```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Travel Buddy"
        android:supportsRtl="true"
        android:theme="@style/Theme.TravelBuddy"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyDvs8kw7EQfAjcmWfMWGRNLd5IgsUEkQP8" />

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:exported="true"
            android:theme="@style/Theme.AppCompat.NoActionBar"
            android:windowSoftInputMode="stateHidden|adjustPan"
            android:configChanges="orientation|screenSize">
            <intent-filter android:priority="100">
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>

        <activity
            android:name=".ExploreActivity"
            android:label="@string/app_name"
            android:exported="true"
            android:theme="@style/Theme.AppCompat.NoActionBar"
            android:windowSoftInputMode="stateHidden|adjustPan"
            android:configChanges="orientation|screenSize">
        </activity>

        <activity
            android:name=".ThingstodoActivity"
            android:label="@string/app_name"
            android:exported="true"
            android:windowSoftInputMode="stateHidden|adjustPan"
            android:configChanges="orientation|screenSize">
        </activity>

        <activity
            android:name=".ThingsToDoCategoryAdapter"
            android:label="@string/app_name"
            android:exported="true"
            android:windowSoftInputMode="stateHidden|adjustPan"
            android:configChanges="orientation|screenSize">
        </activity>

        <activity
            android:name=".MapRecommendationActivity"
            android:label="@string/app_name"
            android:exported="true"
            android:windowSoftInputMode="stateHidden|adjustPan"
            android:configChanges="orientation|screenSize">
        </activity>


    </application>

</manifest>


<!--    android:theme="@style/Theme.TravelBuddy"-->
package com.example.travelbuddy;

public class Places {
    private String name;
    private String address;
    private double latitude;

    private double longitude;

```

```java
    public Places(String name, String address, double latitude, double longitude) {
        this.name = name;
        this.address = address;
        this.latitude = latitude;
        this.longitude = longitude;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public double getLongitude() {
        return longitude;
    }

    public double getLatitude() {
        return latitude;
    }
}

package com.example.travelbuddy;

import android.content.Context;

import io.appwrite.Client;
import io.appwrite.services.Account;
import io.appwrite.services.Databases;


public class AppwriteClientManager {
    private static Client client;
    private static Account account;
    private static Databases database;

    public static void initialize(Context context) {
        client = new Client(context);
        client.setEndpoint("http://10.0.0.172/v1")
                .setProject("64136f06c970db619408")
                .setSelfSigned(true); // For self-signed certificates, only use for development
        account = new Account(client);
        database = new Databases(client);
    }

    public static Client getClient() {
        return client;
    }
    public static Databases getDatabase() {return database;}
    public static Account getAccount() {
        return account;
    }


}
package com.example.travelbuddy

import com.example.travelbuddy.AppwriteClientManager
import com.google.gson.Gson
import com.google.gson.JsonObject
import io.appwrite.Client
import io.appwrite.exceptions.AppwriteException
import io.appwrite.services.Account
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import okhttp3.Response
import java.io.IOException

class AppwriteUserHelper {
    private val client: Client =
AppwriteClientManager.getClient()
    private val account: Account =
AppwriteClientManager.getAccount()
    private var userId: String? = null

    init {
        CoroutineScope(Dispatchers.IO).launch {
            userId = fetchUserId()
        }
    }

    private suspend fun fetchUserId(): String? {
        return withContext(Dispatchers.IO) {
            try {
```

```kotlin
                val account = AppwriteClientManager.getAccount()
                val user = account.get()
                user.id
            } catch (e: AppwriteException) {
                e.printStackTrace()
                null
            }
        }
    }

    fun getUserId(): String? {
        while (userId == null) {
            try {
                Thread.sleep(100)
            } catch (e: InterruptedException) {
                e.printStackTrace()
            }
        }
        return userId
    }
}
```

```java
package com.example.travelbuddy;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

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

public class CustomInfoWindowAdapter implements GoogleMap.InfoWindowAdapter {

    private final View mView;
    private final LayoutInflater layoutInflater;
    private final Context context;

    public CustomInfoWindowAdapter(Context context) {
        mView = LayoutInflater.from(context).inflate(R.layout.custom_info_window, null);
        this.context = context;
        this.layoutInflater = LayoutInflater.from(context);
    }

    @Nullable
    @Override
    public View getInfoWindow(@NonNull Marker marker) {
        return null; // Use the default InfoWindow frame
    }

    @Nullable
    @Override
    public View getInfoContents(Marker marker) {
        View view = layoutInflater.inflate(R.layout.custom_info_window, null);

        TextView titleView = view.findViewById(R.id.title);
        TextView snippetView = view.findViewById(R.id.snippet);

        titleView.setText(marker.getTitle());
        snippetView.setText(marker.getSnippet());

        return view;
    }
}


package com.example.travelbuddy;
import androidx.fragment.app.FragmentActivity;

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

```

```java
public class ExploreActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_explore);
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // Add a marker in a location and move the camera
        LatLng location = new LatLng(-34, 151);
        mMap.addMarker(new MarkerOptions().position(location).title("Marker in a location"));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location, 15));
    }
}
package com.example.travelbuddy;

import android.os.Bundle;

import androidx.activity.ComponentActivity;
import androidx.appcompat.app.AppCompatActivity;

import android.util.Log;
import android.view.Window;
import android.view.WindowManager;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.view.View;

import java.util.List;
import java.util.concurrent.Executor;
import java.util.concurrent.Executors;

import io.appwrite.Client;
import io.appwrite.exceptions.AppwriteException;
import io.appwrite.models.Document;
import io.appwrite.services.Account;

import io.appwrite.coroutines.CoroutineCallback;
import android.widget.Button;
import android.widget.Toast;
import android.content.Intent;
import android.net.Uri;
import io.appwrite.Client;
import io.appwrite.coroutines.CoroutineCallback;
import io.appwrite.services.Account;


public class MainActivity extends AppCompatActivity{
    public static MainActivity instance;

    ImageView imageView;
    TextView textView;
    private EditText etEmail;
    private EditText etPassword;
    private Button btnSignUp;
    private Button btnSignIn;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        instance = this;
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = findViewById(R.id.imageView);
        textView = findViewById(R.id.textView);
        etEmail = findViewById(R.id.email);
        etPassword = findViewById(R.id.password);
        btnSignUp = findViewById(R.id.signUpButton);
        btnSignIn = findViewById(R.id.signInButton);


        // Initialize the Appwrite client manager
        AppwriteClientManager.initialize(this);

        btnSignUp.setOnClickListener(v -> signUp());
        btnSignIn.setOnClickListener(v -> signIn());



    }

    private void signUp() {
        String email = etEmail.getText().toString().trim();
        String password = etPassword.getText().toString().trim();

        if (email.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
            return;
        }

        String[] emailParts = email.split("@");
        String userId = emailParts[0];

        // Access the Appwrite client and account instances
        Client client = AppwriteClientManager.getClient();
        Account account = AppwriteClientManager.getAccount();

        // Use the list() method to check if the account already exists
        Executor executor = Executors.newSingleThreadExecutor();
        executor.execute(() -> {
            try {
                // Create a new account
                account.create(
                        userId,
                        email,
                        password,
                        new CoroutineCallback<>((result, error) -> {
                            if (error != null) {
                                error.printStackTrace();
                                return;
                            }

```

```java
                                Log.d("Appwrite", result.toString());
                                runOnUiThread(() -> {
                                    Toast.makeText(MainActivity.this, "Successfully signed up!", Toast.LENGTH_SHORT).show();
                                });
                            })

                    );
                } catch (AppwriteException e) {
                    Log.e("Appwrite", "AppwriteException: " + e.getMessage());
                }
            });
    }


    private void signIn() {
        String email = etEmail.getText().toString().trim();
        String password = etPassword.getText().toString().trim();

        if (email.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
            return;
        }

        // Access the Appwrite client and account instances
        Client client = AppwriteClientManager.getClient();
        Account account = AppwriteClientManager.getAccount();

        Executor executor = Executors.newSingleThreadExecutor();
        executor.execute(() -> {
            account.createEmailSession(
                    email,
                    password,
                    new CoroutineCallback<>((result, error) -> {
                        if (error != null) {
                            error.printStackTrace();
                            runOnUiThread(() -> Toast.makeText(MainActivity.this, "Failed to sign in.", Toast.LENGTH_SHORT).show());
                            return;
                        }

                        Log.d("Appwrite", result.toString());
                        runOnUiThread(() -> Toast.makeText(MainActivity.this, "Successfully signed in!", Toast.LENGTH_SHORT).show());
                        // Start ExploreActivity
                        Intent intent = new Intent(MainActivity.this, ThingstodoActivity.class);
                        startActivity(intent);
                    })
            );
        });
    }

}
package com.example.travelbuddy;
import com.example.travelbuddy.AppwriteUserHelper;


import android.Manifest;
import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import android.content.pm.ApplicationInfo;

import java.util.UUID;

import io.appwrite.Client;
import io.appwrite.coroutines.CoroutineCallback;
import io.appwrite.exceptions.AppwriteException;

import io.appwrite.services.Account;
import io.appwrite.services.Databases;




import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import androidx.fragment.app.FragmentManager;

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.Task;

import com.google.android.libraries.places.api.model.RectangularBounds;
import com.google.android.libraries.places.api.model.TypeFilter;
import com.google.android.libraries.places.api.net.PlacesClient;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.bottomsheet.BottomSheetDialog;
import com.google.android.material.navigation.NavigationBarView;
import com.google.gson.Gson;
import com.google.gson.JsonObject;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.atomic.AtomicReference;

import okhttp3.Call;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;


public class MapRecommendationActivity extends FragmentActivity implements OnMapReadyCallback {
    private static final String PLACES_API_BASE_URL = "https://maps.googleapis.com/maps/api/place/nearbysearch/json";

    private Fragment homeFragment;
    private Fragment searchFragment;
    private String userId;
    private AppwriteUserHelper appwriteUserHelper;

    private Fragment activeFragment;
    private FragmentManager fragmentManager;

    private GoogleMap mMap;
    private String category;
    PlacesClient placesClient;

    RectangularBounds bounds;
    TypeFilter typeFilter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map_recommendation);
        AppwriteClientManager.initialize(this);

        appwriteUserHelper = new AppwriteUserHelper();
        userId = appwriteUserHelper.getUserId();

        // Get the intent and the category
        Intent intent = getIntent();
        category = intent.getStringExtra("category");

        // 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);


        BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigationView);
        // Initialize your fragments
        homeFragment = getSupportFragmentManager().findFragmentById(R.id.map);
//        searchFragment = new FragmentA(); // Replace with your desired fragment
        AccountFragment accountFragment = new AccountFragment();

        fragmentManager = getSupportFragmentManager();

        // Set the initial fragment
//        fragmentManager.beginTransaction().add(R.id.fragmentContainer, searchFragment).hide(searchFragment).commit();
        fragmentManager.beginTransaction().add(R.id.fragmentContainer, accountFragment).hide(accountFragment).commit();
        activeFragment = homeFragment;

        // Set up the BottomNavigationView listener
        bottomNavigationView.setOnItemSelectedListener(new NavigationBarView.OnItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.navigation_home:
                        replaceFragment(homeFragment);
                        return true;
                    // case R.id.navigation_search:
                    //     replaceFragment(searchFragment);
                    //     return true;
                    case R.id.navigation_account:
                        replaceFragment(accountFragment);
                        return true;
                }
                return false;
            }
        });

    }

    private void replaceFragment(Fragment fragment) {
        if (activeFragment != fragment) {
            fragmentManager.beginTransaction().hide(activeFragment).show(fragment).commit();
            activeFragment = fragment;
        }
    }

    @SuppressLint("PotentialBehaviorOverride")
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        mMap.setInfoWindowAdapter(new CustomInfoWindowAdapter(this));

        mMap.setOnInfoWindowClickListener(marker -> {
            // Show a custom dialog or bottom sheet with interactive elements
            showBottomSheet(marker);
        });


        requestLocationPermissions();
        // Use the Places API to search for nearby places based on the selected category
        // and add markers to the map
    }

    private void showBottomSheet(Marker marker) {
        // Inflate the bottom sheet view
        View bottomSheetView = LayoutInflater.from(this).inflate(R.layout.bottom_sheet, null);
        TextView titleView = bottomSheetView.findViewById(R.id.bottom_sheet_title);
        TextView snippetView = bottomSheetView.findViewById(R.id.bottom_sheet_snippet);
        Button addButton = bottomSheetView.findViewById(R.id.add_to_list_button);

        titleView.setText(marker.getTitle());
        snippetView.setText(marker.getSnippet());

        // Set the click listener for the "Add to list" button
        addButton.setOnClickListener(v -> {
            // Show the AlertDialog when the button is clicked
            new AlertDialog.Builder(MapRecommendationActivity.this)
                    .setTitle("Add to list")
                    .setMessage("Are you sure you want to add '" + marker.getTitle() + "' to your list?")
                    .setPositiveButton("Yes, add it!", (dialog, which) -> {
                        // Call the addPlaceToAppwrite() function to save the place
                        try {
                            addPlaceToAppwrite(marker.getTitle(), marker.getSnippet(), marker.getPosition().latitude, marker.getPosition().longitude);
                        } catch (AppwriteException e) {
                            throw new RuntimeException(e);
                        }
                        Toast.makeText(MapRecommendationActivity.this, "Added: " + marker.getTitle(), Toast.LENGTH_SHORT).show();
                    })
                    .setNegativeButton("No, cancel", null) // No action needed for the negative button
                    .show();
        });

        // Show the bottom sheet
        BottomSheetDialog bottomSheetDialog = new BottomSheetDialog(this);
        bottomSheetDialog.setContentView(bottomSheetView);
        bottomSheetDialog.show();
    }


    private void addPlaceToAppwrite(String name, String address, double latitude, double longitude) throws AppwriteException {
        Client client = AppwriteClientManager.getClient();
        Account account = AppwriteClientManager.getAccount();
        Databases databases =
AppwriteClientManager.getDatabase();


        // Do something with the userId here
            Map<String, Object> placeData = new HashMap<>();
            placeData.put("name", name);
            placeData.put("address", address);
            placeData.put("latitude", latitude);
            placeData.put("longitude", longitude);
        placeData.put("userId", userId);


            String uniqueDocumentId =
UUID.randomUUID().toString();

            databases.createDocument(
                    "642481c8a9ed2d76e6ef",
                    "642481d1ba3a9bd5359b",
                    uniqueDocumentId,
                    placeData,
                    new CoroutineCallback<>((result, error) ->
{
                        if (error != null) {
                            Log.d("Appwrite", "error" + name +
address + latitude + longitude);
                            error.printStackTrace();
                            return;
                        }

                        Log.d("Appwrite", result.toString());
                        Log.d("Appwrite", "Place added
successfully:");
                    })
            );

        }



    private static final int LOCATION_PERMISSION_REQUEST_CODE =
1;

```

```java
    private void requestLocationPermissions() {
        ActivityCompat.requestPermissions(this,
                new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION},
                LOCATION_PERMISSION_REQUEST_CODE);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getCurrentLocation();
            } else {
                Toast.makeText(this, "Location permission
denied", Toast.LENGTH_SHORT).show();
            }
        }
    }

    private FusedLocationProviderClient fusedLocationClient;

    private void getCurrentLocation() {
        fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            requestLocationPermissions();
            return;
        }

        Task<Location> locationTask =
fusedLocationClient.getLastLocation();
        locationTask.addOnSuccessListener(location -> {
            if (location != null) {
                // Use the user's current location
                double latitude = location.getLatitude();
                double longitude = location.getLongitude();
                LatLng currentLatLng = new LatLng(latitude, longitude);

                // Add a marker on the user's current location
              mMap.addMarker(new MarkerOptions().position(currentLatLng).title("You are here"));

                // Move the camera to the user's current location
                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLatLng, 15));
               // fetchNearbyPlaces(curtLatLng);
                try {
                    getNearbyPlacesBasedOnCategory(latitude, longitude);
                } catch (PackageManager.NameNotFoundException e) {
                    throw new RuntimeException(e);
                }

            } else {
                requestLocationPermissions();
                Toast.makeText(this, "Unable to get current location", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void getNearbyPlacesBasedOnCategory(double latitude, double longitude) throws PackageManager.NameNotFoundException {
        LatLng currentLatLng = new LatLng(latitude, longitude);
        double radiusInMeters = 5000; // 5 km

        // Use the Nearby Search API to search for nearby restaurants
        ApplicationInfo appInfo = getPackageManager().getApplicationInfo(getPackageName(), PackageManager.GET_META_DATA);
        String apiKey = appInfo.metaData.getString("com.google.android.geo.API_KEY");
        String url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=" +
                currentLatLng.latitude + "," + currentLatLng.longitude +
                "&radius=" + radiusInMeters +
                "&type=" + category +
                "&key=" + apiKey;

        OkHttpClient client = new OkHttpClient();
        Request request = new Request.Builder().url(url).build();

        client.newCall(request).enqueue(new okhttp3.Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                e.printStackTrace();
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                if (!response.isSuccessful()) {
                    throw new IOException("Unexpected code " + response);
                }

                String responseBody = response.body().string();
                try {
                    JSONObject json = new JSONObject(responseBody);
                    JSONArray results = json.getJSONArray("results");
                    for (int i = 0; i < results.length(); i++) {
                        JSONObject result = results.getJSONObject(i);
                        JSONObject location = result.getJSONObject("geometry").getJSONObject("location");
                        String name = result.getString("name");
                        String address = result.getString("vicinity");
                        double lat = location.getDouble("lat");
                        double lng = location.getDouble("lng");
                        LatLng placeLatLng = new LatLng(lat, lng);

                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                mMap.addMarker(new MarkerOptions()
                                        .position(placeLatLng)
                                        .title(name)
                                        .snippet(address));
                            }
                        });
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            }
        });
    }
}
package com.example.travelbuddy;

public class Places {
    private String name;
    private String address;
    private double latitude;

    private double longitude;

    public Places(String name, String address, double latitude, double longitude) {
        this.name = name;
        this.address = address;
        this.latitude = latitude;
        this.longitude = longitude;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public double getLongitude() {
        return longitude;
    }

    public double getLatitude() {
        return latitude;
    }
}
package com.example.travelbuddy;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;
import android.view.ViewGroup;
import android.widget.TextView;

import
com.example.travelbuddy.databinding.FragmentItemBinding;

import java.util.List;

/**
 * {@link RecyclerView.Adapter} that can display a {@link
PlaceholderItem}.
 * TODO: Replace the implementation with code for your data
type.
 */


public class PlacesItemRecyclerViewAdapter extends
RecyclerView.Adapter<PlacesItemRecyclerViewAdapter.ViewHolder> {

    private final List<Places> mValues;

    public PlacesItemRecyclerViewAdapter(List<Places> items) {
        mValues = items;
    }

    @NonNull
    @Override
```

```java
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        FragmentItemBinding binding = FragmentItemBinding.inflate(LayoutInflater.from(parent.getContext()), parent, false);
        return new ViewHolder(binding);
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        Places places = mValues.get(position);
        holder.mItem = places;
        holder.mIdView.setText(places.getName());
        holder.mContentView.setText(places.getAddress());
    }

    @Override
    public int getItemCount() {
        return mValues.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        public final TextView mIdView;
        public final TextView mContentView;
        public Places mItem;

        public ViewHolder(FragmentItemBinding binding) {
            super(binding.getRoot());
            mIdView = binding.name;
            mContentView = binding.address;
        }

        @Override
        public String toString() {
            return super.toString() + " '" + mContentView.getText() + "'";
        }
    }
}
package com.example.travelbuddy;

import android.content.Intent;
import android.content.pm.ApplicationInfo;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Bundle;
import android.util.Log;
import android.widget.SearchView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.Status;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.libraries.places.api.Places;
import
  com.google.android.libraries.places.api.model.AutocompletePredict
  ion;
import com.google.android.libraries.places.api.model.Place;
import
  com.google.android.libraries.places.api.model.RectangularBounds;
import
  com.google.android.libraries.places.api.model.TypeFilter;
import
  com.google.android.libraries.places.api.net.FindAutocompletePredi
  ctionsRequest;
import
  com.google.android.libraries.places.api.net.PlacesClient;

import java.util.Arrays;
import java.util.List;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.gms.maps.SupportMapFragment;
import
  com.google.android.libraries.places.widget.AutocompleteSupportFra
  gment;
import
  com.google.android.libraries.places.widget.listener.PlaceSelectio
  nListener;

public class ThingstodoActivity extends AppCompatActivity {


    PlacesClient placesClient;
    private RecyclerView recyclerView;
    private ThingsToDoCategoryAdapter thingsToDoCategoryAdapter;
    private List<String> categories;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_thingstodo);
        try {
            ApplicationInfo appInfo = getPackageManager().getApplicationInfo(getPackageName(), PackageManager.GET_META_DATA);
            String apiKey = appInfo.metaData.getString("com.google.android.geo.API_KEY");
            Places.initialize(getApplicationContext(), apiKey);

            placesClient = Places.createClient(this);

            final AutocompleteSupportFragment autocompleteSupportFragment = (AutocompleteSupportFragment) getSupportFragmentManager().findFragmentById(R.id.autocomplete_fragment);

            autocompleteSupportFragment.setPlaceFields(Arrays.asList(Place.Field.ID, Place.Field.LAT_LNG, Place.Field.NAME));
            autocompleteSupportFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {
                @Override
                public void onError(@NonNull Status status) {

                }

                @Override
                public void onPlaceSelected(@NonNull Place place) {
                    final LatLng latLng = place.getLatLng();
                    Log.i("on place selected", "wow" + latLng.latitude);
                }
            });

        }
        catch (PackageManager.NameNotFoundException e) {
            Log.e("TAG", "Failed to load meta-data,
NameNotFound: " + e.getMessage());
        } catch (NullPointerException e) {
            Log.e("TAG", "Failed to load meta-data,
NullPointer: " + e.getMessage());
        }
        List<String> categories = Arrays.asList(
                "accounting", "airport", "amusement_park",
"aquarium", "art_gallery", "atm", "bakery", "bank", "bar",
"beauty_salon", "bicycle_store", "book_store", "bowling_alley",
"bus_station", "cafe", "campground", "car_dealer", "car_rental",
"car_repair", "car_wash", "casino", "cemetery", "church",
"city_hall", "clothing_store", "convenience_store", "courthouse",
"dentist", "department_store", "doctor", "drugstore",
"electrician", "electronics_store", "embassy", "fire_station",
"florist", "funeral_home", "furniture_store", "gas_station",
"gym", "hair_care", "hardware_store", "hindu_temple",
"home_goods_store", "hospital", "insurance_agency",
"jewelry_store", "laundry", "lawyer", "library",
"light_rail_station", "liquor_store", "local_government_office",
"locksmith", "lodging", "meal_delivery", "meal_takeaway",
"mosque", "movie_rental", "movie_theater", "moving_company",
"museum", "night_club", "painter", "park", "parking",
"pet_store", "pharmacy", "physiotherapist", "plumber", "police",
"post_office", "primary_school", "real_estate_agency",
"restaurant", "roofing_contractor", "rv_park", "school",
"secondary_school", "shoe_store", "shopping_mall", "spa",
"stadium", "storage", "store", "subway_station", "supermarket",
"synagogue", "taxi_stand", "tourist_attraction", "train_station",
"transit_station", "travel_agency", "university",
"veterinary_care", "zoo"
        );

        RecyclerView recyclerView =
findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        recyclerView.setHasFixedSize(true); // Set this if your
RecyclerView items have a fixed size

```

```java
        ThingsToDoCategoryAdapter adapter = new ThingsToDoCategoryAdapter(categories, new ThingsToDoCategoryAdapter.OnCategoryClickListener() {
            @Override
            public void onCategoryClick(int position) {
                String selectedCategory = categories.get(position);
                Intent intent = new Intent(ThingstodoActivity.this, MapRecommendationActivity.class);
                intent.putExtra("category", selectedCategory);
                ThingstodoActivity.this.startActivity(intent);
            }

        });
        recyclerView.setAdapter(adapter);}


}

package com.example.travelbuddy;

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

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class ThingsToDoCategoryAdapter extends RecyclerView.Adapter<ThingsToDoCategoryAdapter.CategoryViewHolder> {

    private List<String> categories;
    private OnCategoryClickListener onCategoryClickListener;


    public ThingsToDoCategoryAdapter(List<String> categories, OnCategoryClickListener onCategoryClickListener) {
        this.categories = categories;
        this.onCategoryClickListener = onCategoryClickListener;
    }

    @NonNull
    @Override
    public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.list_item_category, parent, false);
        return new CategoryViewHolder(itemView, onCategoryClickListener);
    }

    @Override
    public void onBindViewHolder(@NonNull CategoryViewHolder holder, int position) {
        String categoryName = categories.get(position);
        holder.categoryName.setText(categoryName);
    }

    @Override
    public int getItemCount() {
        return categories.size();
    }

    public static class CategoryViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {

        TextView categoryName;
        OnCategoryClickListener onCategoryClickListener;

        public CategoryViewHolder(@NonNull View itemView, OnCategoryClickListener onCategoryClickListener) {
            super(itemView);
            categoryName = itemView.findViewById(R.id.categoryName);
            this.onCategoryClickListener = onCategoryClickListener;
            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View v) {
            onCategoryClickListener.onCategoryClick(getAdapterPosition());
        }
    }

    public interface OnCategoryClickListener {
        void onCategoryClick(int position);
    }
}
/**
* Automatically generated file. DO NOT MODIFY
*/
package com.example.travelbuddy;

public final class BuildConfig {
  public static final boolean DEBUG = Boolean.parseBoolean("true");
  public static final String APPLICATION_ID = "com.example.travelbuddy";
  public static final String BUILD_TYPE = "debug";
  public static final int VERSION_CODE = 1;
  public static final String VERSION_NAME = "1.0";
}
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.travelbuddy'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.travelbuddy"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures {
        viewBinding true
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.0'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation
'androidx.test.espresso:espresso-core:3.5.1'
    implementation("io.appwrite:sdk-for-android:1.2.0")
    implementation
'androidx.navigation:navigation-fragment:2.4.1'
    implementation 'androidx.navigation:navigation-ui:2.4.1'
    implementation 'com.github.bumptech.glide:glide:3.7.0'
    implementation
'com.google.android.gms:play-services-maps:17.0.1'
    implementation
'com.google.maps.android:android-maps-utils:2.2.6'
    implementation
'com.google.android.libraries.places:places:2.4.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.recyclerview:recyclerview:1.2.1'
    implementation
'com.google.android.gms:play-services-location:18.0.0'
    implementation 'com.android.volley:volley:1.2.1'
    implementation
'androidx.databinding:databinding-runtime:4.0.0'
    implementation
'androidx.databinding:databinding-adapters:4.0.0'
    implementation 'org.json:json:20210307'
    implementation
"org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'com.squareup.okhttp3:okhttp:4.9.2'
    implementation 'com.google.code.gson:gson:2.8.8'

```

```


}
apply plugin: 'com.android.application'
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
```