DCN Assignment

Finding Location Trail of a device

1. Introduction

We can get the location of a device by knowing its GPS coordinates. GPS stands for Global Positioning System**.** GPS is a satellite navigation system used to determine the ground position of an object. The coordinates are expressed in latitude and longitude format. Today, GPS receivers are included in many commercial products, such as automobiles, smartphones, exercise watches, etc. The data from the GPS receivers can be used to get the location trail of a device.

2. Code Snippet

Here is code for fetching the GPS coordinates of a device to determine its current location and displaying it on Google Maps using Google Map API. These GPS coordinates can be stored for future processing in other location based services.

All codes are written in java language using Android Studio IDE.

Class – MainActivity.java

```
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.LocationManager;
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.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private FirebaseUser mCurrentUser;
    private static final int LOCATION_PERMISSION_REQUESTCODE = 999;
    private static final int STORAGE_PERMISSION_REQUESTCODE = 0;
    EditText phoneNumber;
    Button startTracking;
    Button mLogoutBtn;
    TextView welcome;


    public void onClickActivity(View v){
        Intent i = new Intent(v.getContext(), MapsActivity.class);
        startActivity(i);
    }

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initializing
        mAuth = FirebaseAuth.getInstance();
        mCurrentUser = mAuth.getCurrentUser();
        mLogoutBtn = findViewById(R.id.logout);
        welcome = findViewById(R.id.welcome);

        mLogoutBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mAuth.signOut();
                Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);
                startActivity(loginIntent);
                finish();
            }
        });

        // Getting required permissions
        getPermissions();

        //requesting location updates for location listener
        LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

        if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)){
            Toast.makeText(this, "GPS is Enabled in your device", Toast.LENGTH_SHORT).show();
        }else{
            // GPS not enabled
            Toast.makeText(this, "GPS is not enabled in your device",
Toast.LENGTH_SHORT).show();
            showGPSDisabledAlertToUser();
        }


    }

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

        if(mCurrentUser == null){
            Intent loginIntent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(loginIntent);
            finish();
        }else{
            welcome.setText("Welcome, " + mCurrentUser.getPhoneNumber());
        }
    }

    private  void getPermissions() {

        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            // Permission is not granted
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_PERMISSION_REQUESTCODE);
        }

        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE)
                != PackageManager.PERMISSION_GRANTED) {
            // Permission is not granted
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_REQUESTCODE);
        }

    }

    private void showGPSDisabledAlertToUser(){
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
        alertDialogBuilder.setMessage("GPS is disabled in your device please enable it.")
                .setCancelable(false)
                .setPositiveButton("Goto Settings Page To Enable GPS",
```

```
                        new DialogInterface.OnClickListener(){
                            public void onClick(DialogInterface dialog, int id){
                                Intent callGPSSettingIntent = new
Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                                startActivity(callGPSSettingIntent);
                            }
                        });

        alertDialogBuilder.setNegativeButton("Cancel",
                new DialogInterface.OnClickListener(){
                    public void onClick(DialogInterface dialog, int id){
                        dialog.cancel();
                    }
                });

        AlertDialog alert = alertDialogBuilder.create();
        alert.show();
    }
}
```

Class – LoginActivity.java

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

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;

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

import java.util.concurrent.TimeUnit;

public class LoginActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private FirebaseUser mCurrentUser;

    private PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks;

    private EditText mPhoneNumber;
    private Button mGenerateBtn;
    private ProgressBar mLoginProgress;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        mAuth = FirebaseAuth.getInstance();
        mCurrentUser = mAuth.getCurrentUser();

        mPhoneNumber = findViewById(R.id.phoneNumber);
        mGenerateBtn = findViewById(R.id.generateOTP);
        mLoginProgress = findViewById(R.id.progressBar);

        mGenerateBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phone_number = mPhoneNumber.getText().toString();
                if(!isValidNumber(phone_number)) {
                    mPhoneNumber.setError("Enter valid number!");
                } else {
                    mLoginProgress.setVisibility(View.VISIBLE);
                    mGenerateBtn.setEnabled(false);
```

```
                    PhoneAuthProvider.getInstance().verifyPhoneNumber(
                            "+91" + phone_number,        // Phone number to verify
                            60,                 // Timeout duration
                            TimeUnit.SECONDS,   // Unit of timeout
                            LoginActivity.this,               // Activity (for callback
binding)
                            mCallbacks);        // OnVerificationStateChangedCallbacks
                }
            }
        });

        mCallbacks = new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
            @Override
            public void onVerificationCompleted(@NonNull PhoneAuthCredential
phoneAuthCredential) {
                signInWithPhoneAuthCredential(phoneAuthCredential);
            }

            @Override
            public void onVerificationFailed(@NonNull FirebaseException e) {
                mPhoneNumber.setError("Please Try Again!");
                mLoginProgress.setVisibility(View.INVISIBLE);
                mGenerateBtn.setEnabled(true);
            }

            @Override
            public void onCodeSent(@NonNull String s, @NonNull
PhoneAuthProvider.ForceResendingToken forceResendingToken) {
                super.onCodeSent(s, forceResendingToken);

                Intent otpIntent = new Intent(LoginActivity.this, OtpActivity.class);
                otpIntent.putExtra("VerificationID", s);
                startActivity(otpIntent);
            }
        };
    }

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

        if(mCurrentUser != null){
            sendUserToMain();
        }
    }

    private boolean isValidNumber(String phone) {

        return android.util.Patterns.PHONE.matcher(phone).matches() && phone.length() == 10;

    }

    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            FirebaseUser user = task.getResult().getUser();
                            sendUserToMain();
                        } else {
                            // Sign in failed, display a message and update the UI
                            if (task.getException() instanceof
FirebaseAuthInvalidCredentialsException) {
                                // The verification code entered was invalid
                                mPhoneNumber.setError("Error in Verifying Phone Number!");
                            }
                        }
                        mLoginProgress.setVisibility(View.INVISIBLE);
                        mGenerateBtn.setEnabled(true);
                    }
                });
    }

    public void sendUserToMain(){
        Intent mainIntent = new Intent(LoginActivity.this, MainActivity.class);
```

```
        startActivity(mainIntent);
        finish();
    }
}
```

Class – OtpActivity.java

```
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.ProgressBar;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;

public class OtpActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private FirebaseUser mCurrentUser;

    private String mAuthVerificationID;

    private EditText mOtp;
    private Button mVerifyBtn;
    private ProgressBar mProgBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_otp);

        mAuth = FirebaseAuth.getInstance();
        mCurrentUser = mAuth.getCurrentUser();

        mOtp = findViewById(R.id.otp);
        mVerifyBtn = findViewById(R.id.verifyOtp);
        mProgBar = findViewById(R.id.otp_progressBar);

        mAuthVerificationID = getIntent().getStringExtra("VerificationID");

        mVerifyBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String otp = mOtp.getText().toString();
                if(otp.isEmpty()){
                    mOtp.setError("Enter Valid OTP!");
                }else{
                    mProgBar.setVisibility(View.VISIBLE);
                    mVerifyBtn.setEnabled(false);
                    PhoneAuthCredential credential =
PhoneAuthProvider.getCredential(mAuthVerificationID, otp);
                    signInWithPhoneAuthCredential(credential);
                }
            }
        });
    }

    private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(OtpActivity.this, new OnCompleteListener<AuthResult>()
{
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            FirebaseUser user = task.getResult().getUser();
```

```
                            sendUserToMain();
                        } else {
                            // Sign in failed, display a message and update the UI
                            if (task.getException() instanceof
FirebaseAuthInvalidCredentialsException) {
                                // The verification code entered was invalid
                                mOtp.setError("Invalid Code!");
                            }
                        }
                        mProgBar.setVisibility(View.INVISIBLE);
                        mVerifyBtn.setEnabled(true);
                    }
                });
    }

    protected void onStart() {
        super.onStart();
        if(mCurrentUser != null) {
            sendUserToMain();
        }
    }

    public void sendUserToMain(){
        Intent mainIntent = new Intent(OtpActivity.this, MainActivity.class);
        startActivity(mainIntent);
        finish();
    }
}
```

Class - MapsActivity.java

```
import androidx.fragment.app.FragmentActivity;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private static final long MIN_TIME = 10000 ;
    private static final float MIN_DISTANCE = 10;
    private GoogleMap mMap;

    private ArrayList<LatLng> locationTrails;
    private ArrayList<String> timeStamps;

    private LocationListener locationListener;
    private LocationManager locationManager;

    PolylineOptions polylineOptions;
    Polyline polyline;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

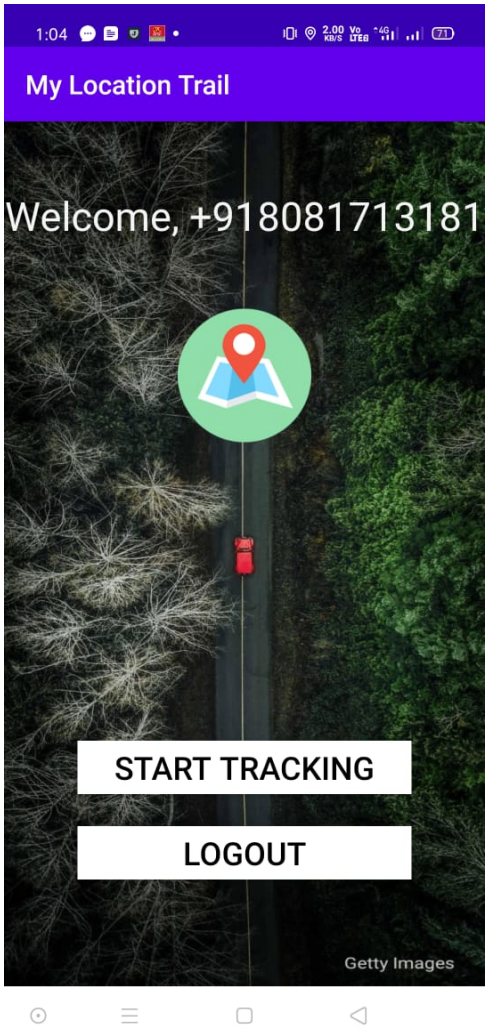
```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        locationTrails = new ArrayList<>();
        timeStamps = new ArrayList<>();
    }


    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this
case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to
install
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        //requesting location updates for location listener
        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);

        if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)){
            Toast.makeText(this, "GPS is Enabled in your device", Toast.LENGTH_SHORT).show();
        }else{
            // GPS not enabled
            Toast.makeText(this, "GPS is not enabled in your device",
Toast.LENGTH_SHORT).show();
            showGPSDisabledAlertToUser();
        }

        polylineOptions = new PolylineOptions()
                .addAll(locationTrails).clickable(true);
        polyline = googleMap.addPolyline(polylineOptions);

        locationListener = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                drawOnMap(location);
            }

            @Override
            public void onStatusChanged(String provider, int status, Bundle extras) {

            }

            @Override
            public void onProviderEnabled(String provider) {

            }

            @Override
            public void onProviderDisabled(String provider) {

            }
        };

        //setting location update time and distance
        try {
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, MIN_TIME,
MIN_DISTANCE, locationListener);

            Location startLocation =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
            drawOnMap(startLocation);

        } catch (SecurityException | NullPointerException e) {
            e.printStackTrace();
        }
```

```
    }

    private void drawOnMap(Location location){
        LatLng updated = new LatLng(location.getLatitude(), location.getLongitude());
        locationTrails.add(updated);

        //adding marker
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        String timeStamp  = dateFormat.format(new Date());
        timeStamps.add(timeStamp);
        mMap.clear();
        mMap.addMarker(new MarkerOptions().position(locationTrails.get(0)).title("Start: " +
timeStamps.get(0)).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GRE
EN)));
        mMap.addMarker(new MarkerOptions().position(locationTrails.get(locationTrails.size()-
1)).title("End: " + timeStamps.get(timeStamps.size()-
1)).icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED)));
        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(updated,18.0f));

        //adding new polyline
        polylineOptions.add(updated);
        polyline = mMap.addPolyline(polylineOptions);
        polyline.setColor(Color.rgb(74,137,243));

        //save externally updated location
        String toCSV = location.getLatitude() + "," + location.getLongitude() + "," +
location.getTime() + "\n";
        exportToCSV(toCSV);
    }

    private void showGPSDisabledAlertToUser(){
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
        alertDialogBuilder.setMessage("GPS is disabled in your device please enable it.")
                .setCancelable(false)
                .setPositiveButton("Goto Settings Page To Enable GPS",
                        new DialogInterface.OnClickListener(){
                            public void onClick(DialogInterface dialog, int id){
                                Intent callGPSSettingIntent = new
Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                                startActivity(callGPSSettingIntent);
                            }
                        });

        alertDialogBuilder.setNegativeButton("Cancel",
                new DialogInterface.OnClickListener(){
                    public void onClick(DialogInterface dialog, int id){
                        dialog.cancel();
                    }
                });

        AlertDialog alert = alertDialogBuilder.create();
        alert.show();
    }

    private void exportToCSV(String loca){
        File file = new File(getExternalCacheDir(), "Locations.csv");
        try {
            FileWriter fw = new FileWriter(file, true);
            fw.write(loca);
            fw.close();
        }catch (IOException e){
            e.printStackTrace();
        }
    }
}
```

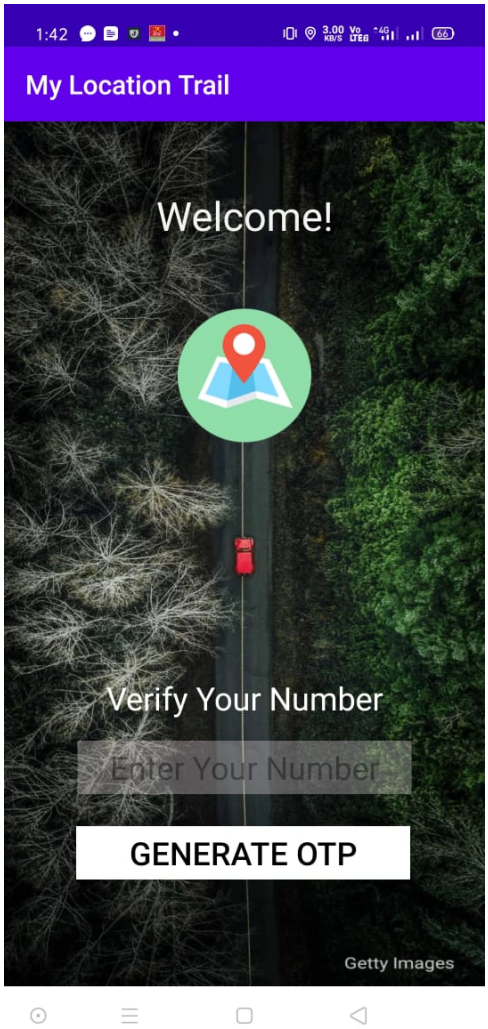
3. App Demo

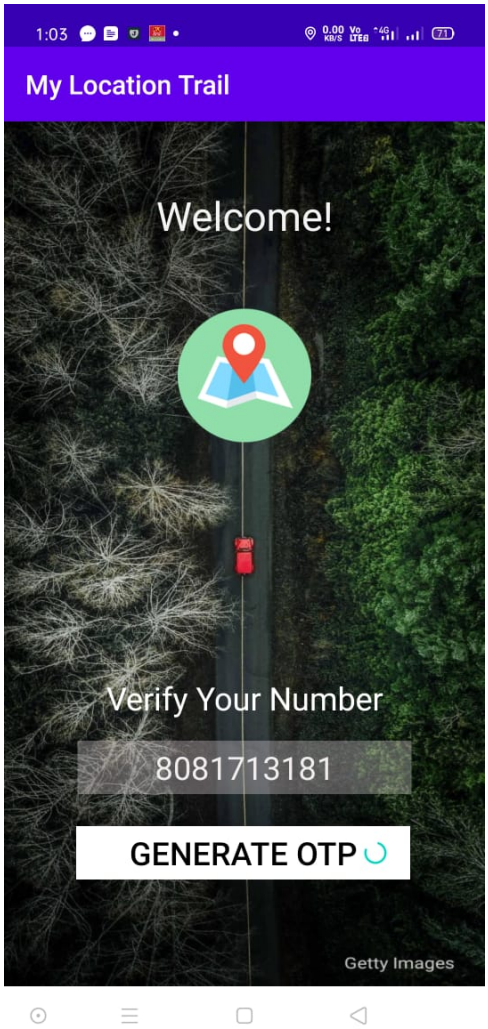
1. On the first window when the app opens, it checks whether any user is logged in or not.

2. If some user is logged in then it display “Welcome, <phone_number>”.



3. If no one is logged in takes us to the login page.

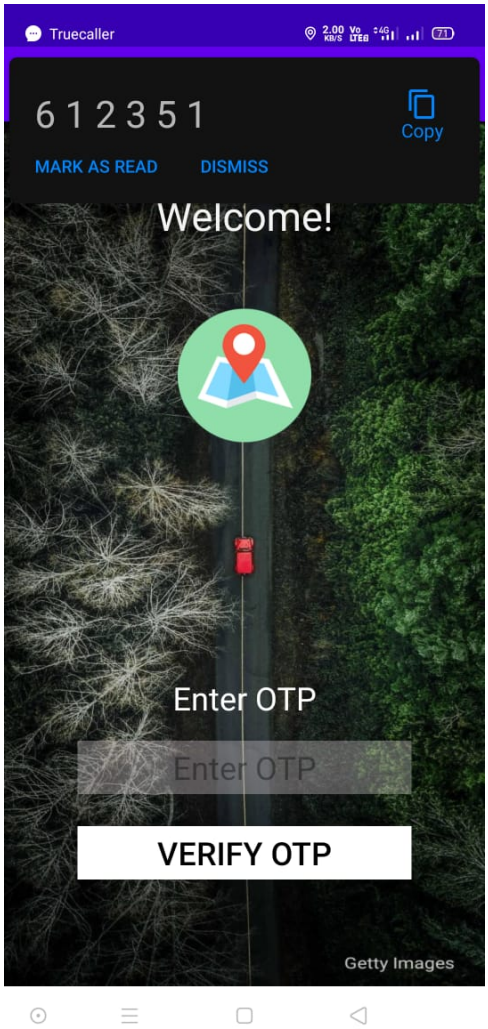


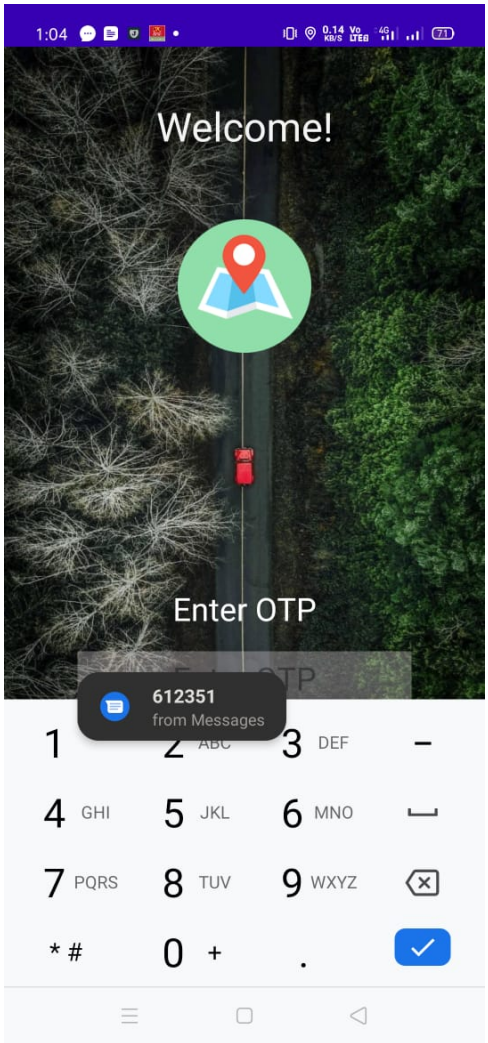


4. On Login page it asks us to enter the number and click on ‘Generate OTP’ button to generate the OTP for user verification.

5. After entering the number when the button is clicked, the app checks whether the number is valid or not. If the number is invalid, it displays an invalid input error message and asks user to re-enter the number.

6. If the entered number is valid, the app send an OTP request to Firebase Server and waits for response. If the OTP request is sent successfully, the OTP input window is opened. Otherwise it asks user to retry.





7. When the OTP is received, the app suggest us to autofill the OTP.

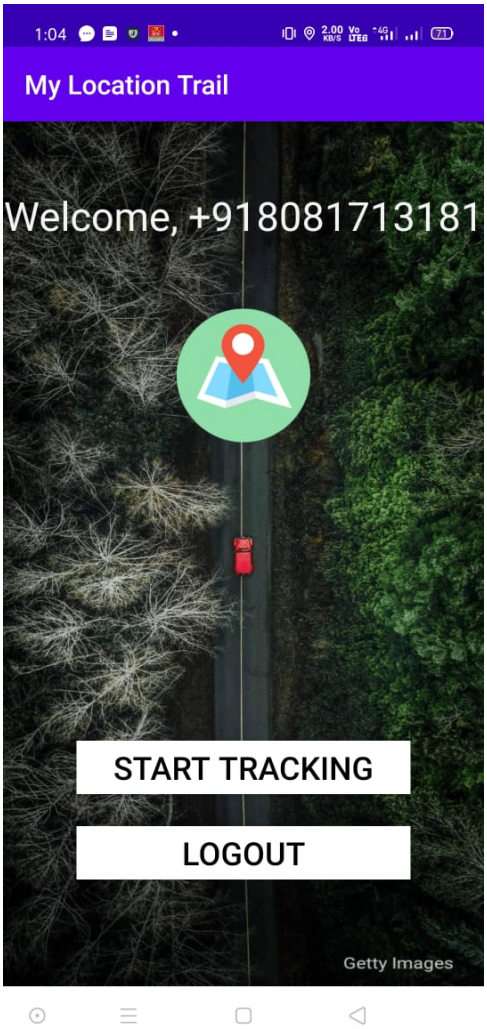
8. On clicking the autofill popup, we can automatically fetch the OTP from the messages.

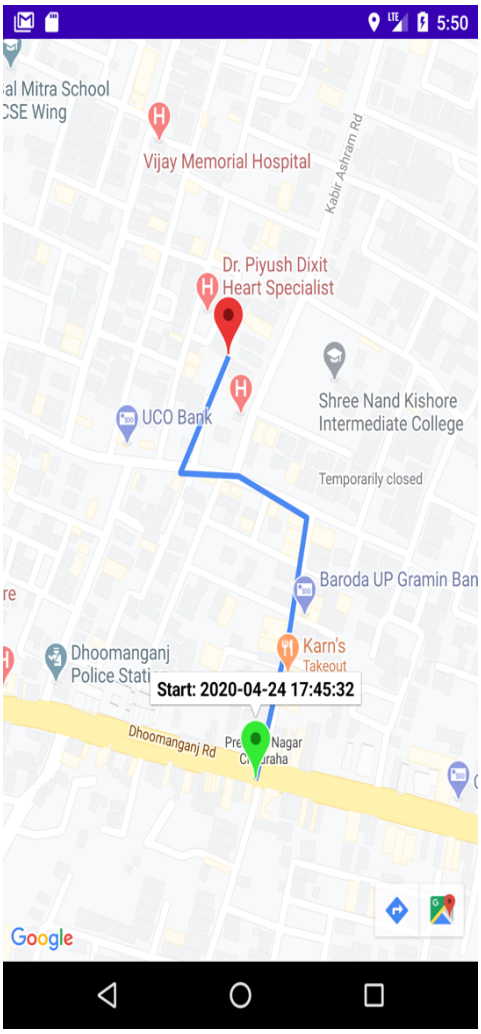
9. On entering the OTP and we need to click on 'Verify OTP' button.

10. When the button is clicked the app checks if the OTP is valid from the Firebase Server.

11. If the OTP is valid, the window in step 2 opens(Main window) , else the app asks us to enter a valid OTP.

    Note: If the app is closed after login and reopened again after some time, the previous logged in user is retained unless the user is logged out.





12. If we want to start the tracking, we need to click on the respective button and the Google Maps window will open. Else if we want to logout from the current user we can click on respective button, and the app will take us back to login page.

13. The maps window opens after pressing the 'Start Tracking' button and the app starts to track the location of the device and drawing the location trail.

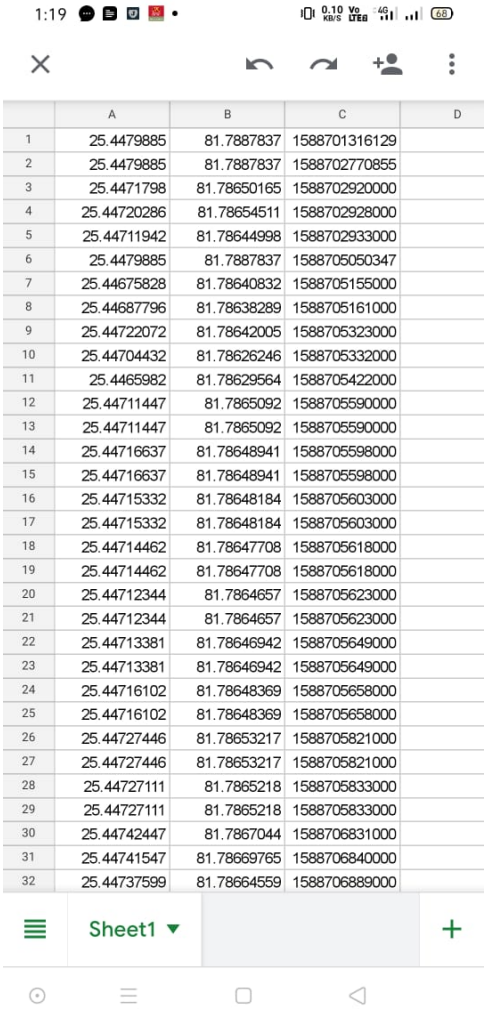
14. The map is shown using the 'Google Map API' provided by google.

15. The 'Green Marker' in the image shows the starting point.

16. The 'Red Marker' in the image shows the end point.

17. The 'Blue Line' show the location trail between the starting point and the ending point.

18. The location coordinate is saved each time the location of the device gets updated to the internal storage as Comma Separated-Values(.csv) file in the app's cache director.



	A	B	C	D
1	25.4479885	81.7887837	1588701316129	
2	25.4479885	81.7887837	1588702770855	
3	25.4471798	81.78650165	1588702920000	
4	25.44720286	81.78654511	1588702928000	
5	25.44711942	81.78644998	1588702933000	
6	25.4479885	81.7887837	1588705050347	
7	25.44675828	81.78640832	1588705155000	
8	25.44687796	81.78638289	1588705161000	
9	25.44722072	81.78642005	1588705323000	
10	25.44704432	81.78626246	1588705332000	
11	25.4465982	81.78629564	1588705422000	
12	25.44711447	81.7865092	1588705590000	
13	25.44711447	81.7865092	1588705590000	
14	25.44716637	81.78648941	1588705598000	
15	25.44716637	81.78648941	1588705598000	
16	25.44715332	81.78648184	1588705603000	
17	25.44715332	81.78648184	1588705603000	
18	25.44714462	81.78647708	1588705618000	
19	25.44714462	81.78647708	1588705618000	
20	25.44712344	81.7864657	1588705623000	
21	25.44712344	81.7864657	1588705623000	
22	25.44713381	81.78646942	1588705649000	
23	25.44713381	81.78646942	1588705649000	
24	25.44716102	81.78648369	1588705658000	
25	25.44716102	81.78648369	1588705658000	
26	25.44727446	81.78653217	1588705821000	
27	25.44727446	81.78653217	1588705821000	
28	25.44727111	81.7865218	1588705833000	
29	25.44727111	81.7865218	1588705833000	
30	25.44742447	81.7867044	1588706831000	
31	25.44741547	81.78669765	1588706840000	
32	25.44737599	81.78664559	1588706889000	

(File path:- "/storage/Android/data/com.example.mylocationtrail/cache/Locations.csv")

----------END----------

Submitted By:-

Rishi Raj Chaurasia
Roll no. 31