**WEEK-7**

**AIM:** Develop a native application that uses GPS location information to track the mobile.

**DESCRIPTION:**

This native Android application utilizes GPS location services to track the mobile device's real-time location. It requests **ACCESS\_FINE\_LOCATION** and **ACCESS\_COARSE\_LOCATION** permissions in the **AndroidManifest.xml** to ensure accurate tracking. The app features a **MainActivity** that integrates Android's **LocationManager** to obtain precise latitude and longitude coordinates.

**CODES:**

**1.AndroidManifest.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"  
 package="com.example.gps">  
 <!-- Location Permissions -->  
 <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="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Gps"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>

**2.activity\_main.xml:-**

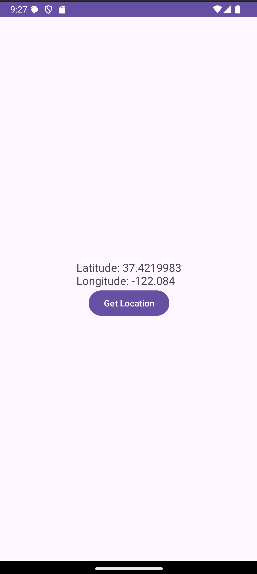
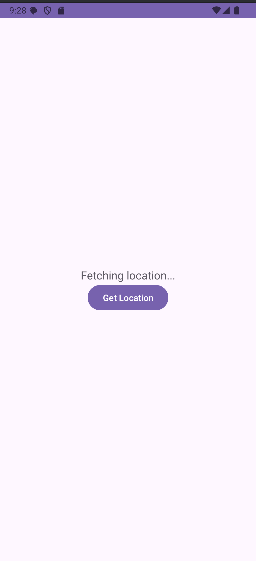
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="16dp">  
 <TextView  
 android:id="@+id/locationText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Fetching location..."  
 android:textSize="18sp" />  
 <Button  
 android:id="@+id/getLocationButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Get Location" />  
</LinearLayout>

**3. MainActivity.java:-**

package com.example.gps;  
import android.os.Bundle;  
import android.Manifest;  
import android.content.Intent;  
import android.content.IntentSender;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.os.Bundle;  
import android.provider.Settings;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.activity.result.ActivityResultLauncher;  
import androidx.activity.result.contract.ActivityResultContracts;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.ContextCompat;  
import com.google.android.gms.common.api.ResolvableApiException;  
import com.google.android.gms.location.\*;  
  
public class MainActivity extends AppCompatActivity {  
 private TextView locationText;  
 private Button getLocationButton;  
 private FusedLocationProviderClient fusedLocationClient;  
 private LocationRequest locationRequest;  
 private LocationCallback locationCallback;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 locationText = findViewById(R.id.*locationText*);  
 getLocationButton = findViewById(R.id.*getLocationButton*);  
 fusedLocationClient = LocationServices.*getFusedLocationProviderClient*(this);  
 getLocationButton.setOnClickListener(v -> checkGPSAndFetchLocation());  
 }  
 private void checkGPSAndFetchLocation() {  
 LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()  
 .addLocationRequest(locationRequest);  
  
 SettingsClient settingsClient = LocationServices.*getSettingsClient*(this);  
 settingsClient.checkLocationSettings(builder.build())  
 .addOnSuccessListener(locationSettingsResponse -> getLocation())  
 .addOnFailureListener(e -> {  
 if (e instanceof ResolvableApiException) {  
 try {  
 ((ResolvableApiException) e).startResolutionForResult(MainActivity.this, 101); fusedLocationClient.getLastLocation().addOnCompleteListener(task -> {  
 Location location = task.getResult();  
 if (location != null) {  
 double latitude = location.getLatitude();  
 double longitude = location.getLongitude();  
 locationText.setText("Latitude: " + latitude + "\nLongitude: " + longitude);  
 } else {  
 requestNewLocation();  
 }  
 });  
 } else {  
 requestPermissionLauncher.launch(Manifest.permission.*ACCESS\_FINE\_LOCATION*);  
 }  
 }  
 private void requestNewLocation() {  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) == PackageManager.*PERMISSION\_GRANTED*) {  
 fusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);  
 }  
 } {  
 checkGPSAndFetchLocation();  
 } else {  
 Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 );  
}

**4.** **gradel module app**plugins {  
 alias(*libs*.*plugins*.*android*.*application*)  
 // If using Kotlin, add:  
 // alias(libs.plugins.kotlin.android)  
}  
android {  
 namespace = "com.example.gps"  
 compileSdk = 35  
  
 defaultConfig {  
 applicationId = "com.example.gps"  
 minSdk = 24  
 targetSdk = 34  
 versionCode = 1  
 versionName = "1.0"  
  
 testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"  
 }  
 buildTypes {  
 release {  
 isMinifyEnabled = false  
 proguardFiles(  
 getDefaultProguardFile("proguard-android-optimize.txt"),  
 "proguard-rules.pro"  
 )  
 }  
 }  
 compileOptions {  
 sourceCompatibility = JavaVersion.VERSION\_11  
 targetCompatibility = JavaVersion.VERSION\_11  
 }  
}  
dependencies {  
 implementation(*libs*.*appcompat*)

appcompat = "1.7.0"  
material = "1.12.0"  
activity = "1.10.0"  
constraintlayout = "2.2.0"  
playServicesLocation = "21.0.1"  
playServicesBasement = "18.5.0" # Added Google Play Services Location  
  
" }  
  
[plugins]  
android-application = { id = "com.android.application", version.ref = "agp" }

**Output:**

**WEEK-8**

**AIM:** Develop a Mobile application for simple needs. (Attendance Management system, Bus tracking system and chat bot like etc.).

**DESCRIPTION:**The Truth and Dare Android app is an interactive party game that allows players to engage in classic truth and dare challenges through a digital medium. The game provides a structured yet engaging way for users to play, featuring a spinning bottle mechanism to determine turns, followed by a choice between answering a truth question or completing a dare challenge.

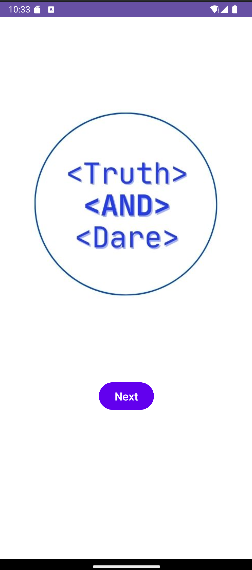
**Key Feature:**

**User-Friendly Interface:** The app features a simple and intuitive design, ensuring ease of navigation. The UI elements are arranged to provide a smooth experience, with properly spaced buttons, text fields, and images that contribute to the overall aesthetics of the app. The colour scheme and button designs maintain an engaging visual appeal.

**CODE:**

**1.activity\_main.xml:**  
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFFFFF">  
  
 <!-- Truth and Dare Logo -->  
 <ImageView  
 android:id="@+id/logoImage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:src="@drawable/logo" />  
  
 <!-- Next Button -->  
 <Button  
 android:id="@+id/nextButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/next"  
 android:layout\_below="@id/logoImage"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="30dp"  
 android:backgroundTint="#6200EE"  
 android:textColor="#FFFFFF"  
 android:padding="10dp"  
 android:textSize="18sp" />  
</RelativeLayout>

**2.MainActivity.java:**  
package com.example.truthanddareapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 Button nextButton = findViewById(R.id.*nextButton*);  
  
 // Navigate to Start Button Page  
 nextButton.setOnClickListener(v -> {  
 Intent intent = new Intent(MainActivity.this, StartActivity.class);  
 startActivity(intent);  
 });  
 }  
}

**OUTPUT**

**WEEK-9**

**AIM:** Develop a Mobile application for simple needs. (Attendance Management system, Bus tracking system and chat bot like etc.).

**DESCRIPTION:**The Truth and Dare Android app is an interactive party game that allows players to engage in classic truth and dare challenges through a digital medium. The game provides a structured yet engaging way for users to play, featuring a spinning bottle mechanism to determine turns, followed by a choice between answering a truth question or completing a dare challenge.

**Key Feature:**

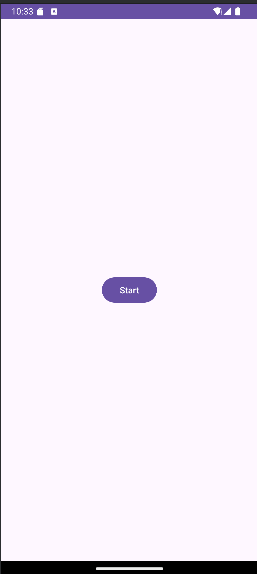
**Start Page:** Upon launching the app, users are greeted with a start page containing a “Start” button. This page is designed to set the tone for the game and directs the user to the main gameplay interface.

**CODE:**

**1.activity\_start.xml:**  
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/startGameButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/start\_button"  
 android:layout\_centerInParent="true" />  
</RelativeLayout>

**2.StartActivity.java:**  
package com.example.truthanddareapp;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
public class StartActivity extends AppCompatActivity{  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_start*);  
  
 // Change the ID to match the XML layout  
 Button startButton = findViewById(R.id.*startGameButton*);  
 startButton.setOnClickListener(v -> {  
 Intent intent = new Intent(StartActivity.this, SpinTheBottleActivity.class);  
 startActivity(intent);  
 });  
 }

**OUTPUT**



**WEEK-10**

**AIM:** Develop a Mobile application for simple needs. (Attendance Management system, Bus tracking system and chat bot like etc.).

**DESCRIPTION:**The Truth and Dare Android app is an interactive party game that allows players to engage in classic truth and dare challenges through a digital medium. The game provides a structured yet engaging way for users to play, featuring a spinning bottle mechanism to determine turns, followed by a choice between answering a truth question or completing a dare challenge.

**Key Features**

1. **Spin the Bottle Mechanism**: One of the most exciting elements of the app is the Spin the Bottle feature. After entering the game, users can tap the “Spin” button, which triggers an animation that rotates a bottle to select a random participant. This feature closely mimics the traditional physical game, adding an element of suspense.
2. **Truth and Dare Selection:** Once the bottle stops spinning, the selected player can choose between Truth and Dare options. Clicking the “Truth” button redirects the user to a screen displaying a randomly selected truth question, while clicking the “Dare” button leads to a screen with a randomly assigned dare challenge. These questions and dares are pre-loaded into the app and are selected randomly to ensure variety.

**CODE:**

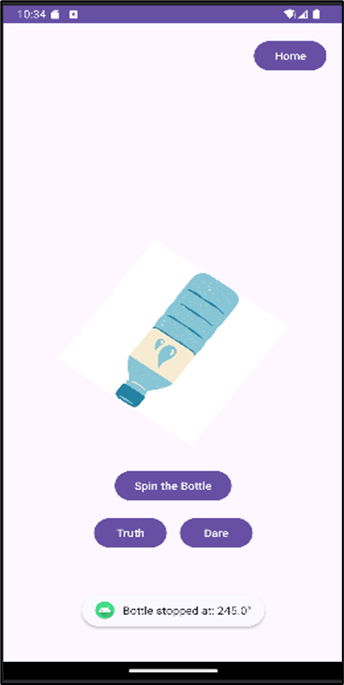
**1.activity\_dare.xml:**  
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="16dp">  
 <!-- Random Dare Question -->  
 <TextView  
 android:id="@+id/dareQuestion"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Random Dare Question Will Appear Here"  
 android:textSize="16sp"  
 android:gravity="center" />  
  
</LinearLayout>

**2.activity\_spin\_the\_bottle.xml:**  
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <!-- Spin Button -->  
 <Button  
 android:id="@+id/spinButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/spin\_bottle"  
 android:layout\_below="@id/bottleImage"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="20dp" />  
  
 <!-- LinearLayout for Truth and Dare Buttons -->  
 <LinearLayout  
 android:id="@+id/truthDareLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:padding="16dp"  
 android:layout\_below="@id/spinButton">  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:text="@string/dare" />  
 </LinearLayout>  
  
 <!-- Home Button -->  
 <Button  
 android:id="@+id/homeButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/home"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="20dp" />  
</RelativeLayout>

**3.DareActivity.java:**  
package com.example.truthanddareapp;  
  
import android.os.Bundle;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.Random;  
  
public class DareActivity extends AppCompatActivity {  
  
 private static final String[] *DARE\_QUESTIONS* = {  
 "Do 20 push-ups.",  
 "Sing the chorus of your favorite song.",  
 "Run around the room acting like a chicken.",  
 "Send a funny selfie to the last person you texted.",  
 "Do your best impression of a celebrity of your choice."  
 };  
  
 @Override  
 // Select a random dare question  
 Random random = new Random();  
 String randomDare = *DARE\_QUESTIONS*[random.nextInt(*DARE\_QUESTIONS*.length)];  
  
 // Set the random question  
 dareQuestionTextView.setText(randomDare);  
 }  
}

**4.SpinTheBottleActivity.java:**  
package com.example.truthanddareapp;  
  
import android.animation.ObjectAnimator;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.Random;  
  
public class SpinTheBottleActivity extends AppCompatActivity {  
  
 private ImageView bottleImage;  
   
 public void onAnimationEnd(android.animation.Animator animation) {  
 super.onAnimationEnd(animation);  
 // Calculate the final angle  
 float finalAngle = randomAngle % 360; // Get the final angle within 0-360  
  
 // Show a toast indicating the final direction of the bottle  
 Toast.*makeText*(SpinTheBottleActivity.this, "Bottle stopped at: " + finalAngle + "°", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}

**OUTPUT:**

** **

**WEEK-11**

**AIM:** Develop a Mobile application for simple needs. (Attendance Management system, Bus tracking system and chat bot like etc.).

**DESCRIPTION:**The Truth and Dare Android app is an interactive party game that allows players to engage in classic truth and dare challenges through a digital medium. The game provides a structured yet engaging way for users to play, featuring a spinning bottle mechanism to determine turns, followed by a choice between answering a truth question or completing a dare challenge.

**Key Features**

1. **Randomized Questions and Dares**: The app includes a built-in list of truth questions and dare challenges, offering a mix of fun, embarrassing, and thought-provoking tasks. When a player selects a truth or dare, the app picks a question or challenge randomly, ensuring that each game session feels unique.
2. **Navigation and Home Button**: Users can seamlessly navigate between different sections of the app. The Home button allows users to return to the Spin the Bottle screen, maintaining the game's flow without the need to restart the application.

**CODE:**

**1.activity\_truth.xml:**  
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="16dp">  
  
 <!-- Heading Text -->  
 <TextView  
 android:id="@+id/truthHeading"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="This question below is randomly selected Truth question"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:paddingBottom="20dp" />  
  
 <!-- Random Truth Question -->  
 <TextView  
 android:id="@+id/truthQuestion"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Random Truth Question Will Appear Here"  
 android:textSize="16sp"  
 android:gravity="center" />  
  
</LinearLayout>

**2.TruthActivity.java:**  
package com.example.truthanddareapp;  
  
import android.os.Bundle;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.Random;  
  
public class TruthActivity extends AppCompatActivity {  
  
 private static final String[] *TRUTH\_QUESTIONS* = {  
 "What is your biggest fear?",  
 "Have you ever lied to your best friend?",  
 "What is your most embarrassing moment?",  
 "Who do you have a crush on?",  
 "What’s a secret you’ve never told anyone?"  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_truth*);  
  
 // Reference to the TextView  
 TextView truthQuestionTextView = findViewById(R.id.*truthQuestion*);  
  
 // Select a random truth question  
 Random random = new Random();  
 String randomTruth = *TRUTH\_QUESTIONS*[random.nextInt(*TRUTH\_QUESTIONS*.length)];  
  
 // Set the random question  
 truthQuestionTextView.setText(randomTruth);  
 }  
}

**OUTPUT:**

**WEEK-12**

**AIM:** Develop a Mobile application for simple needs. (Attendance Management system, Bus tracking system and chat bot like etc.).

**DESCRIPTION:**The Truth and Dare Android app is an interactive party game that allows players to engage in classic truth and dare challenges through a digital medium. The game provides a structured yet engaging way for users to play, featuring a spinning bottle mechanism to determine turns, followed by a choice between answering a truth question or completing a dare challenge.

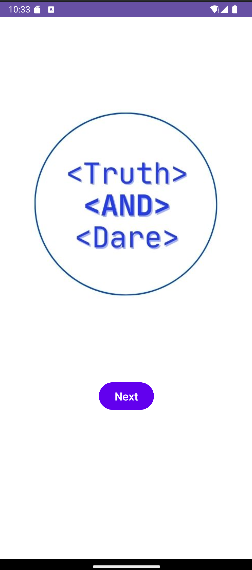
**Gameplay Flow**

1. Launch the App → The user starts the game by tapping the Start button on the opening screen.
2. Spin the Bottle → The app displays a spinning bottle, which rotates randomly before stopping to determine the next player.
3. Choose Truth or Dare → The selected player decides whether they want to answer a truth question or take on a dare challenge.
4. Receive a Random Question or Dare → The app randomly picks a truth question or dare challenge and displays it on the screen.
5. Return to the Main Game → After completing the challenge, the player can return to the Spin the Bottle screen and continue playing.

Technical Implementation:

The app is developed using Java and XML in Android Studio, following best practices for Android app development. The UI layouts are built using RelativeLayout and LinearLayout, ensuring responsive design. Animations are used for the bottle spin effect, adding realism to the gameplay.

Each activity in the app is dedicated to a specific function, such as MainActivity for the start screen, SpinTheBottleActivity for the bottle spinning mechanism, and separate activities for Truth and Dare screens. The app also includes event listeners to handle user interactions, such as button clicks for navigation and spin mechanics.

**OUTPUT**

