Ambekeshwar Group Of Institutions

Technology & Management,
Lucknow

Session: 2024-25

PRACTICAL FILE

Branch:- CSE 3rd Year | 6th Sem

Subject: Development of Android Applications

Name: Suraj Arya

Date:

	Name	Signature
Subject Teacher	Miss. Fiza Hussain	…………………………….
Academic Co-Ordinator	Mr. Ashish Mishra	…………………………….

INDEX

Program: 01

Objective:

Install the Android Studio and set up the Development Environment.

Software Requirement:

- Environment Used
- Windows 7/8/10
- JDK 6
- •Android SDK for Windows
- Eclipse IDE

Setting Up Environment:

a) Installing JDK6 (Java Development Kit)

- download the JDK from:
- http://www.oracle.com/technetwork/java/javase/downloads/index.html
- To install the JDK, double-click on the downloaded exe file.
- Double Click the icon of the downloaded exe from the downloaded location.

You will see JDK 6 update window as shown below.

Now a **"License Agreement"** window opens. Just read the agreement and click accept and go further.

Now a **"Custom Setup"** window opens.

Clicking the **"OK"** button starts the installation.

The next window asks to install Runtime Environment.

Click the **"Next"** button.

Click the **"OK"** button starts the installation.

Click the **"Finish"** button to exit the installation process.

(b) Installing Android SDK Tools & API

- Download Android SDK from
- http://developer.android.com/sdk/index.html. as shown below.

Download Android Studio

Double-click on the SDK Installer exe file to install.

Select Location and Click Next

- Click **Add,** in the top-right corner.
- In the Add Repository dialog that appears, enter “ADT plugin” for the Name and the following URL for the Location:

URL for the Location:

https://dl-ssl.google.com/android/eclipse/

- Click **OK**.

Read and accept the license agreements, then click **Finish.**

Program: 02

Objective:

Write a program to demonstrate activity (Application Life Cycle).

Source Code

Creating a New project:

- Open Android Stdio and then click on File > New New project.
- Then type the Application name as "activitylifecycle" and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.
- will take some time to build and load the project.
- After completion it will look as given below.

Designing layout for the Android Application:

- Click on app->res layout activity_main.xml.
- Now click on Text as shown below.
- Type the code as given below.

Activity_main.xml

```
<RelativeLayout 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"
        tools:context="MainActivity">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
RelativeLayout>
```

MainActivity.java

```
Package com.example.activitylifecycle;
Import android.os.Bundle;
Import android.app.Activity;
Import android.view.Menu;
Import android.util.Log;
public class MainActivity extends Activity
{
```

Program: 03

Objective:

Write a program to store and fetch data from SQL life database.

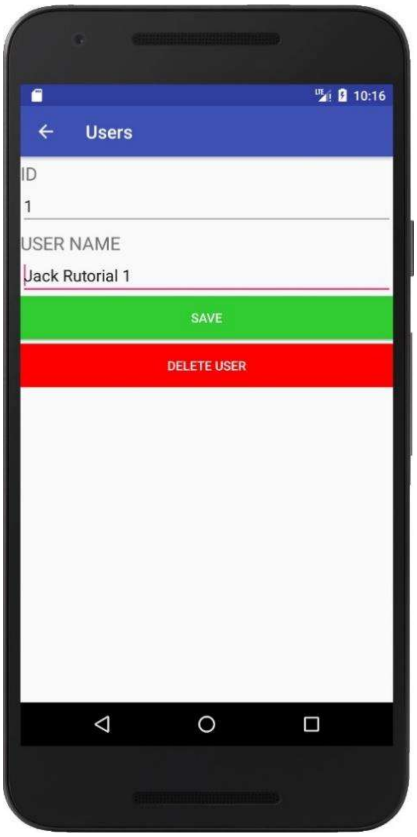
Source Code: SQLite database, insert and show the details from the SQLite database into an android listview using the SQLiteOpenHelper class.

(a) Activity Main.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 "SQIExample MainActivity">
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android layout centerVertical-"true"
        android:orientation="vertical"
        android padding "16dp">
<TextView
        android:layout_width="match_parent"
        android layout_height="wrap_content"
        android:layout_marginBottom="12dp"
        android:text="Add a new Employee"
        android textAlignment="center"
        android:textAppearance"@style Base. TextAppearance AppCompat Large>
<EditText
        android id-"+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android orientation="horizontal"
        android:padding "8dp">
<LinearLayout
        android:layout_width="230dp"
        android:layout_height="wrap_content"
        android:orientation="vertical">
<TextView
        android:id="@+id/textViewName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="5dp"
        android:text="Mayank"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.
        Large" />
<TextView
        android:id="@+id/textViewDepartment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Compuer"
        android:textAppearance-"@style/Base.TextAppearance. AppCompat.
        Mediun" />
<TextView
        android:id="@+id/textViewSalary"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="INR 50000"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.
        Medium" />
<TextView
        android:id="@+id/textViewJoiningDate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android sext "2021-3-01 11:05:17">>
</LinearLayout>
<LinearLayout
        android layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
<Button
        android:id="@+id/buttonEditEmployee"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background-"@color/colorPrimary"
        android:text="Edit">>
        android:layout_margin="5dp"
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:id="@+id/buttonDeleteEmployee"
        android:background="@color/colorAccent"
        android text "Delete"/>
<LinearLayout>
```

```
<LinearLayout>
```

Output



ID

1

USER NAME

Jack Rutorial 1

Program: 04

Objective:

Write a program to create a text file in a external memory.

Source Code:

Store text data into the external storage and fetch to see that data.

1. Create a new project in Android Studio
2. Access Permission to External Storage

To read and write data to external storage, the app required WRITE_EXTERNAL_STORAGE and READ_EXTERNAL_STORAGE system permission. These permissions are added to the AndroidManifest.xml file. Add these permissions just after the package name.

```
<manifest... >
<uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
        <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<application ... >
        <activity android:name=".MainActivity" ..
</activity>
</application>
</manifest>
```

3. Before creating the layout and corresponding java files let's add a few string attributes which we are using in our layout files

Go to app res > values> string.xml and insert the following code.

```
<resources>
…
<string name="text_view_data">Enter the Text Data</string>
<string name="edit_text_data">Enter your information</string>
<string name="view_button">View Information</string>
<string name="save_button_public">Save Publicly</string>
<string name="save_button_private">Save Privately</string>
<string name="text_view_saved_data">Saved Text Data</string>
<string name="saved_information">Click to view saved
information</string>
<string name="back_button">Go Back</string>
<string name="show_button_public">Public Data</string>
<string name="show_button_private">Private Data</string>
…
</resources>
}
```

```
// 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 writeText Data (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();
}

}

}

}

}
```

5. Create a new Empty Activity

Create a new activity and name it ViewInformationActivity. We create a layout for this activity similar to the MainActivity layout. activity_view_information.xml layout code.

```
<?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_below="@+id/textView_get_saved_data"

android:layout_alignParentEnd="true"
```

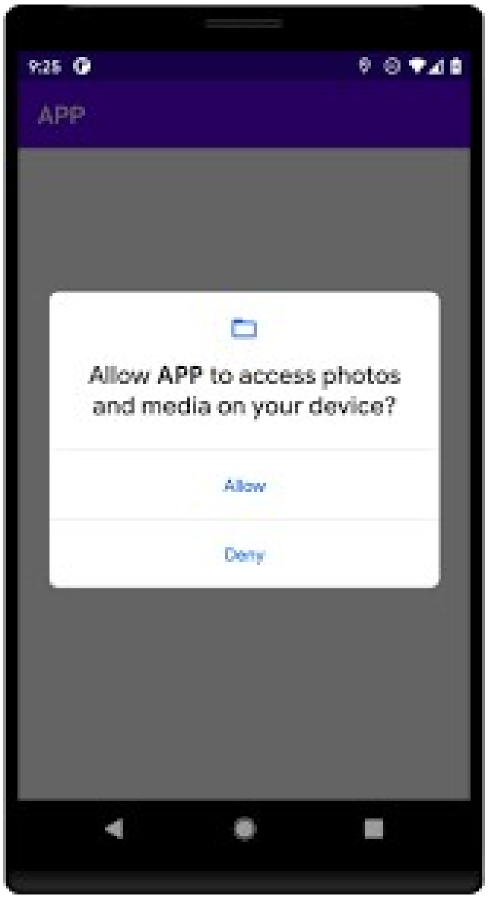
```
android:layout_marginEnd="48dp"

android:layout_marginTop="8dp"

android:background="@drawable/button_layout"

) catch (IOException e) {
e.printStackTrace();
}
}
}
return null;
}
}
```

Output:



Allow APP to access photos and media on your device?

Program: 05

Objective:

Write a program to demonstrate listview.

Source Code: We will define a ListView in the main layout XML. file activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.Constraint Layout
        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:background="#FFEB3B"
        tools:context="com.example.android.MainActivity">
        android:layout_width="match_parent"
        android:layout_height="match_parent"
<ListView>
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:divider="@android:color/black"
        android:dividerHeight="1dp"/>
</android.support.constraint.ConstraintLayout>
```

So we need a dataset and a View into which the dataset will be converted by the Adapter.

Here we have a simple Array with festivals names in it:

```
String[] festivals - (
        "Diwali",
        "Holi",
        Christmas",
        "Eid",
        "Baisakhi",
        "Halloween"
);
```

- As our data set has simple text values, so we can define a simple TextView to hold these values and populate the ListView. Does it sound confusing? Let it sink in.

- If our dataset would have had, an image and some text along with it, then we can also define a TextView along with an ImageView to display the data in the List.

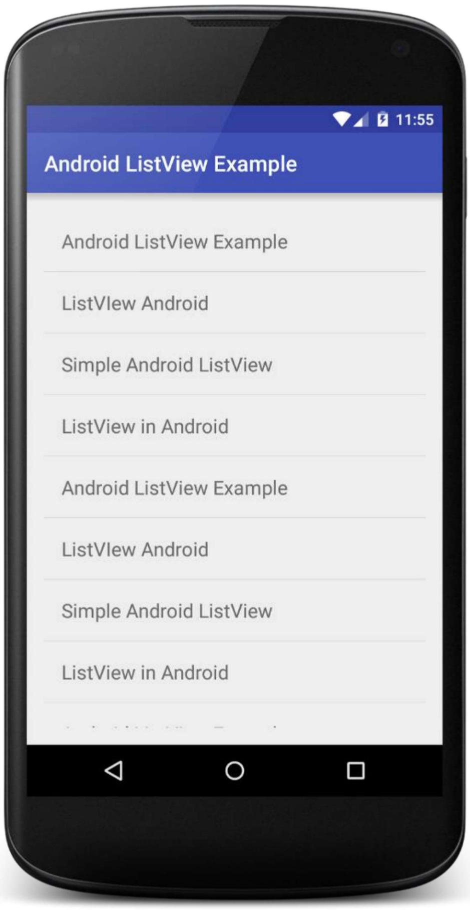
- So now we will create a new XML file, with name list_item.xml in the layout folder, and add a TextView in it like this,

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/textView"
android:layout_width="wrap_content"
android:textStyle="bold"
android:layout_marginTop="5dp"

// TODO Auto-generated method stub

/*appending Happy with festival name */
String value "Happy" adapter.getItem(position);
/ Display the Toast +/
Toast.makeText(getApplicationContext(), value,
Toast.LENGTH_SHORT).show();
)
));
)
)
```

Output:



11:55

Android ListView Example

Android ListView Example

ListVIew Android

Simple Android ListView

ListView in Android

Android ListView Example

ListVIew Android

Simple Android ListView

ListView in Android

Program: 06

Objective:

Write a program to demonstrate different types of layouts.

a) Create Application by Using Building Blocks for Android Application Design by using Linear Layout

b) Create Application by Using Building Blocks for Android Application Design by using Relative Layout.

c) Create Application by Using Building Blocks for Android Application Design by using Absolute Layout.

Source Code:

Creating a New project:

- Open Android Stdio and then click on File-> New -> New project.
- Then type the Application name as "linear_layout" and click Next.
- Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.
- It will take some time to build and load the project.
- After completion it will look as given below.

Designing layout for the Android Application:

- Click on app -> res-> layout-> activity_main.xml.
- Now click on Text as shown below.
- Type the code as given below.

Activity_linear_layout_app.xml

```
(LinearLayout
xmlns:android="http://schemas.android.com/apk/res/
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >
<Button
        android:id="@+id/Apple"
        android:text="Apple"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" /
                                                                <
<Button
        android:id="@+id/Mango"
        android:text="Mango"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
<Button
        android:id="@+id/Banana"
        android:text="Banana"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Activity_linear_layout_app.xml File on Setting Horizontal

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/and
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
<Button
        android:text="Apple"
        android:id="@+id/Apple"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<Button
        android:id="@+id/Mango"
        android:text="Mango"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<Button
        android:id="@+id/Banana"
        android:text="Banana"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```