

steps to create and link firebase with gps.

step 1- creating a firebase project and add a user Authenticating via email/password(done>>.email = Rector971120 password 971120)

step 2-Connecting the app to the firebase(android studio)

>Create a new Android project and then open the Firebase Assistant, by selecting Tools > Firebase from the Android Studio toolbar.

>In the Firebase Assistant, expand the "Authentication" section and then select "Email and password authentication." email = Rector971120 password 971120

>Select "Connect to Firebase."

>In the subsequent dialog, select "Choose an existing Firebase or Google project."

>Select the project that you just created, and then click "Connect to Firebase."(Accident Alert application)

>Click "Add Firebase Authentication to your app."

>Check the subsequent popup, then click "Accept Changes."

>Once the Firebase Assistant is displaying the "Dependencies set up correctly" message, exit this part of the Assistant by clicking the little backwards-arrow in its upper-left corner.

>Expand the "Realtime Database" section, and then select the "Save and retrieve data" link.

>Select "Add the Realtime Database to your app."

>Check the changes that Android Studio is about to make to your project, and then click "Accept Changes."

>Add play-services-location as a project dependency.

///Your project's "dependencies" section should now look something like this:

XML

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-
layout:1.0.2'
    implementation 'com.google.firebase:firebase-auth:11.8.0'
    implementation 'com.google.android.gms:play-services-
location:11.8.0'
    implementation 'com.google.firebase:firebase-database:11.8.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation
'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

step 3 getting access to the location

Since our app is going to access the device's location and then send this information to a remote database, it needs the location and internet permissions, so add the following to the Manifest

XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jessicathornsby.mytrackerapp">

    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>
```

step 4 setting up a request to the user for a permission to share this sensitive information to the internet.

app should provide a visual indication the entire time it's recording and sharing this potentially sensitive information, and the user must be able to suspend tracking at any time.

create a persistent notification that'll be onscreen the entire time this application is recording the device's location. The user can suspend tracking by tapping this notification – which will be the only way of dismissing the notification.

create the icon and text that we'll use in this notification. via Android Studio's built-in Asset Studio:

```
>Control-click your project's "res/drawable" folder and then select
New > Image Asset.
>Control-click your project's "res/drawable" folder and then select
New > Image Asset.
>Open the "Icon Type" dropdown and then select "Notification Icons."
>Click the little button that appears alongside the "Clip Art"
label.
>Choose the icon you want to use; like "My Location." Click "OK."
>Name this icon "tracking_enabled," and then click "Next."
>Check the information on the subsequent screen, and then click
"Finish."
```

Next, open the strings.xml file and create the notification text. While the strings.xml file is open, add firebase_path label that'll appear alongside the data in Firebase, plus the email and password for the test user we registered in the Firebase .

XML

```
<string name="tracking_enabled_notif">Tracking is currently enabled.
Tap to cancel.</string>
<string name="test_email">Rector971120@gmail.com</string>
<string name="test_password">766569874</string>
<string name="firebase_path">location</string>
```

step 5 start tracking the device location.

in MainActivity:

- Request access to the device's location. MainActivity will need to check whether it currently has access to the user's location, every single time it's launched
- Start location tracking. The MainActivity should check whether location tracking is enabled, and then start the location tracking service if necessary.
- Display a toast. MainActivity should display a toast when location tracking is first enabled.
- Exit the app. Once MainActivity has started the location tracking service, its work is done. Rather than waiting for the user to close the application, MainActivity should take the initiative and shut itself down automatically.

code

JAVA

```
import android.app.Activity;
import android.support.v4.app.ActivityCompat;
import android.os.Bundle;
import android.support.v4.content.ContextCompat;
import android.content.Intent;
import android.location.LocationManager;
import android.Manifest;
import android.content.pm.PackageManager;
import android.widget.Toast;

public class MainActivity extends Activity {

    private static final int PERMISSIONS_REQUEST = 100;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //Check whether GPS tracking is enabled//

        LocationManager lm = (LocationManager)
getSystemService(LOCATION_SERVICE);
        if (!lm.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
            finish();
        }

        //Check whether this app has access to the location permission//

        int permission = ContextCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION);

        //If the location permission has been granted, then start the
TrackerService//

        if (permission == PackageManager.PERMISSION_GRANTED) {
```

```

        startTrackerService();
    } else {

//If the app doesn't currently have access to the user's location,
then request access//

        ActivityCompat.requestPermissions(this,
            new String[]
{Manifest.permission.ACCESS_FINE_LOCATION},
            PERMISSIONS_REQUEST);
    }

    }

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

//If the permission has been granted...//

        if (requestCode == PERMISSIONS_REQUEST && grantResults.length
== 1
            && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

//...then start the GPS tracking service//

            startTrackerService();
        } else {

//If the user denies the permission request, then display a toast
with some more information//

            Toast.makeText(this, "Please enable location services to
allow GPS tracking", Toast.LENGTH_SHORT).show();
        }
    }

//Start the TrackerService//

    private void startTrackerService() {
        startService(new Intent(this, TrackingService.class));
    }

//Notify the user that tracking has been enabled//

    Toast.makeText(this, "GPS tracking enabled",
Toast.LENGTH_SHORT).show();

//Close MainActivity//

    finish();
}

}

```

step 6. create a location tracking service

We now need to create the service that's responsible for sending the device's location to Firebase.

Create a new service, by selecting File > New > Service > Service from the Android Studio toolbar. Name the service "TrackingService," and then add the following:

JAVA

```
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
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.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.support.v4.content.ContextCompat;
import android.os.IBinder;
import android.content.Intent;
import android.content.IntentFilter;
import android.util.Log;
import android.Manifest;
import android.location.Location;
import android.app.Notification;
import android.content.pm.PackageManager;
import android.app.PendingIntent;
import android.app.Service;
```

```
public class TrackingService extends Service {
```

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

```
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
```

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

```

        loginToFirebase();
    }

    //Create the persistent notification//

    private void buildNotification() {
        String stop = "stop";
        registerReceiver(stopReceiver, new IntentFilter(stop));
        PendingIntent broadcastIntent = PendingIntent.getBroadcast(
            this, 0, new Intent(stop),
            PendingIntent.FLAG_UPDATE_CURRENT);

    // Create the persistent notification//
        Notification.Builder builder = new Notification.Builder(this)
            .setContentTitle(getString(R.string.app_name))
            .setContentText(getString(R.string.tracking_enabled_n
otif))

    //Make this notification ongoing so it can't be dismissed by the
    user//

        .setOngoing(true)
        .setContentIntent(broadcastIntent)
        .setSmallIcon(R.drawable.tracking_enabled);
        startForeground(1, builder.build());
    }

    protected BroadcastReceiver stopReceiver = new
BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {

    //Unregister the BroadcastReceiver when the notification is tapped//

            unregisterReceiver(stopReceiver);

    //Stop the Service//

            stopSelf();
        }
    };

    private void loginToFirebase() {

    //Authenticate with Firebase, using the email and password we
    created earlier//

        String email = getString(R.string.test_email);
        String password = getString(R.string.test_password);

    //Call OnCompleteListener if the user is signed in successfully//

        FirebaseAuth.getInstance().signInWithEmailAndPassword(
            email, password).addOnCompleteListener(new

```

```

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

//If the user has been authenticated...//

        if (task.isSuccessful()) {

//...then call requestLocationUpdates//

            requestLocationUpdates();
        } else {

//If sign in fails, then log the error//

            Log.d(TAG, "Firebase authentication failed");
        }
    }
});
}

//Initiate the request to track the device's location//

private void requestLocationUpdates() {
    LocationRequest request = new LocationRequest();

//Specify how often your app should request the device's location//

    request.setInterval(10000);

//Get the most accurate location data available//

    request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    FusedLocationProviderClient client =
LocationServices.getFusedLocationProviderClient(this);
    final String path = getString(R.string.firebase_path);
    int permission = ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION);

//If the app currently has access to the location permission...//

    if (permission == PackageManager.PERMISSION_GRANTED) {

//...then request location updates//

        client.requestLocationUpdates(request, new
LocationCallback() {
            @Override
            public void onLocationResult(LocationResult
locationResult) {

//Get a reference to the database, so your app can perform read and
write operations//

```

```

        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference(path);
        Location location =
locationResult.getLastLocation();
        if (location != null) {

//Save the location data to the database//

            ref.setValue(location);
        }
    }, null);
}
}

import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
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.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.support.v4.content.ContextCompat;
import android.os.IBinder;
import android.content.Intent;
import android.content.IntentFilter;
import android.util.Log;
import android.Manifest;
import android.location.Location;
import android.app.Notification;
import android.content.pm.PackageManager;
import android.app.PendingIntent;
import android.app.Service;

public class TrackingService extends Service {

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

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override

```



```

    public void onCreate() {
        super.onCreate();
        buildNotification();
        loginToFirebase();
    }

    //Create the persistent notification//

    private void buildNotification() {
        String stop = "stop";
        registerReceiver(stopReceiver, new IntentFilter(stop));
        PendingIntent broadcastIntent = PendingIntent.getBroadcast(
            this, 0, new Intent(stop),
            PendingIntent.FLAG_UPDATE_CURRENT);

        // Create the persistent notification//
        Notification.Builder builder = new Notification.Builder(this)
            .setContentTitle(getString(R.string.app_name))
            .setContentText(getString(R.string.tracking_enabled_n
otif))

        //Make this notification ongoing so it can't be dismissed by the
        user//

        builder.setOngoing(true)
            .setContentIntent(broadcastIntent)
            .setSmallIcon(R.drawable.tracking_enabled);
        startForeground(1, builder.build());
    }

    protected BroadcastReceiver stopReceiver = new
BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {

            //Unregister the BroadcastReceiver when the notification is tapped//

            unregisterReceiver(stopReceiver);

            //Stop the Service//

            stopSelf();
        }
    };

    private void loginToFirebase() {

        //Authenticate with Firebase, using the email and password we
        created earlier//

        String email = getString(R.string.test_email);
        String password = getString(R.string.test_password);

        //Call OnCompleteListener if the user is signed in successfully//

```

```

        FirebaseAuth.getInstance().signInWithEmailAndPassword(
            email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task) {

//If the user has been authenticated...//

                if (task.isSuccessful()) {

//...then call requestLocationUpdates//

                    requestLocationUpdates();
                } else {

//If sign in fails, then log the error//

                    Log.d(TAG, "Firebase authentication failed");
                }
            }
        });
    }

//Initiate the request to track the device's location//

    private void requestLocationUpdates() {
        LocationRequest request = new LocationRequest();

//Specify how often your app should request the device's location//

        request.setInterval(10000);

//Get the most accurate location data available//

        request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
        FusedLocationProviderClient client =
LocationServices.getFusedLocationProviderClient(this);
        final String path = getString(R.string.firebase_path);
        int permission = ContextCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION);

//If the app currently has access to the location permission...//

        if (permission == PackageManager.PERMISSION_GRANTED) {

//...then request location updates//

            client.requestLocationUpdates(request, new
LocationCallback() {
                @Override
                public void onLocationResult(LocationResult
locationResult) {

```

```
//Get a reference to the database, so your app can perform read and write operations//
```

```
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference(path);
        Location location =
locationResult.getLastLocation();
        if (location != null) {

//Save the location data to the database//

                ref.setValue(location);
            }
        }
    }, null);
}
}
```

step 7 testing the application

- >Select "More or ..." from the strip of buttons that appears alongside the emulator window
- >Select "Location" from the left-hand menu.
- >Enter your new coordinates into the "Longitude" and "Latitude" fields, and then click "Send." The emulator will now use these new coordinates.

To check that your app is recording the device's location to Firebase:

- >Install the project/application on your device.
- >When prompted, grant the app access to your location. You should see a toast, informing you that GPS tracking is now enabled.
- >Open the notification drawer, and check that the persistent notification has been created.
- >head over to the Firebase Console and open the project that's linked to the tracking app.(i used my google account to log in to firebase so i will check for you)
- >In the left-hand menu, select "Database."
- >Select the "Data" tab, we should see that location data is now appearing in the Firebase Console.

Good Luck