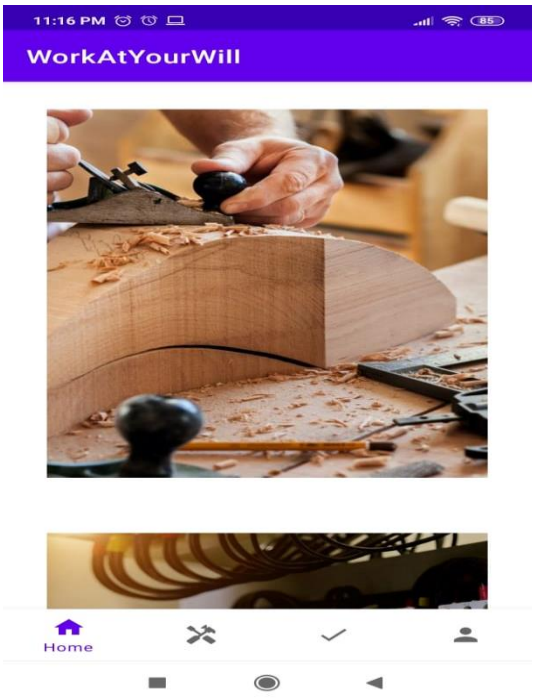


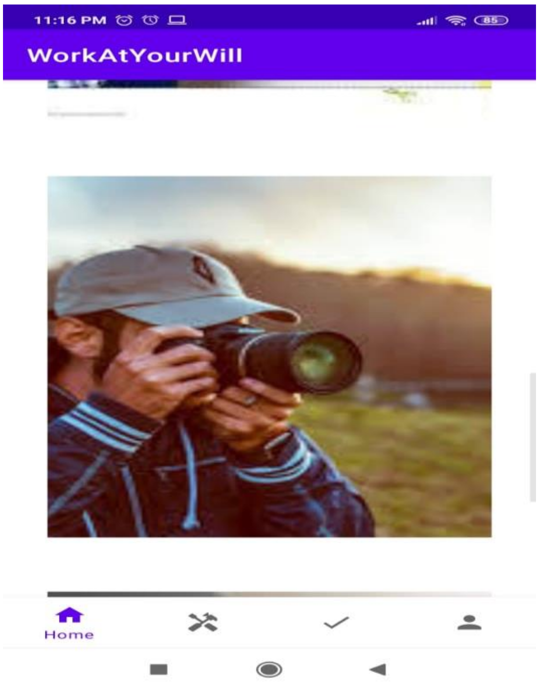
Search Tab for User





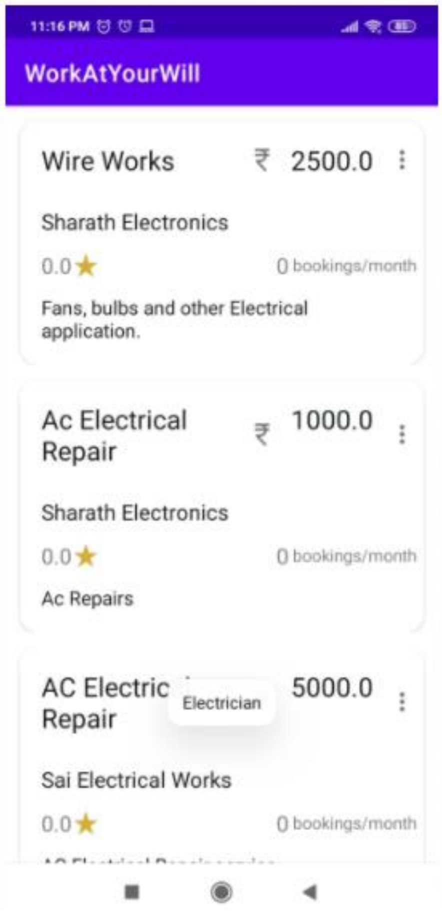






-

Details of Works :



11:16 PM

WorkAtYourWill

Wire Works ₹ 2500.0

Sharath Electronics

0.0★ 0 bookings/month

Fans, bulbs and other Electrical application.

Ac Electrical Repair ₹ 1000.0

Sharath Electronics

0.0★ 0 bookings/month

Ac Repairs

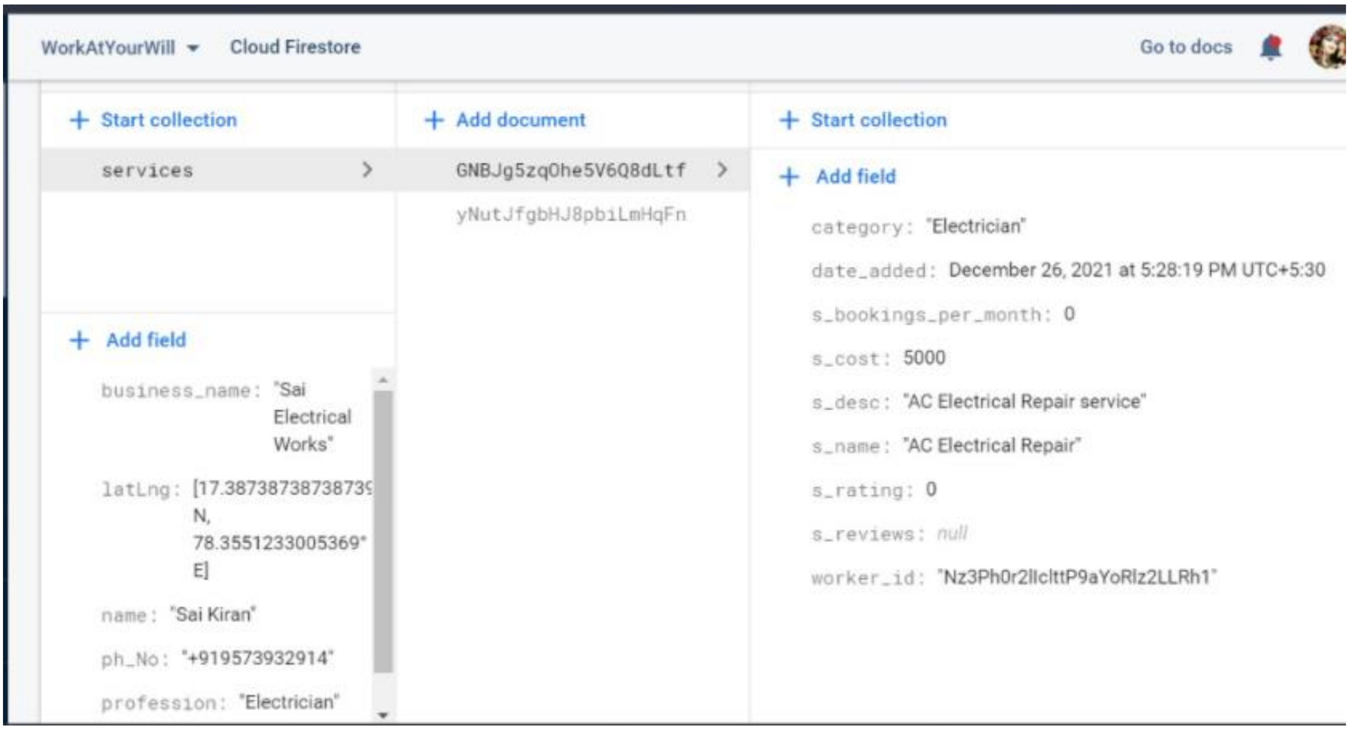
AC Electric Repair 5000.0

Electrician

Sai Electrical Works

0.0★ 0 bookings/month

FIRESTORE DESIGN



services

GNBJg5zqOhe5V6Q8dLtf

yNutJfgbHJ8pbiLmHqFn

business_name: "Sai Electrical Works"

latLng: [17.38738738738739 N, 78.3551233005369° E]

name: "Sai Kiran"

ph_No: "+919573932914"

profession: "Electrician"

category: "Electrician"

date_added: December 26, 2021 at 5:28:19 PM UTC+5:30

s_bookings_per_month: 0

s_cost: 5000

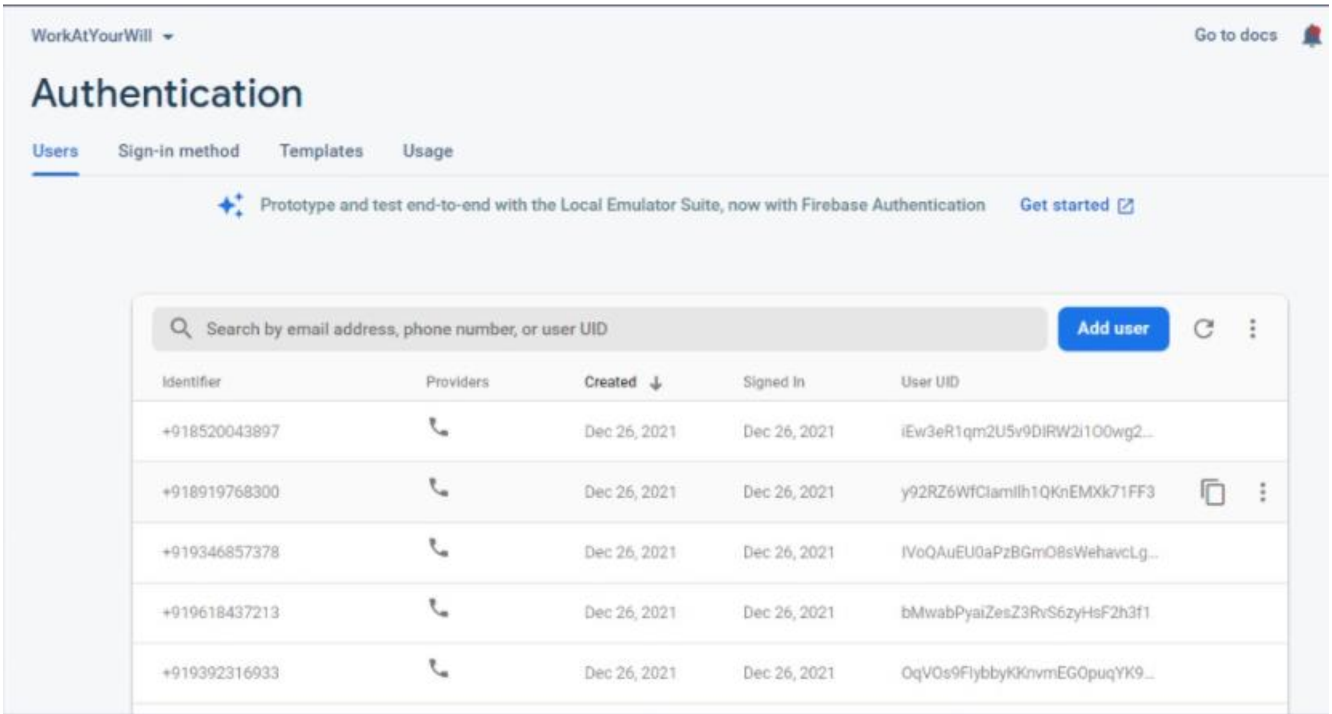
s_desc: "AC Electrical Repair service"

s_name: "AC Electrical Repair"

s_rating: 0

s_reviews: null

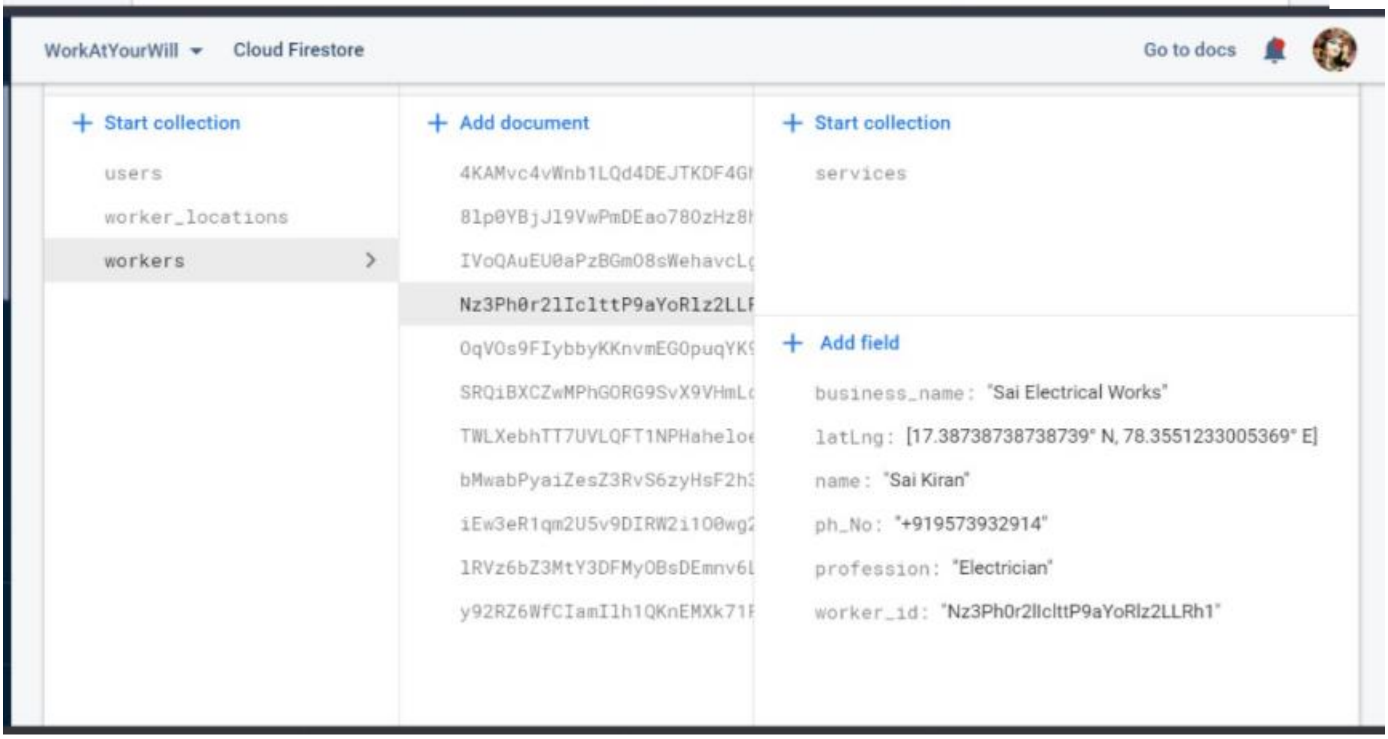
worker_id: "Nz3Ph0r2llcIttP9aYoRlz2LLRh1"



Authentication

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Identifier	Providers	Created ↓	Signed In	User UID
+918520043897	☎	Dec 26, 2021	Dec 26, 2021	iEw3eR1qm2U5v9DIRW2i1O0wg2...
+918919768300	☎	Dec 26, 2021	Dec 26, 2021	y92RZ6WfClamllh1QKnEMXk71FF3
+919346857378	☎	Dec 26, 2021	Dec 26, 2021	IVoQAuEU0aPzBGmO8sWehavcLg...
+919618437213	☎	Dec 26, 2021	Dec 26, 2021	bMwabPyaiZesZ3RvS6zyHsF2h3f1
+919392316933	☎	Dec 26, 2021	Dec 26, 2021	OqVOs9FlybbyKKnvmEGOpuqYK9...



users
worker_locations
workers

4KAMvc4vWnb1LQd4DEJTKDF4G
81p0YBjJ19VwPmDEao780zHz8
IVoQAuEU0aPzBGmO8sWehavcL
Nz3Ph0r21IclttP9aYoR1z2LLF
OqVOs9FIybbyKKnvmEGOpuqYK
SRQiBXCZwMPhGORG9SvX9VHmL
TWLXebhTT7UVLQFT1NPHahelo
bMwabPyaiZesZ3RvS6zyHsF2h
iEw3eR1qm2U5v9DIRW2i100wg
1RVz6bZ3MtY3DFMyOBsDEmnv6
y92RZ6WfCIamIlh1QKnEMXk71F

services

business_name: "Sai Electrical Works"
latLng: [17.38738738738739° N, 78.3551233005369° E]
name: "Sai Kiran"
ph_No: "+919573932914"
profession: "Electrician"
worker_id: "Nz3Ph0r2llclttP9aYoRlz2LLRh1"

CHAPTER 7

TESTING

Unit Testing: This is the lowest level of testing that is conducted to remove syntax & logic errors from a single unit. Individual components are used to ensure that they operate correctly. Each component is tested independently, without other system components.

Module Testing: A module is a collection of dependent components such as an object class, an abstract data type or some looser collection of procedures and functions. A module encapsulates related components, so can be tested without other system modules.

System Testing: The sub-systems are integrated to make up the system. The system as a complete entity is tested over here. This process is concerned with finding errors that result from unanticipated interactions between sub-systems. It is also concerned with validating that the system meets its functional and non-functional requirements and testing the emerged system properties.

Accepting Testing: This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system customer rather than simulated test data. Acceptance testing may reveal errors and omissions in the system requirements defined because the real data exercise the system in different ways from test data. It may also reveal requirements problems where the system ‘s facilities do not really meet the user’s needs or the system performance is unacceptable.

CHAPTER 8

CONCLUSION:

We have successfully developed a secure, user-friendly Android application. This system is capable of searching service providers.

It is developed in Android Studio. The user performs a search for the service providers listings by putting either City/State in the search textbox. The business logic tier communicates with the Firestore tier requesting the results of the query sent by it. The results obtained by the Firestore are displayed on the data grid, by refreshing the grid rather than refreshing the entire Android page.

This system will definitely reduce the time to search for a workers to have it for by having all the service providers online.

FUTURE SCOPE:

This project has built as per the basic requirements of a user. But as time goes on there will be change in the requirements of the user.

For the further development of the system an mobile application for Android or iOS can be developed which can make use of the GPS built into the mobile devices to enable service providers to navigate to the location of the user more easily, and also a location based recommendation system can be implemented.

The document verification process will be done in an advanced manner so that any kind of fraud attempts will be reduced to zero. The future verification process will include physical confirmation of the property by our employee and detailed verification of the site documents so that the users will have complete trust.

As of now this project is using the google maps to locate the property, but in future we are planning to have a separate interface in which the location is also navigated in our Android Application itself.

We are also working on to make a schedule the works by the users

CHAPTER 9

REFERENCES

- https://www.javatpoint.com/dbms-tutorial
- https://docs.oracle.com/javase/8/docs/api/
- **https://developer.android.com/guide/topics/resources/runtime**