ASSIGNMENT 1

MAD LAB FILE

STEPS To Install Android Studio:

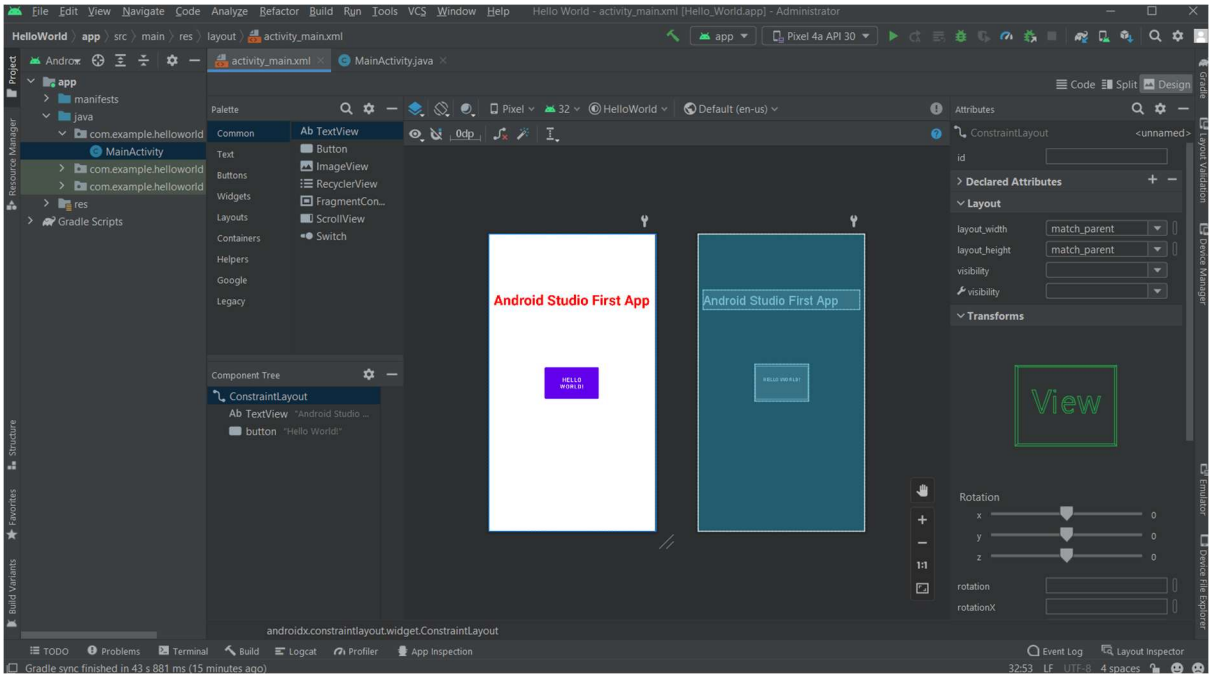
Step1: Open https://developer.android.com/studio in browser.

Step2: Check System minimum Requirements i.e.

- 64-bit Microsoft® Windows® 8/10.
- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
- 8 GB RAM or more.
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution.

Step3: Click on download Android Studio.

Hello World Program:



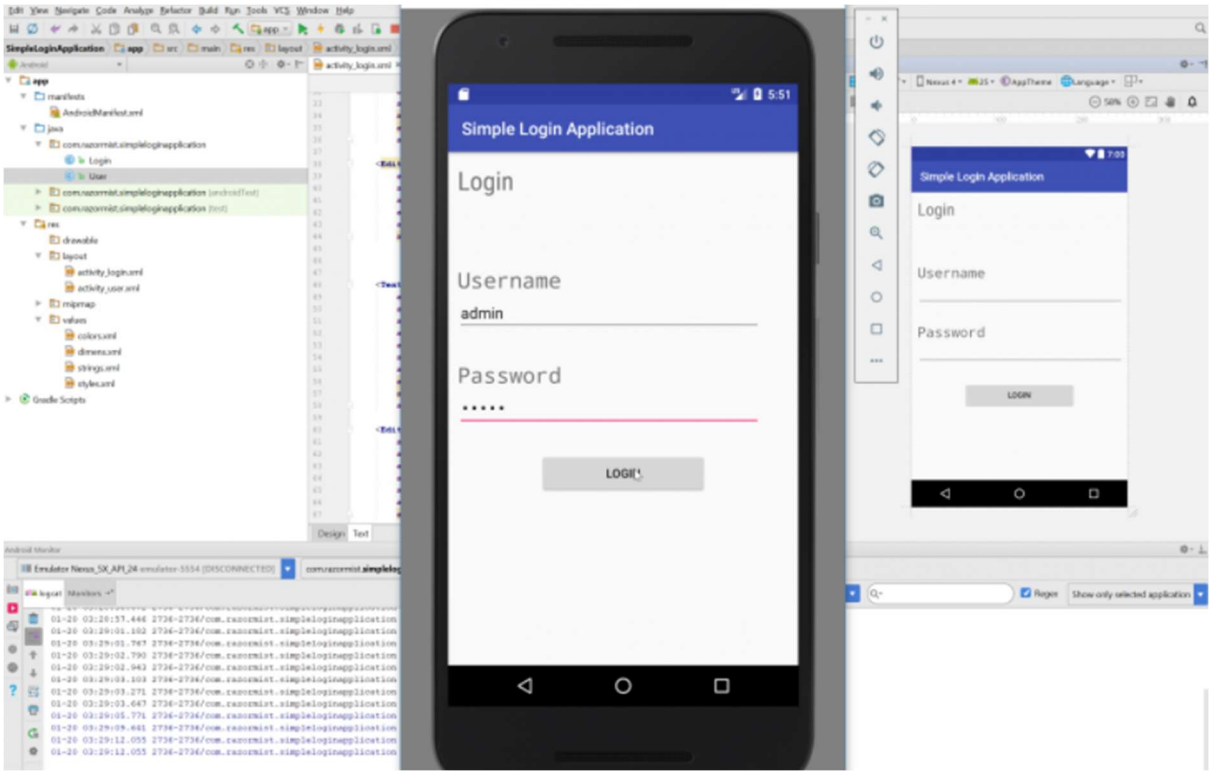
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height= "wrap_content"
        android:textStyle="bold"
        android:text="Android Studio First App"
        android:textColor="#F00"
        android:textSize="35sp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="134dp"
        android:layout_height="90dp"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

ASSIGNMENT 2

AIM: Sample application with Login module and draw main activity as well as connecting activities's layout upto 2 level for any mobile app

Description about aim - First, we define two TextView asking username and password of the user. The password TextView must have inputType set to password. Then define a button with login text

and set its onClick Property. inside the method of onClick get the username and passwords text and match it with the text . Put all together in a class then apply and result is a login page.



XML File:

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

    <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/AppTheme">
        <activity android:name=".Login"
```

```
        android:configChanges="orientation"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".User"
        android:configChanges="orientation"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="com.razormist.simpleloginapplication.User" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
  </application>

</manifest>
```

User.java

```
package com.razormist.simpleloginapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class User extends AppCompatActivity {
```

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

LogIn.java :

```
package com.razormist.simpleloginapplication;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Login extends AppCompatActivity {

    EditText et_username, et_password;
    Button btn_login;

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

```
    Login();
  }

  void Login(){
    et_username = (EditText)findViewById(R.id.et_username);
    et_password = (EditText)findViewById(R.id.et_password);
    btn_login = (Button)findViewById(R.id.btn_login);

    btn_login.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        if(et_username.getText().toString().equals("admin") &&
et_password.getText().toString().equals("admin")){
          Toast.makeText(Login.this, "Username and Password is correct",
Toast.LENGTH_SHORT).show();
          Intent intent = new Intent("com.razormist.simpleloginapplication.User");
          startActivity(intent);
        }else{
          Toast.makeText(Login.this, "Username or Password is incorrect",
Toast.LENGTH_SHORT).show();
        }
      }
    });
  }
}
```

ASSIGNMENT 3

AIM: Develop application that makes use of RSS feed

Description about aimThe RSS stands for Rich Site Summary. RSS is a document that is created by any website. We use Android studio to create an Android application that shows RSS feed. Steps to create RSS feed1)First modify java file to add necessary code & respective XML components.

2)Create a new java file to fetch and parse XML data. 3)Create a new java file to display result of XML. 4)Modify add necessary internet permission. 5)Run the application of RSS Feed.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

<ListView
    android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

Code for AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.exno6" >

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

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    <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>
```

Code for MainActivity.java:

```
package com.example.exno6;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
    List headlines;
    List links;
```

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
   super.onCreate(savedInstanceState);
   new MyAsyncTask().execute();
}

class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
{
   @Override
   protected ArrayAdapter doInBackground(Object[] params)
   {
      headlines = new ArrayList();
      links = new ArrayList();
      try
      {
         URL url = new URL("https://codingconnect.net/feed");
         XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
         factory.setNamespaceAware(false);
         XmlPullParser xpp = factory.newPullParser();

         // We will get the XML from an input stream
         xpp.setInput(getInputStream(url), "UTF_8");
         boolean insideItem = false;

         // Returns the type of current event: START_TAG, END_TAG, etc..
         int eventType = xpp.getEventType();
         while (eventType != XmlPullParser.END_DOCUMENT)
         {
            if (eventType == XmlPullParser.START_TAG)
            {
```

```
                if (xpp.getName().equalsIgnoreCase("item"))
                {
                   insideItem = true;
                }
                else if (xpp.getName().equalsIgnoreCase("title"))
                {
                   if (insideItem)
                      headlines.add(xpp.nextText()); //extract the headline
                }
                else if (xpp.getName().equalsIgnoreCase("link"))
                {
                   if (insideItem)
                      links.add(xpp.nextText()); //extract the link of article
                }
             }
             else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
             {
                insideItem=false;
             }
             eventType = xpp.next(); //move to next element
          }

       }
       catch (MalformedURLException e)
       {
          e.printStackTrace();
       }
       catch (XmlPullParserException e)
       {
          e.printStackTrace();
```

```
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        return null;
    }
    protected void onPostExecute(ArrayAdapter adapter)
    {
        adapter = new ArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1,
headlines);
        setListAdapter(adapter);
    }
  }

  @Override
  protected void onListItemClick(ListView l, View v, int position, long id)
  {
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
  }

  public InputStream getInputStream(URL url)
  {
    try
    {
      return url.openConnection().getInputStream();
    }
    catch (IOException e)
    {
```

```
        return null;
      }
    }
}
```

ASSIGNMENT 4

AIM: Write an android application that draws basic graphical primitives on screen

Description about aim- Steps for draws basic graphical primitives on screen: -1) First create new project file. 2)Go to res folder and select layout - the main xml file . type the code or drag and drop various components. 3)Drag and drop relative layout and change its properties 4)Drag and drop image view and change its properties according to our programs 5)Screen layout can be viewed by clicking graphics layout tab 6)Then include necessary files and override OnCreate() function 7) Now Create Image view and initialize its using id of some components used in the xml program 8)At last result will be the view of the basic drawings.

Code for Activity_main.xml:

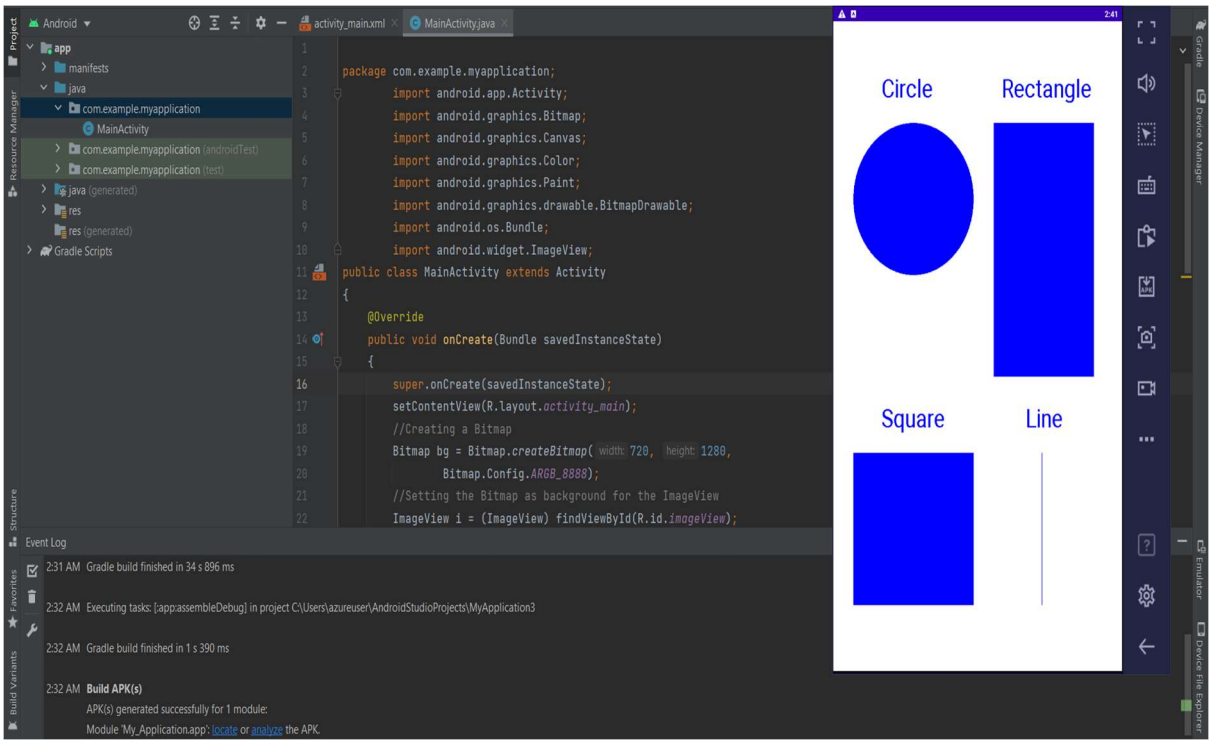
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <ImageView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/imageView" />
</RelativeLayout>
```

Code for MainActivity.java:

```
package com.example.myapplication;
      import android.app.Activity;
      import android.graphics.Bitmap;
      import android.graphics.Canvas;
      import android.graphics.Color;
      import android.graphics.Paint;
      import android.graphics.drawable.BitmapDrawable;
      import android.os.Bundle;
      import android.widget.ImageView;
```

```
public class MainActivity extends Activity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      //Creating a Bitmap
      Bitmap bg = Bitmap.createBitmap(720, 1280,
            Bitmap.Config.ARGB_8888);
      //Setting the Bitmap as background for the ImageView
      ImageView i = (ImageView) findViewById(R.id.imageView);
      i.setBackgroundDrawable(new BitmapDrawable(bg));
      //Creating the Canvas Object
      Canvas canvas = new Canvas(bg);
      //Creating the Paint Object and set its color & TextSize
      Paint paint = new Paint();
      paint.setColor(Color.BLUE);
      paint.setTextSize(50);
      //To draw a Rectangle
      canvas.drawText("Rectangle", 420, 150, paint);
      canvas.drawRect(400, 200, 650, 700, paint);
      //To draw a Circle
      canvas.drawText("Circle", 120, 150, paint);
      canvas.drawCircle(200, 350, 150, paint);
      //To draw a Square
      canvas.drawText("Square", 120, 800, paint);
      canvas.drawRect(50, 850, 350, 1150, paint);
      //To draw a Line
      canvas.drawText("Line", 480, 800, paint);
      canvas.drawLine(520, 850, 520, 1150, paint);
   }
}
```

OUTPUT:

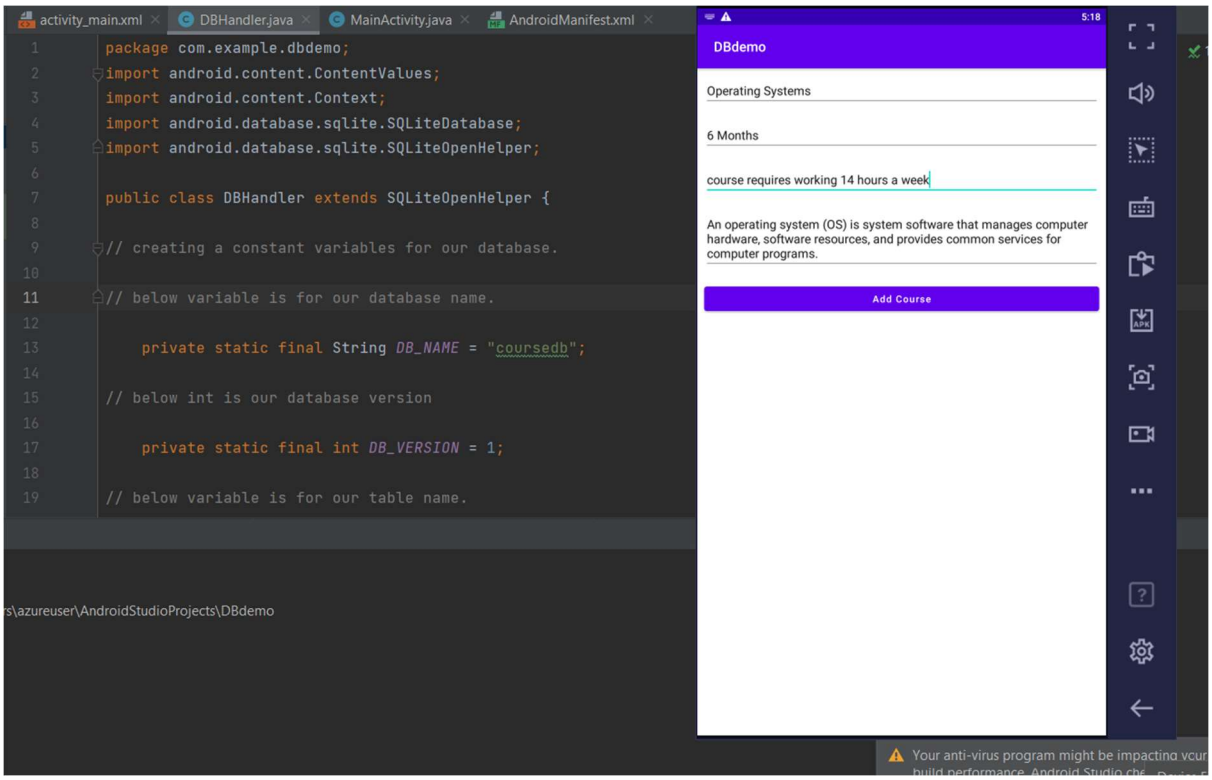


ASSIGNMENT 5

AIM: Create an android app for database creation using SQLite Database

Description about aim- Steps to Create an android app for database creation using SQLite Database - 1) Create a New Project and Name it SQLiteOperations. 2)Open res -> layout -> activity_main.xml (or) main.xml and add its related code. 3) In this step we create a layout in our XML file adding textbox, buttons, edittext. On button onclick is defined which associate it with related function. 4)Now open app -> java -> package -> MainActivity.java and add its related code. In this step create a java class myDbAdapter. java. 5)In this step create another java class Message.class 6)After running the application result will be an android database.

Output:



Code for Activity_main.xml:

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

<LinearLayout

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

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

   android:layout_width="match_parent"

   android:layout_height="match_parent"

   android:orientation="vertical"

   tools:context=".MainActivity">

   <!--Edit text to enter course name-->

   <EditText

      android:id="@+id/idEdtCourseName"

      android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter course Name" />
<!--edit text to enter course duration-->
<EditText
    android:id="@+id/idEdtCourseDuration"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Duration" />
<!--edit text to display course tracks-->
<EditText
    android:id="@+id/idEdtCourseTracks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Tracks" />
<!--edit text for course description-->
<EditText
    android:id="@+id/idEdtCourseDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Description" />
```

```
    <!--button for adding new course-->
    <Button
        android:id="@+id/idBtnAddCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Add Course"
        android:textAllCaps="false" />
</LinearLayout>
```

Code for MainActivity.java:

```
package com.example.dbdemo;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;


public class MainActivity extends AppCompatActivity {

// creating variables for our edittext, button and dbhandler
    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
    private Button addCourseBtn;
    private DBHandler dbHandler;
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
// initializing all our variables.
        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        addCourseBtn = findViewById(R.id.idBtnAddCourse);
// creating a new dbhandler class
// and passing our context to it.
        dbHandler = new DBHandler(MainActivity.this);
// below line is to add on click listener for our add course button.
        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
// below line is to get data from all edit text fields.
                String courseName = courseNameEdt.getText().toString();
                String courseTracks = courseTracksEdt.getText().toString();
                String courseDuration = courseDurationEdt.getText().toString();
                String courseDescription = courseDescriptionEdt.getText().toString();
// validating if the text fields are empty or not.
                if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() &&
                        courseDescription.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter all the data..",
```

```
                Toast.LENGTH_SHORT).show();

            return;

        }

// on below line we are calling a method to add new

// course to sqlite data and pass all our values to it.

        dbHandler.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);

// after adding the data we are displaying a toast message.

        Toast.makeText(MainActivity.this, "Course has been added.",
Toast.LENGTH_SHORT).show();

        courseNameEdt.setText("");

        courseDurationEdt.setText("");

        courseTracksEdt.setText("");

        courseDescriptionEdt.setText("");

      }

    });

  }

}
```

Class DBHandler.java:

```
package com.example.dbdemo;
import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHandler extends SQLiteOpenHelper {

// creating a constant variables for our database.

// below variable is for our database name.
```

```
    private static final String DB_NAME = "coursedb";
// below int is our database version
    private static final int DB_VERSION = 1;
// below variable is for our table name.
    private static final String TABLE_NAME = "mycourses";
// below variable is for our id column.
    private static final String ID_COL = "id";
// below variable is for our course name column
    private static final String NAME_COL = "name";
// below variable id for our course duration column.
    private static final String DURATION_COL = "duration";
// below variable for our course description column.
    private static final String DESCRIPTION_COL = "description";
// below variable is for our course tracks column.
    private static final String TRACKS_COL = "tracks";
// creating a constructor for our database handler.
    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
// below method is for creating a database by running a sqlite query
    @Override
    public void onCreate(SQLiteDatabase db) {
// on below line we are creating
// an sqlite query and we are
```

```
// setting our column names

// along with their data types.

        String query = "CREATE TABLE " + TABLE_NAME + " ("

                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "

                + NAME_COL + " TEXT,"

                + DURATION_COL + " TEXT,"

                + DESCRIPTION_COL + " TEXT,"

                + TRACKS_COL + " TEXT)";

// at last we are calling a exec sql

// method to execute above sql query

        db.execSQL(query);

    }

// this method is use to add new course to our sqlite database.

    public void addNewCourse(String courseName, String courseDuration, String
            courseDescription, String courseTracks) {

// on below line we are creating a variable for

// our sqlite database and calling writable method

// as we are writing data in our database.

        SQLiteDatabase db = this.getWritableDatabase();

// on below line we are creating a

// variable for content values.

        ContentValues values = new ContentValues();

// on below line we are passing all values

// along with its key and value pair.

        values.put(NAME_COL, courseName);
```

```
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);
// after adding all values we are passing
// content values to our table.
        db.insert(TABLE_NAME, null, values);
// at last we are closing our
// database after adding database.
        db.close();
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
// this method is called to check if the table exists already.
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

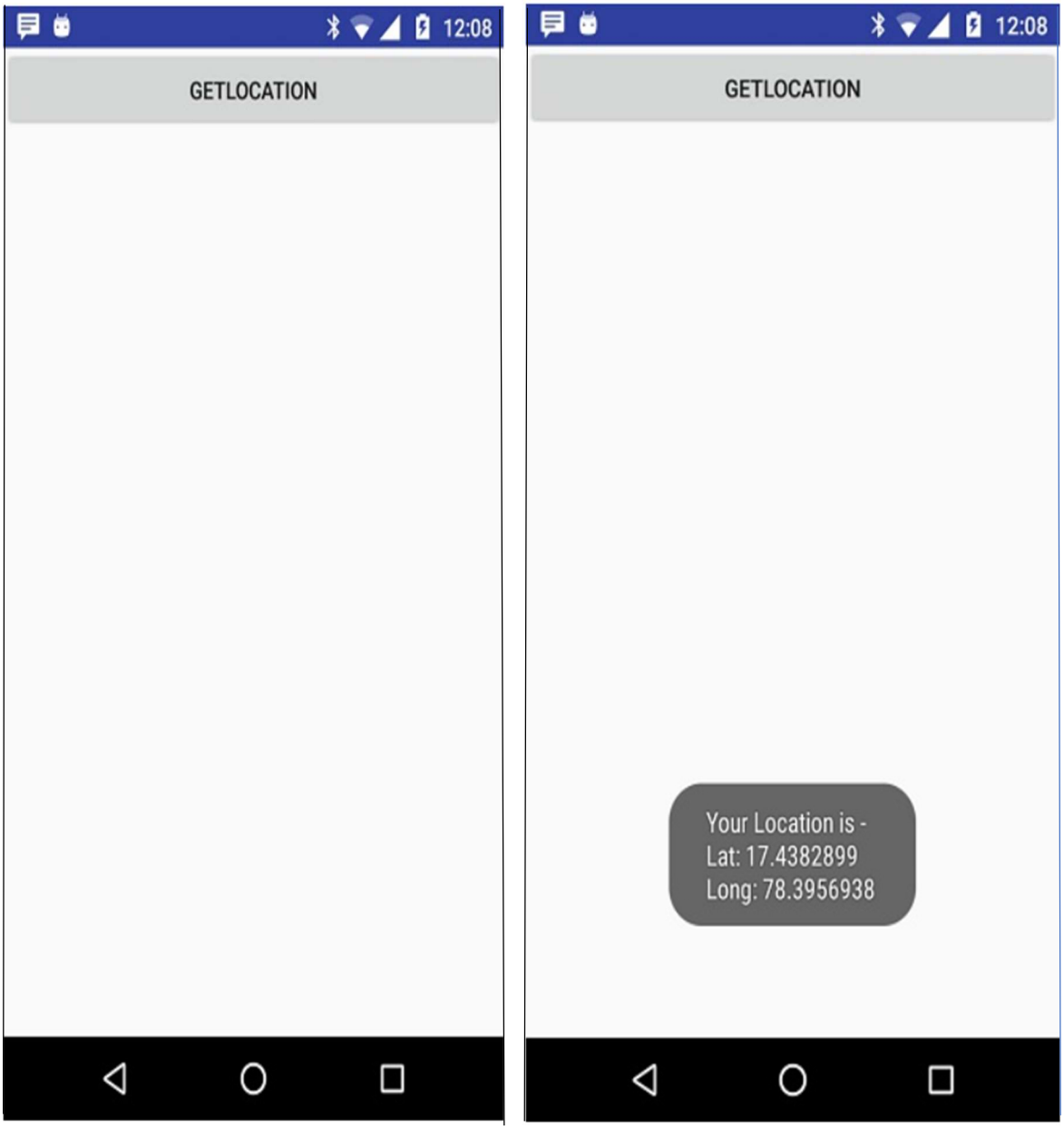
ASSIGNMENT 6

AIM: Develop a native application that use GPS location information

Description about aim- Steps to Develop a native application that use GPS location information1) We use Android studio IDE to create an Android application and name it. 2)add src/GPSTracker.java file

and add required code. 3)Modify src/MainActivity.java file and add required code to take care of getting current location and its equivalent address. 4)Modify layout XML file res/layout/activity_main.xml to add all GUI components which include three buttons and two text views to show location/address. 5)Modify res/values/strings.xml to define required constant values 6)Modify AndroidManifest.xml. 7)Run the application to launch Android emulator and verify the result of the changes done in the application.

Output:



Code for Activity_main.xml:

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

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

```
  android:layout_width = "fill_parent"
  android:layout_height = "fill_parent"
  android:orientation = "vertical" >

  <Button
    android:id = "@+id/button"
    android:layout_width = "fill_parent"
    android:layout_height = "wrap_content"
    android:text = "getlocation"/>

</LinearLayout>
```

Code for MainActivity.java:

```
package com.javacodegeeks.android.lbs;

import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class LbsGeocodingActivity extends Activity {

   private static final long MINIMUM_DISTANCE_CHANGE_FOR_UPDATES = 1; // in Meters
   private static final long MINIMUM_TIME_BETWEEN_UPDATES = 1000; // in Milliseconds
```

```
protected LocationManager locationManager;

protected Button retrieveLocationButton;

@Override
public void onCreate(Bundle savedInstanceState) {

   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);

   retrieveLocationButton = (Button) findViewById(R.id.retrieve_location_button);

   locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

   locationManager.requestLocationUpdates(
         LocationManager.GPS_PROVIDER,
         MINIMUM_TIME_BETWEEN_UPDATES,
         MINIMUM_DISTANCE_CHANGE_FOR_UPDATES,
         new MyLocationListener()
   );

retrieveLocationButton.setOnClickListener(new OnClickListener() {
      @Override
      public void onClick(View v) {
         showCurrentLocation();
      }
});

}

protected void showCurrentLocation() {
```

```
    Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

    if (location != null) {
        String message = String.format(
                "Current Location \n Longitude: %1$s \n Latitude: %2$s",
                location.getLongitude(), location.getLatitude()
        );
        Toast.makeText(LbsGeocodingActivity.this, message,
                Toast.LENGTH_LONG).show();
    }

}

private class MyLocationListener implements LocationListener {

    public void onLocationChanged(Location location) {
        String message = String.format(
                "New Location \n Longitude: %1$s \n Latitude: %2$s",
                location.getLongitude(), location.getLatitude()
        );
        Toast.makeText(LbsGeocodingActivity.this, message, Toast.LENGTH_LONG).show();
    }

    public void onStatusChanged(String s, int i, Bundle b) {
        Toast.makeText(LbsGeocodingActivity.this, "Provider status changed",
                Toast.LENGTH_LONG).show();
    }

    public void onProviderDisabled(String s) {
        Toast.makeText(LbsGeocodingActivity.this,
```

```
                "Provider disabled by the user. GPS turned off",
                Toast.LENGTH_LONG).show();
        }

        public void onProviderEnabled(String s) {
           Toast.makeText(LbsGeocodingActivity.this,
                "Provider enabled by the user. GPS turned on",
                Toast.LENGTH_LONG).show();
        }

    }

}
```

Code for AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.javacodegeeks.android.lbs"
    android:versionCode="1"
    android:versionName="1.0">

   <application android:icon="@drawable/icon" android:label="@string/app_name">
      <activity android:name=".LbsGeocodingActivity"
                android:label="@string/app_name">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>

   </application>
```

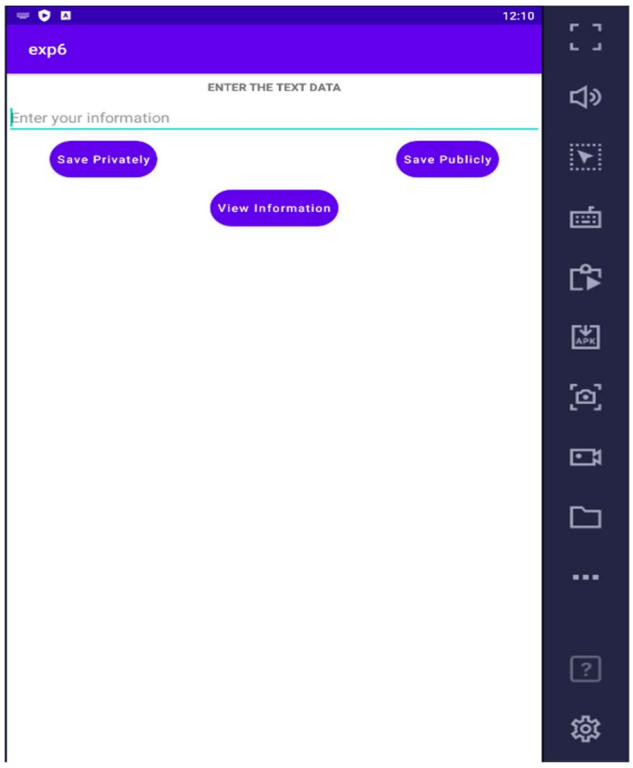
```
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
 <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />


    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

ASSIGNMENT 7

AIM: Implement an applcation that writes data to the SD card

Description about aim- Creating a New project: ▪ Open Android Studio and then click on File -> New -> New project. Designing layout for the Android Application: ▪ Click on app -> res -> layout -> activity_main.xml. ▪ Then delete the code which is there and type the code . ▪ Now click on Design and your application will completed. Adding permissions in Manifest for the Android Application: ▪ Click on app -> manifests -> AndroidManifest.xml ▪ Now include the WRITE_EXTERNAL_STORAGE permissions in the AndroidManifest.xml file Java Coding for the Android Application: o Click on app -> java -> com.example.exno9 -> MainActivity. • So now the Coding part is also completed. Now run the application to see the output

Output:



Code for Activity_main.xml:

```
package com.example.exp6;

import android.Manifest;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    // After API 23 the permission request for accessing external storage is
changed
    // Before API 23 permission request is asked by the user during
installation of app
    // After API 23 permission request is asked at runtime
    private int EXTERNAL_STORAGE_PERMISSION_CODE = 23;
    EditText editText;

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

        // findViewById return a view, we need to cast it to EditText View
        editText = (EditText) findViewById(R.id.editText_data);
    }

    public void savePublicly(View view) {
        // Requesting Permission to access External Storage
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                EXTERNAL_STORAGE_PERMISSION_CODE);
        String editTextData = editText.getText().toString();

        // getExternalStoragePublicDirectory() represents root of external
storage, we are using DOWNLOADS
```

```
        // We can use following directories: MUSIC, PODCASTS, ALARMS,
RINGTONES, NOTIFICATIONS, PICTURES, MOVIES
        File folder =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS)
;

        // Storing the data in file with name as geeksData.txt
        File file = new File(folder, "geeksData.txt");
        writeTextData(file, editTextData);
        editText.setText("");
    }

    public void savePrivately(View view) {
        String editTextData = editText.getText().toString();

        // Creating folder with name GeekForGeeks
        File folder = getExternalFilesDir("GeeksForGeeks");

        // Creating file with name gfg.txt
        File file = new File(folder, "gfg.txt");
        writeTextData(file, editTextData);
        editText.setText("");
    }

    public void viewInformation(View view) {
        // Creating an intent to start a new activity
        Intent intent = new Intent(MainActivity.this,
ViewInformationActivity.class);
        startActivity(intent);
    }

    // writeTextData() method save the data into the file in byte format
    // It also toast a message "Done/filepath_where_the_file_is_saved"
    private void writeTextData(File file, String data) {
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream(file);
            fileOutputStream.write(data.getBytes());
            Toast.makeText(this, "Done" + file.getAbsolutePath(),
Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileOutputStream != null) {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
```

```
                }
            }
        }
    }
}
```

Code for AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exp6">

    <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.Exp6">
        <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>
```

Code for activity_view_information.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".ViewInformationActivity">

    <Button
```

```
        android:id="@+id/showButton_public"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_below="@+id/textView_get_saved_data"
        android:layout_marginEnd="48dp"
        android:layout_marginTop="8dp"
        android:background="@drawable/button_layout"
        android:onClick="showPublicData"
        android:padding="8dp"
        android:text="@string/show_button_public"
        android:textAllCaps="false"
        android:textColor="@color/textColor" />

    <Button
        android:id="@+id/showButton_private"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textView_get_saved_data"
        android:layout_marginStart="48dp"
        android:layout_marginTop="8dp"
        android:layout_toEndOf="@+id/showButton_public"
        android:background="@drawable/button_layout"
        android:onClick="showPrivateData"
        android:padding="8dp"
        android:text="@string/show_button_private"
        android:textAllCaps="false"
        android:textColor="@color/textColor" />

    <Button
        android:id="@+id/goBackButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/showButton_public"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:background="@drawable/button_layout"
        android:onClick="back"
        android:padding="8dp"
        android:text="@string/back_button"
        android:textAllCaps="false"
        android:textColor="@color/textColor" />

    <TextView
        android:id="@+id/textView_get_saved_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/textView_saved_data"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:hint="@string/saved_information" />

    <TextView
        android:id="@+id/textView_saved_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:text="@string/text_view_saved_data"
        android:textAllCaps="true"
        android:textStyle="bold" />

</RelativeLayout>
```

Code for ViewInformationAcitity.java:

```
package com.example.exp6;

import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;

public class ViewInformationActivity extends AppCompatActivity {

    TextView textView;

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

        // findViewById returns a view, we need to cast it into TextView
        textView = (TextView) findViewById(R.id.textView_get_saved_data);
    }

    public void showPublicData(View view) {
        // Accessing the saved data from the downloads folder
```

```
        File folder =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS)
;

        // geeksData represent the file data that is saved publicly
        File file = new File(folder, "geeksData.txt");
        String data = getdata(file);
        if (data != null) {
            textView.setText(data);
        } else {
            textView.setText("No Data Found");
        }
    }

    public void showPrivateData(View view) {

        // GeeksForGeeks represent the folder name to access privately saved
data
        File folder = getExternalFilesDir("GeeksForGeeks");

        // gft.txt is the file that is saved privately
        File file = new File(folder, "gfg.txt");
        String data = getdata(file);
        if (data != null) {
            textView.setText(data);
        } else {
            textView.setText("No Data Found");
        }
    }

    public void back(View view) {
        Intent intent = new Intent(ViewInformationActivity.this,
MainActivity.class);
        startActivity(intent);
    }

    // getdata() is the method which reads the data
    // the data that is saved in byte format in the file
    private String getdata(File myfile) {
        FileInputStream fileInputStream = null;
        try {
            fileInputStream = new FileInputStream(myfile);
            int i = -1;
            StringBuffer buffer = new StringBuffer();
            while ((i = fileInputStream.read()) != -1) {
                buffer.append((char) i);
            }
            return buffer.toString();
```

```
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        return null;
    }
}
```