1. Develop a mobile application that uses GPS location information

Xml file

*<?***xml version="1.0" encoding="utf-8"***?>*<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
 <**TextView  
 android:id="@+id/showLocation2"  
 android:layout\_width="316dp"  
 android:layout\_height="66dp"  
 android:hint="Location"  
 android:textColor="#E91E63"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.294"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.627"** />  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="273dp"  
 android:layout\_height="66dp"  
 android:gravity="center"  
 android:text="Google Location "  
 android:textColor="#E91E63"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.34"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.069"** />  
  
 <**TextView  
 android:id="@+id/showLocation"  
 android:layout\_width="316dp"  
 android:layout\_height="66dp"  
 android:hint="Location"  
 android:textColor="#E91E63"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.294"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.478"** />  
  
 <**Button  
 android:id="@+id/retrieve\_location\_button"  
 android:layout\_width="244dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="getLocation"  
 android:text="Retrieve Location"  
 android:textColor="#FFFFFF"  
 app:backgroundTint="#E91E63"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.598"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.812"** />  
  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

Java file

**package** com.example.madseventh;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
**import** androidx.core.content.ContextCompat;  
  
**import** android.Manifest;  
**import** android.app.AlertDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.location.Location;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.os.Bundle;  
**import** android.provider.Settings;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 GpsTracker **gpsTracker**;  
 TextView **tvLatitude**,**tvLongitude**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
  
  
 **tvLatitude** = (TextView)findViewById(R.id.***showLocation***);  
 **tvLongitude** = (TextView)findViewById(R.id.***showLocation2***);  
  
 **try** {  
 **if** (ContextCompat.*checkSelfPermission*(getApplicationContext(), android.Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** ) {  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{android.Manifest.permission.***ACCESS\_FINE\_LOCATION***}, 101);  
 }  
 } **catch** (Exception e){  
 e.printStackTrace();  
 }  
 }  
  
 **public void** getLocation(View view){  
 **gpsTracker** = **new** GpsTracker(MainActivity.**this**);  
 **if**(**gpsTracker**.canGetLocation()){  
 **double** latitude = **gpsTracker**.getLatitude();  
 **double** longitude = **gpsTracker**.getLongitude();  
 **tvLatitude**.setText(**"Latitude---"**+String.*valueOf*(latitude));  
 **tvLongitude**.setText(**"Longitude---"**+String.*valueOf*(longitude));  
 }**else**{  
 **gpsTracker**.showSettingsAlert();  
 }  
  
 }  
  
  
  
}

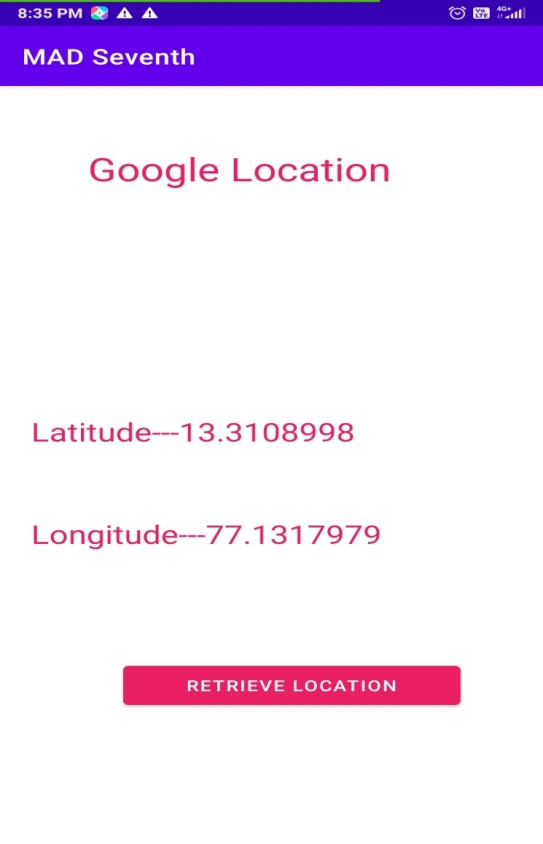
Gps Tracker java file

**package** com.example.madseventh;  
  
**import** android.Manifest;  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.app.Service;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.location.Location;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.os.Bundle;  
**import** android.os.IBinder;  
**import** android.provider.Settings;  
**import** android.util.Log;  
**import** android.widget.Toast;  
  
**import** androidx.core.app.ActivityCompat;  
  
**import** java.security.Provider;  
  
**public class** GpsTracker **extends** Service **implements** LocationListener {  
 **private final** Context **mContext**;  
  
 *// flag for GPS status* **boolean isGPSEnabled** = **false**;  
  
 *// flag for network status* **boolean isNetworkEnabled** = **false**;  
  
 *// flag for GPS status* **boolean canGetLocation** = **false**;  
  
 Location **location**; *// location* **double latitude**; *// latitude* **double longitude**; *// longitude  
  
 // The minimum distance to change Updates in meters* **private static final long *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES*** = 10; *// 10 meters  
  
 // The minimum time between updates in milliseconds* **private static final long *MIN\_TIME\_BW\_UPDATES*** = 1000 \* 60 \* 1; *// 1 minute  
  
 // Declaring a Location Manager* **protected** LocationManager **locationManager**;  
  
 **public** GpsTracker(Context context) {  
 **this**.**mContext** = context;  
 getLocation();  
 }  
  
 **public** Location getLocation() {  
 **try** {  
 **locationManager** = (LocationManager) **mContext**.getSystemService(***LOCATION\_SERVICE***);  
  
 *// getting GPS status* **isGPSEnabled** = **locationManager**.isProviderEnabled(LocationManager.***GPS\_PROVIDER***);  
  
 *// getting network status* **isNetworkEnabled** = **locationManager** .isProviderEnabled(LocationManager.***NETWORK\_PROVIDER***);  
  
 **if** (!**isGPSEnabled** && !**isNetworkEnabled**) {  
 *// no network provider is enabled* } **else** {  
 **this**.**canGetLocation** = **true**;  
 *// First get location from Network Provider* **if** (**isNetworkEnabled**) {  
 *//check the network permission* **if** (ActivityCompat.*checkSelfPermission*(**mContext**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**mContext**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 ActivityCompat.*requestPermissions*((Activity) **mContext**, **new** String[]{android.Manifest.permission.***ACCESS\_FINE\_LOCATION***, Manifest.permission.***ACCESS\_COARSE\_LOCATION***}, 101);  
 }  
 **locationManager**.requestLocationUpdates(  
 LocationManager.***NETWORK\_PROVIDER***,  
 ***MIN\_TIME\_BW\_UPDATES***,  
 ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, **this**);  
  
 Log.*d*(**"Network"**, **"Network"**);  
 *//Toast.makeText(this, "Check the network", Toast.LENGTH\_SHORT).show();* **if** (**locationManager** != **null**) {  
 **location** = **locationManager** .getLastKnownLocation(LocationManager.***NETWORK\_PROVIDER***);  
  
 **if** (**location** != **null**) {  
 **latitude** = **location**.getLatitude();  
 **longitude** = **location**.getLongitude();  
 }  
 }  
 }  
  
 *// if GPS Enabled get lat/long using GPS Services* **if** (**isGPSEnabled**) {  
 **if** (**location** == **null**) {  
 *//check the network permission* **if** (ActivityCompat.*checkSelfPermission*(**mContext**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**mContext**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 ActivityCompat.*requestPermissions*((Activity) **mContext**, **new** String[]{android.Manifest.permission.***ACCESS\_FINE\_LOCATION***, Manifest.permission.***ACCESS\_COARSE\_LOCATION***}, 101);  
 }  
 **locationManager**.requestLocationUpdates(  
 LocationManager.***GPS\_PROVIDER***,  
 ***MIN\_TIME\_BW\_UPDATES***,  
 ***MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***, **this**);  
  
 Log.*d*(**"GPS Enabled"**, **"GPS Enabled"**);  
 *// Toast.makeText(this, "Enable GPS", Toast.LENGTH\_SHORT).show();* **if** (**locationManager** != **null**) {  
 **location** = **locationManager** .getLastKnownLocation(LocationManager.***GPS\_PROVIDER***);  
  
 **if** (**location** != **null**) {  
 **latitude** = **location**.getLatitude();  
 **longitude** = **location**.getLongitude();  
 }  
 }  
 }  
 }  
 }  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return location**;  
 }  
  
 */\*\*  
 \* Stop using GPS listener  
 \* Calling this function will stop using GPS in your app  
 \* \*/* **public void** stopUsingGPS(){  
 **if**(**locationManager** != **null**){  
 **locationManager**.removeUpdates(GpsTracker.**this**);  
 }  
 }  
  
 */\*\*  
 \* Function to get latitude  
 \* \*/* **public double** getLatitude(){  
 **if**(**location** != **null**){  
 **latitude** = **location**.getLatitude();  
 }  
  
 *// return latitude* **return latitude**;  
 }  
  
 */\*\*  
 \* Function to get longitude  
 \* \*/* **public double** getLongitude(){  
 **if**(**location** != **null**){  
 **longitude** = **location**.getLongitude();  
 }  
  
 *// return longitude* **return longitude**;  
 }  
  
 */\*\*  
 \* Function to check GPS/wifi enabled  
 \** ***@return*** *boolean  
 \* \*/* **public boolean** canGetLocation() {  
 **return this**.**canGetLocation**;  
 }  
  
 */\*\*  
 \* Function to show settings alert dialog  
 \* On pressing Settings button will lauch Settings Options  
 \* \*/* **public void** showSettingsAlert(){  
 AlertDialog.Builder alertDialog = **new** AlertDialog.Builder(**mContext**);  
  
 *// Setting Dialog Title* alertDialog.setTitle(**"GPS is settings"**);  
  
 *// Setting Dialog Message* alertDialog.setMessage(**"GPS is not enabled. Do you want to go to settings menu?"**);  
  
 *// On pressing Settings button* alertDialog.setPositiveButton(**"Settings"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog,**int** which) {  
 Intent intent = **new** Intent(Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  
 **mContext**.startActivity(intent);  
 }  
 });  
  
 *// on pressing cancel button* alertDialog.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 dialog.cancel();  
 }  
 });  
 alertDialog.show();  
 }  
  
 @Override  
 **public void** onLocationChanged(Location location) {  
  
 }  
  
 @Override  
 **public void** onProviderDisabled(String provider) {  
 }  
  
 @Override  
 **public void** onProviderEnabled(String provider) {  
 }  
  
 @Override  
 **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  
 }  
  
 @Override  
 **public** IBinder onBind(Intent arg0) {  
 **return null**;  
 }  
}

These lines in manifest files

<**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
<**uses-permission android:name="android.permission. ACCESS\_COARSE\_LOCATION"** />  
<**uses-permission android:name="android.permission.INTERNET"**/>

output



1. Create an application that writes data to the SD card.

Xml file

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#F1C2BE"  
 android:orientation="vertical"  
 tools:context=".MainActivity"**>  
  
  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="110dp"  
 android:background="#E3A0B7"  
 android:text="Reading and Writing to External Storage"  
 android:textSize="24sp"** />  
  
 <**EditText  
 android:id="@+id/myInputText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="150dp"  
 android:background="#D57696"  
 android:ems="10"  
 android:gravity="top|left"  
 android:inputType="textMultiLine"  
 android:lines="5"  
 android:minLines="3"**>  
  
 <**requestFocus** />  
 </**EditText**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:orientation="horizontal"  
 android:weightSum="1.0"**>  
  
 <**Button  
 android:id="@+id/saveExternalStorage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.5"  
 android:text="SAVE"  
 app:backgroundTint="#2C89D3"** />  
  
 <**Button  
 android:id="@+id/getExternalStorage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.5"  
 android:text="READ"  
 app:backgroundTint="#2C89D3"** />  
  
  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:id="@+id/response"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="25sp"  
 android:text=""  
 android:textAppearance="?android:attr/textAppearanceMedium"** />  
  
  
</**LinearLayout**>

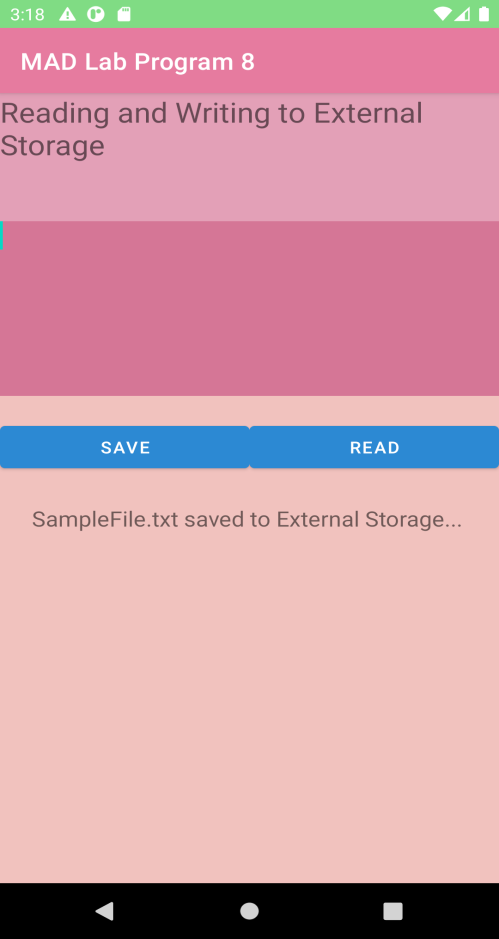
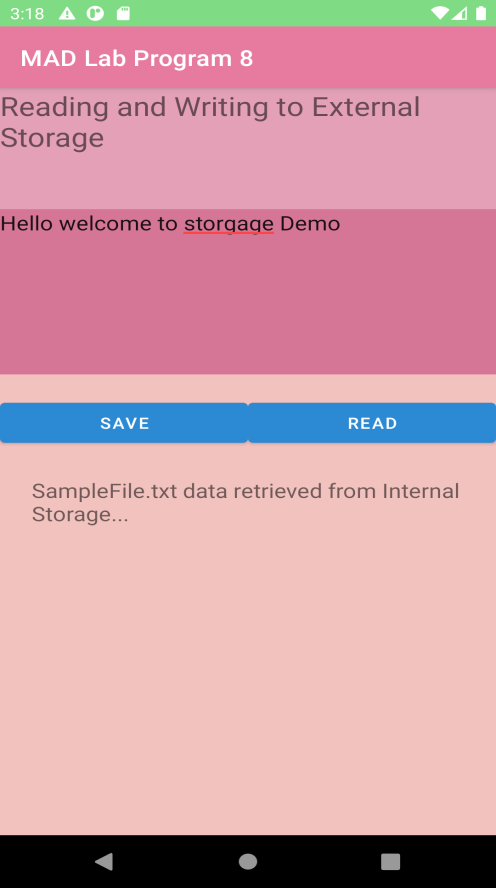
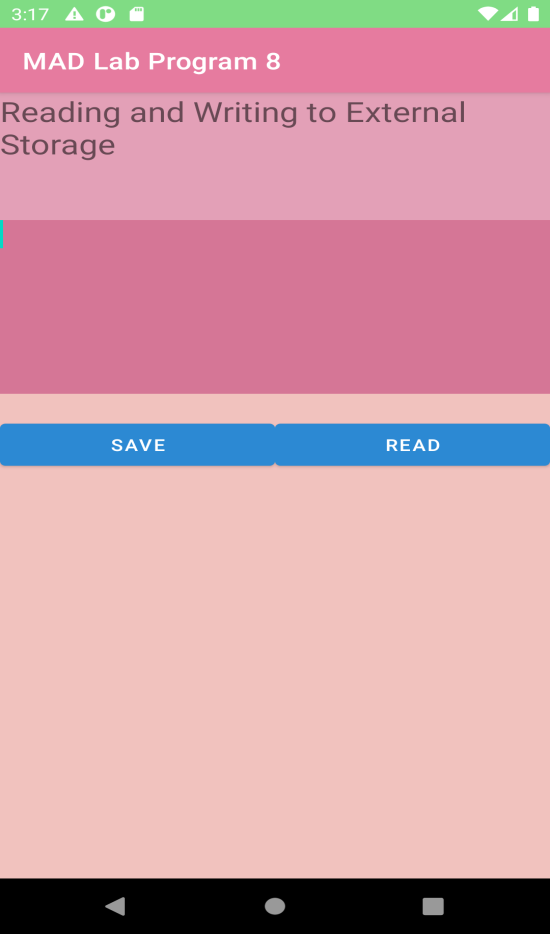
Java File

**package** com.example.mad\_lab8;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.SharedPreferences;  
**import** android.os.Bundle;  
**import** android.os.Environment;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.BufferedReader;  
**import** java.io.DataInputStream;  
**import** java.io.File;  
**import** java.io.FileInputStream;  
**import** java.io.FileOutputStream;  
**import** java.io.FileReader;  
**import** java.io.FileWriter;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 EditText **inputText**;  
 TextView **response**;  
 Button **saveButton**, **readButton**;  
  
 **private** String **filename** = **"SampleFile.txt"**;  
 **private** String **filepath** = **"MyFileStorage"**;  
 File **myExternalFile**;  
 String **myData** = **""**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **inputText** = (EditText) findViewById(R.id.***myInputText***);  
 **response** = (TextView) findViewById(R.id.***response***);  
  
  
 **saveButton** =(Button) findViewById(R.id.***saveExternalStorage***);  
  
 **saveButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **try** {  
 FileOutputStream fos = **new** FileOutputStream(**myExternalFile**);  
 fos.write(**inputText**.getText().toString().getBytes());  
 fos.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **inputText**.setText(**""**);  
 **inputText**.setFocusable(**true**);  
 **response**.setText(**"SampleFile.txt saved to External Storage..."**);  
 }  
 });  
  
  
 **readButton** = (Button) findViewById(R.id.***getExternalStorage***);  
  
  
 **readButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **try** {  
 FileInputStream fis = **new** FileInputStream(**myExternalFile**);  
 DataInputStream in = **new** DataInputStream(fis);  
 BufferedReader br =  
 **new** BufferedReader(**new** InputStreamReader(in));  
 String strLine;  
 **while** ((strLine = br.readLine()) != **null**) {  
 **myData** = **myData** + strLine;  
 }  
 in.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **inputText**.setText(**myData**);  
 **response**.setText(**"SampleFile.txt data retrieved from Internal Storage..."**);  
 }  
 });  
  
 **if** (!*isExternalStorageAvailable*() || *isExternalStorageReadOnly*()) {  
 **saveButton**.setEnabled(**false**);  
 }  
 **else** {  
 **myExternalFile** = **new** File(getExternalFilesDir(**filepath**), **filename**);  
 }  
  
  
 }  
 **private static boolean** isExternalStorageReadOnly() {  
 String extStorageState = Environment.*getExternalStorageState*();  
 **if** (Environment.***MEDIA\_MOUNTED\_READ\_ONLY***.equals(extStorageState)) {  
 **return true**;  
 }  
 **return false**;  
 }  
  
 **private static boolean** isExternalStorageAvailable() {  
 String extStorageState = Environment.*getExternalStorageState*();  
 **if** (Environment.***MEDIA\_MOUNTED***.equals(extStorageState)) {  
 **return true**;  
 }  
 **return false**;  
 }  
}

Manifest file

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.mad\_lab8"**>  
 <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
 <**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MAD\_Lab8"**>  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

Output:

1. Implement an application that creates an alert upon receiving a message.

**activity\_main.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Message!"

android:textSize="30sp"/>

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true"

android:textSize="30sp"/>

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="30dp"

android:layout\_gravity="center"

android:text="notify"

android:textSize="30sp"

/>

</LinearLayout>

**MainActivity.java**

package com.example.administrator.expg09;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity{

Button notify;

EditText e;

@Override

protected void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

notify=(Button) findViewById(R.id.button);

e=(EditText)findViewById(R.id.editText);

notify.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view){

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

PendingIntent pending = PendingIntent.getActivity(MainActivity.this,0,intent,0);

Notification noti= new Notification.Builder(

MainActivity.this).setContentTitle("New Message"). setContentText(e.getText().toString()). setSmallIcon(R.mipmap.ic\_launcher). setContentIntent(pending).build();

NotificationManager manager=(NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

noti.flags|=Notification.FLAG\_AUTO\_CANCEL;

manager.notify(0,noti);

}

});

}

}

Output:



[](https://codingconnect.net/wp-content/uploads/2016/03/Screenshot_2016-03-16-00-41-53-e1458112335383.png)

1. Devise a mobile application that creates alarm clock.

XML file

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:orientation="vertical"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
 <**TimePicker  
 android:id="@+id/timePicker"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"** />  
  
 <**ToggleButton  
 android:id="@+id/toggleButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="20dp"  
 android:checked="false"  
 android:onClick="OnToggleClicked"** />  
  
  
</**LinearLayout**>

Java file

**package** com.example.madlabprogram10;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.TimePicker;  
**import** android.widget.Toast;  
**import** android.widget.ToggleButton;  
  
**import** java.util.Calendar;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 TimePicker **alarmTimePicker**;  
 PendingIntent **pendingIntent**;  
 AlarmManager **alarmManager**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **alarmTimePicker** = (TimePicker) findViewById(R.id.***timePicker***);  
 **alarmManager** = (AlarmManager) getSystemService(***ALARM\_SERVICE***);  
  
 }  
  
 **public void** OnToggleClicked(View view) {  
 **long** time;  
 **if** (((ToggleButton) view).isChecked()){  
 Toast.*makeText*(MainActivity.**this**, **"ALARM ON"**, Toast.***LENGTH\_SHORT***).show();  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.set(Calendar.***HOUR\_OF\_DAY***,**alarmTimePicker**.getCurrentHour());  
 calendar.set(Calendar.***MINUTE***,**alarmTimePicker**.getCurrentMinute());  
 Intent intent = **new** Intent(**this**, AlarmReceiver.**class**);  
 **pendingIntent**=PendingIntent.*getBroadcast*(**this**, 0, intent,0);  
  
 time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));  
 *// the operating system for some the time is measured in milliseconds* **if**(System.*currentTimeMillis*()>time){  
 **if** (calendar.***AM\_PM*** == 0)  
 time = time + (1000\*60\*60\*12);  
 **else** time = time + (1000\*60\*60\*24);  
 }  
 **alarmManager**.setRepeating(AlarmManager.***RTC\_WAKEUP***, time,10000, **pendingIntent**);  
  
 }  
 **else**{  
 **alarmManager**.cancel(**pendingIntent**);  
 Toast.*makeText*(MainActivity.**this**, **"ALARM OFF"**,Toast.***LENGTH\_SHORT***).show();  
 }  
  
  
 }  
}

Alaram reaciver java file

**package** com.example.madlabprogram10;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.TimePicker;  
**import** android.widget.Toast;  
**import** android.widget.ToggleButton;  
  
**import** java.util.Calendar;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 TimePicker **alarmTimePicker**;  
 PendingIntent **pendingIntent**;  
 AlarmManager **alarmManager**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **alarmTimePicker** = (TimePicker) findViewById(R.id.***timePicker***);  
 **alarmManager** = (AlarmManager) getSystemService(***ALARM\_SERVICE***);  
  
 }  
  
 **public void** OnToggleClicked(View view) {  
 **long** time;  
 **if** (((ToggleButton) view).isChecked()){  
 Toast.*makeText*(MainActivity.**this**, **"ALARM ON"**, Toast.***LENGTH\_SHORT***).show();  
 Calendar calendar = Calendar.*getInstance*();  
 calendar.set(Calendar.***HOUR\_OF\_DAY***,**alarmTimePicker**.getCurrentHour());  
 calendar.set(Calendar.***MINUTE***,**alarmTimePicker**.getCurrentMinute());  
 Intent intent = **new** Intent(**this**, AlarmReceiver.**class**);  
 **pendingIntent**=PendingIntent.*getBroadcast*(**this**, 0, intent,0);  
  
 time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));  
 *// the operating system for some the time is measured in milliseconds* **if**(System.*currentTimeMillis*()>time){  
 **if** (calendar.***AM\_PM*** == 0)  
 time = time + (1000\*60\*60\*12);  
 **else** time = time + (1000\*60\*60\*24);  
 }  
 **alarmManager**.setRepeating(AlarmManager.***RTC\_WAKEUP***, time,10000, **pendingIntent**);  
  
 }  
 **else**{  
 **alarmManager**.cancel(**pendingIntent**);  
 Toast.*makeText*(MainActivity.**this**, **"ALARM OFF"**,Toast.***LENGTH\_SHORT***).show();  
 }  
  
  
 }  
}

Output

