LAB MANUAL

SMART DEVICE PROGRAMMING LAB

6138

Contents

Introduction

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008. The source code for Android is available under free and open source software licenses.

Android is an operating system based on Linux with a Java programming interface. Android provides a rich application framework that allows to build innovative apps and games for mobile devices in a Java language environment. The Android Software Development Kit (Android SDK) provides all necessary tools to develop Android applications. The Android SDK contains tools to create, compile and package Android application. The Android SDK also provides an Android device emulator, so that Android applications can be tested without a real Android phone. We can create Android virtual devices (AVD) via the Android SDK, which run in this emulator. The Android SDK contains the Android debug bridge (adb) tool which allows to connect to a virtual or real Android device.

Google offers the Google Play service in which programmers can offer their Android application to Android users. Google phones include the Google Play application which allows installing applications. Google Play also offers an update service ie. if a programmer uploads a new version of his application to Google Play, this service will notify existing users that an update is available and allow to install it.

Chapter-1

Installing Android Studio on Ubuntu

With the advancement of smart phones in the recent years, Android has become one of the biggest phone platforms and all the tools required to build Android applications are also freely available. Android Studio is an Integrated Development Environment (IDE) for developing Android applications.

Installation on Ubuntu 15.04

We can install Android Studio in two ways.

➢ The first methodis to set up the required repository and install it,
➢ The second method is to download it from the official Android site and install it locally.

In the first method, we will be setting up the repo using command line and install it. Before proceeding, we need to make sure that we have JDK version1.6 or greater installed.

To install JDK 1.8.

Step1 : $ sudo add-apt-repository ppa:webupd8team/java

Step2: $ sudo apt-get update

Step 3: $ sudo apt-get install oracle-java8-installer oracle-java8-set-default

Verify if java installation was successful:

Step4: :~$ java -version

To install Android Studio

Step1: Setup the repo for installing Android Studio

$ sudo apt-add-repository ppa:paolorotolo/android-studio

```
poornima@poornima-Lenovo: ~
File  Edit  View  Search  Terminal  Help
poornima@poornima-Lenovo:~$ sudo add-apt-repository ppa:paolorotolo/android-stud
io
[sudo] password for poornima:
 Automatic builds for Android Studio by Google packaged for Ubuntu.
 More info: https://launchpad.net/~paolorotolo/+archive/ubuntu/android-studio
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmpb64x0bql/secring.gpg' created
gpg: keyring `/tmp/tmpb64x0bql/pubring.gpg' created
gpg: requesting key 7B9B74AA from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpb64x0bql/trustdb.gpg: trustdb created
gpg: key 7B9B74AA: public key "Launchpad PPA for Paolo Rotolo" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
OK
poornima@poornima-Lenovo:~$ ▮
```
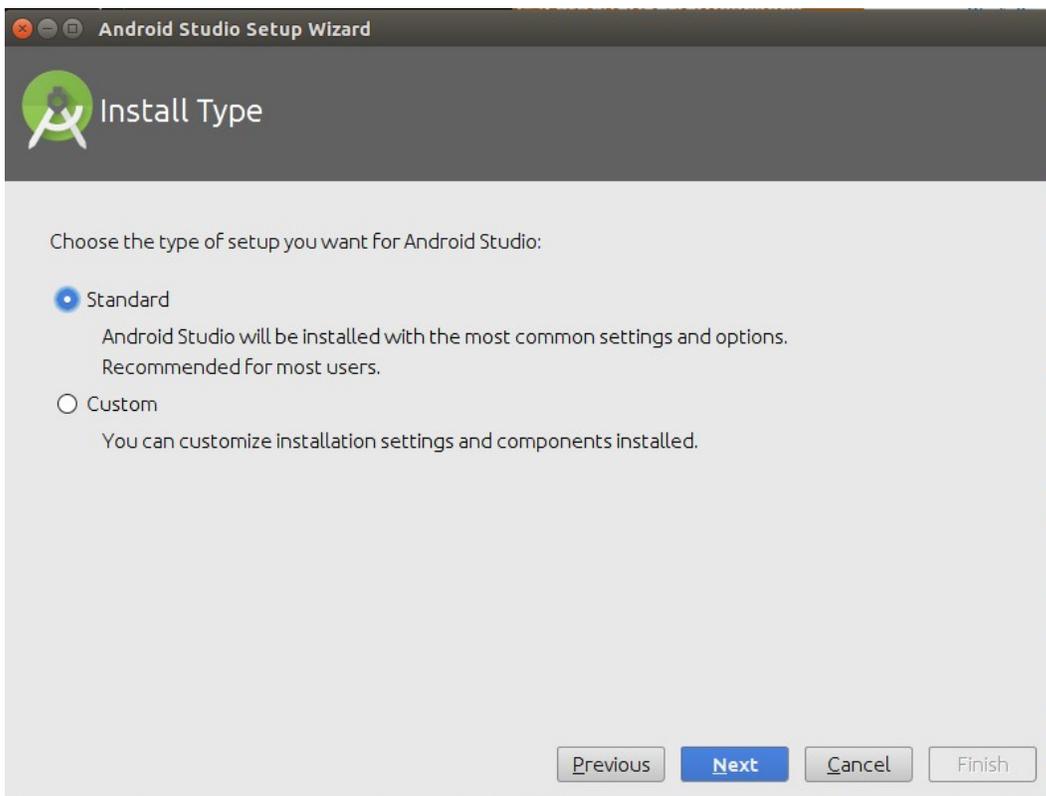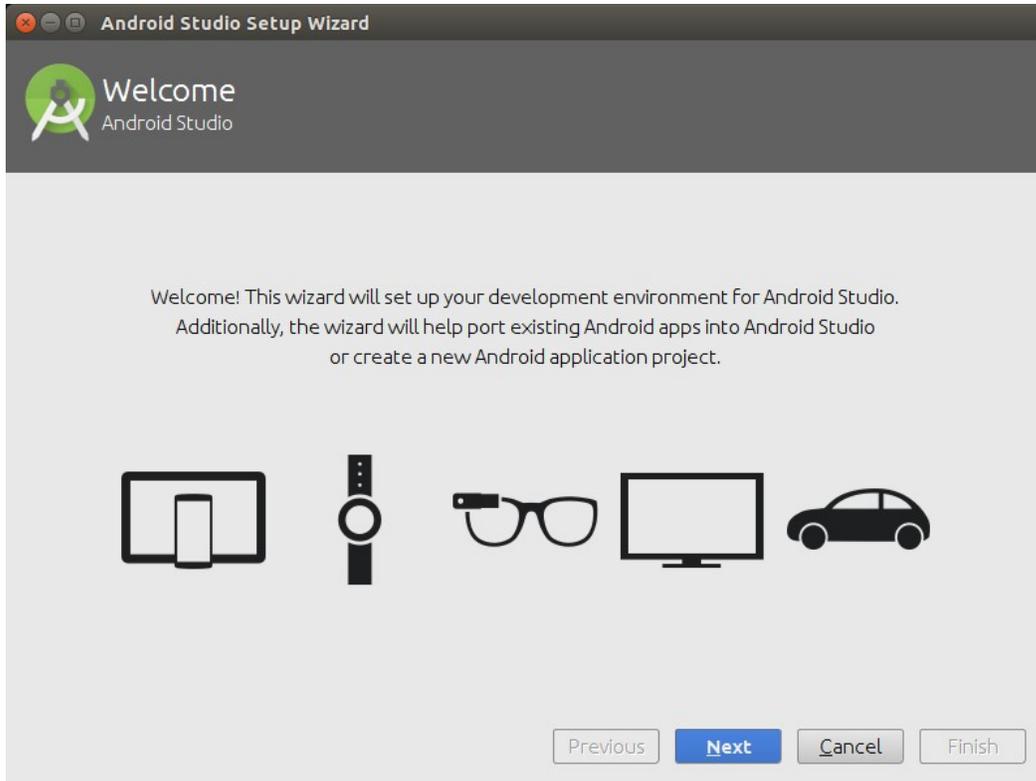
Step2: $ sudo apt-get update

Step3: $ sudo apt-get install android-studio

      Above install command will install android-studio in the directory /opt.

      Now, run the following command to start the setup wizard:

Step4: $ /opt/android-studio/bin/studio.sh

      This will invoke the setup screen. Following are the screen shots that follow to set up Android studio:

Android Studio Setup Wizard

Welcome
Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.

Previous | **Next** | Cancel | Finish

Android Studio Setup Wizard

Install Type

Choose the type of setup you want for Android Studio:

◉ Standard

Android Studio will be installed with the most common settings and options.
Recommended for most users.

○ Custom

You can customize installation settings and components installed.

Previous | **Next** | Cancel | Finish

Once you press the Finish button, Licence agreement will be displayed. After you accept the licence, it starts downloading the required components.
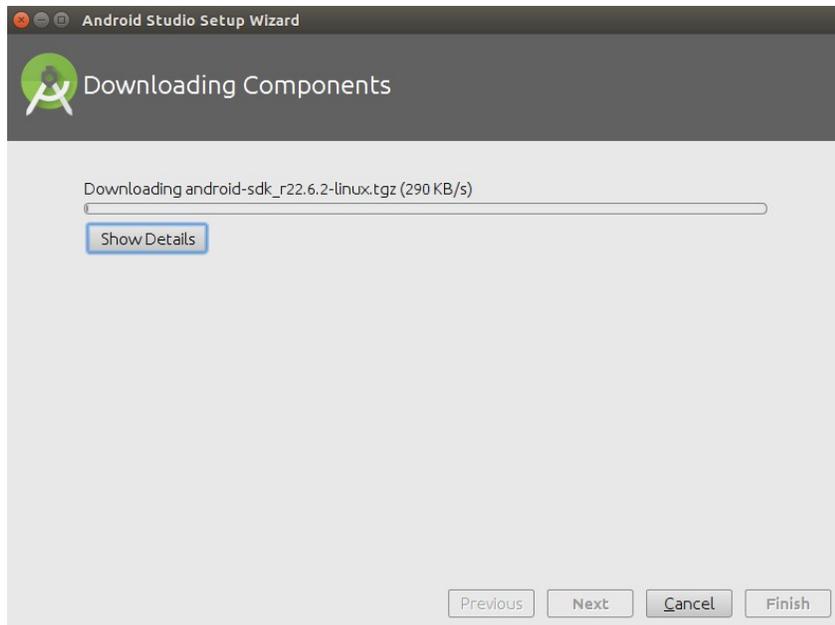
Android studio installation will be complete after this step. When you relaunch Android studio, you will be shown the following welcome screen from where you will be able to start working with your Android Studio.

Chapter-2

Installing Android Studio on Windows

Android Studio is one of the best and most popular IDE for development of Android application which is released by IntelliJ IDEA. Android studio has rich functionalists and easy to work. Android studio has great and awesome user interface, it is also smart with coding and will save your time.

Android programming is Java based programming so it requires JDK & JRE environment. You must have toinstall JDK & JREfor working with android. After installing JDK and JRE you need to configure Environment variables on your computer. Environment variables let to provide some specific path for android. There will be JDK/bin or JDK only. After installing JDK & JRE and configuring environment variables, the next step will be installation of SDK (Software Development Kit). This is another important component for android programming. If you are thinking to develop android apps, then you must install SDK on your computer.

Installing Andrio Studio on Windows

To download Android Studio for you computer, you need to visit official site of android. There you will see android studio download button.
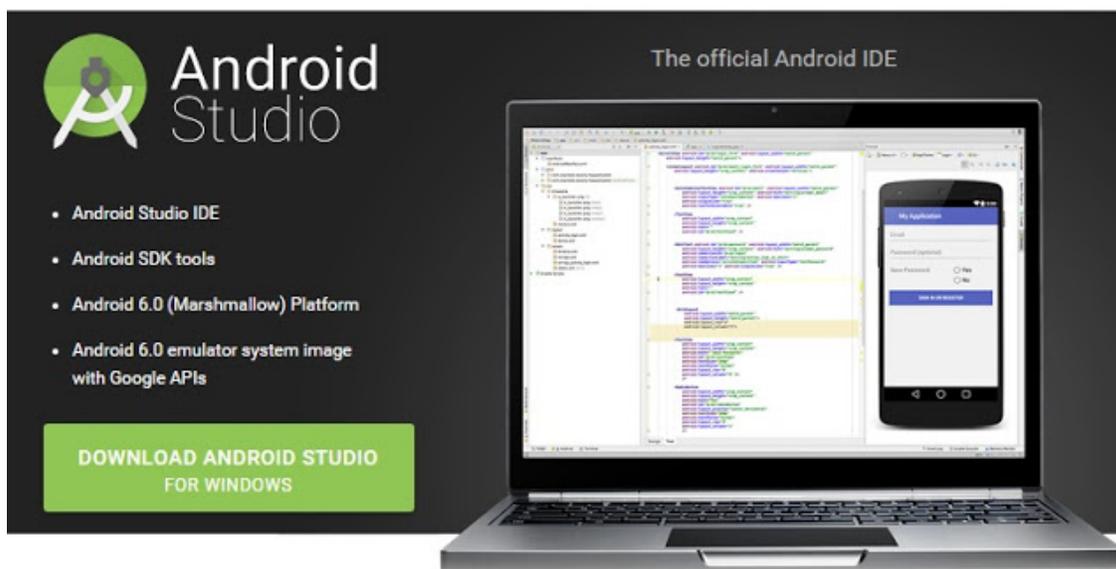
After downloading android studio, follow below steps to install android studio on your computer.

Step1: Open you downloaded android studio file and wait for few seconds, it will take some time to load on your screen.

Step2: After that the window will appear on your screen, where it will show you welcome to android studio.

Go for **"Next"**

Step3: Now choose what you want to install, all are most important component in android programming.

Go for "Next"
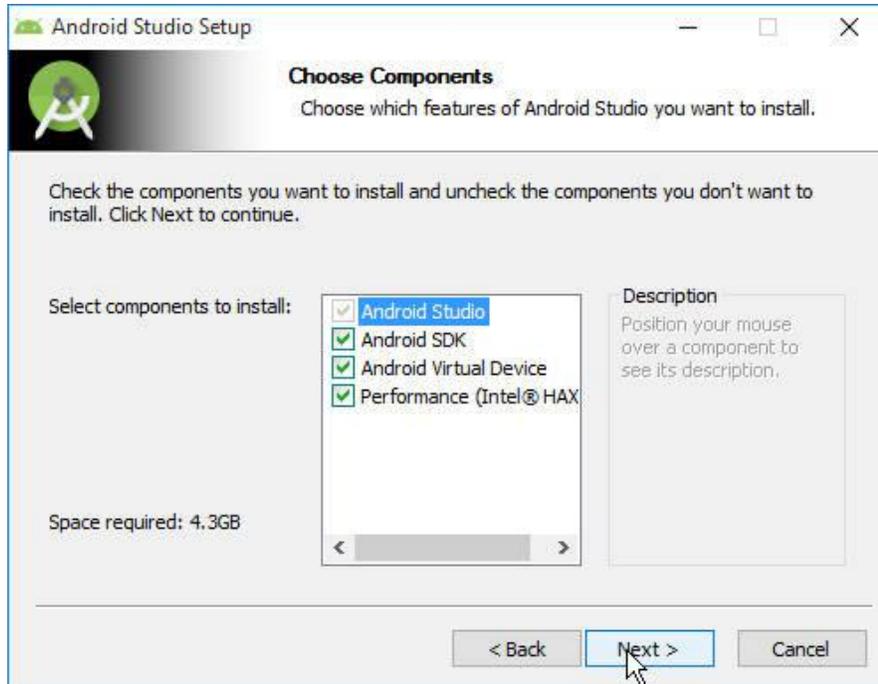
Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation.
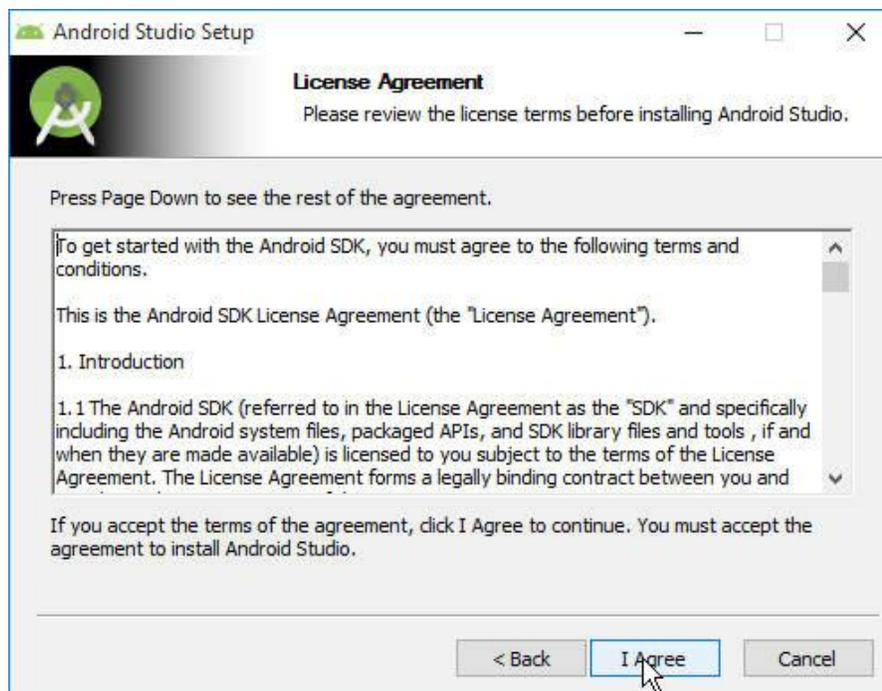
Step5: Go for "I Agree", It's for Haxm installation.
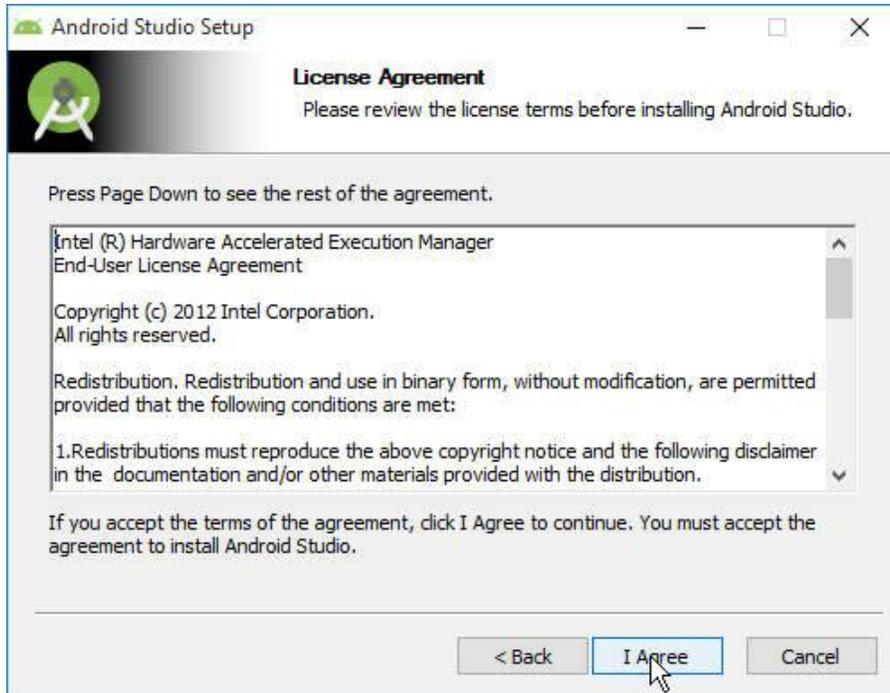
Step6: Now choose the location for Android Studio and SDK.

Go for "Next"
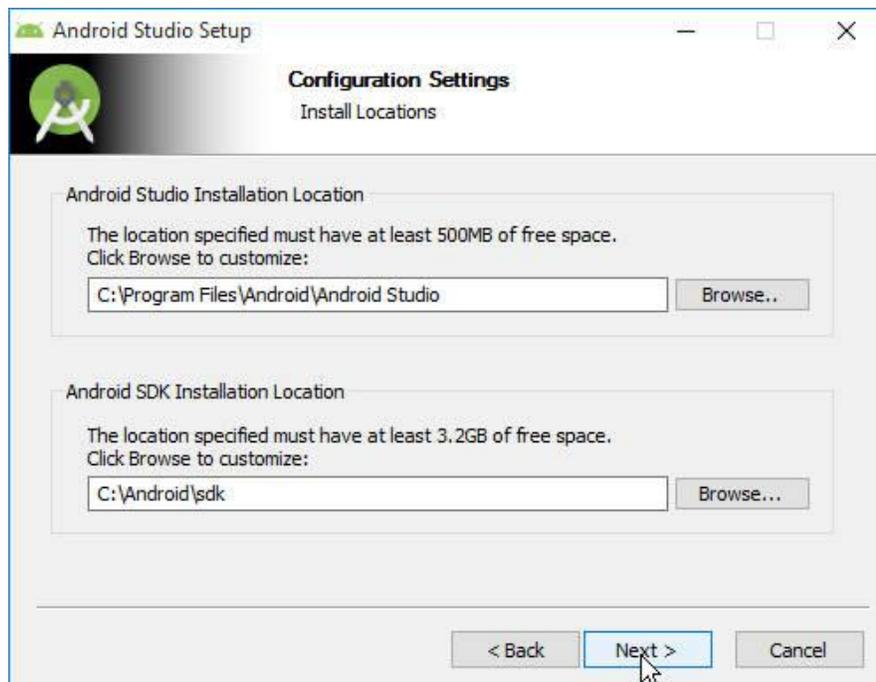
Step7: Here you need to define the size of android emulator processor. You can define up to 2GB and more than that, depends on your RAM capacity. The size will be determined through Haxm. This will help your android studio to run faster.
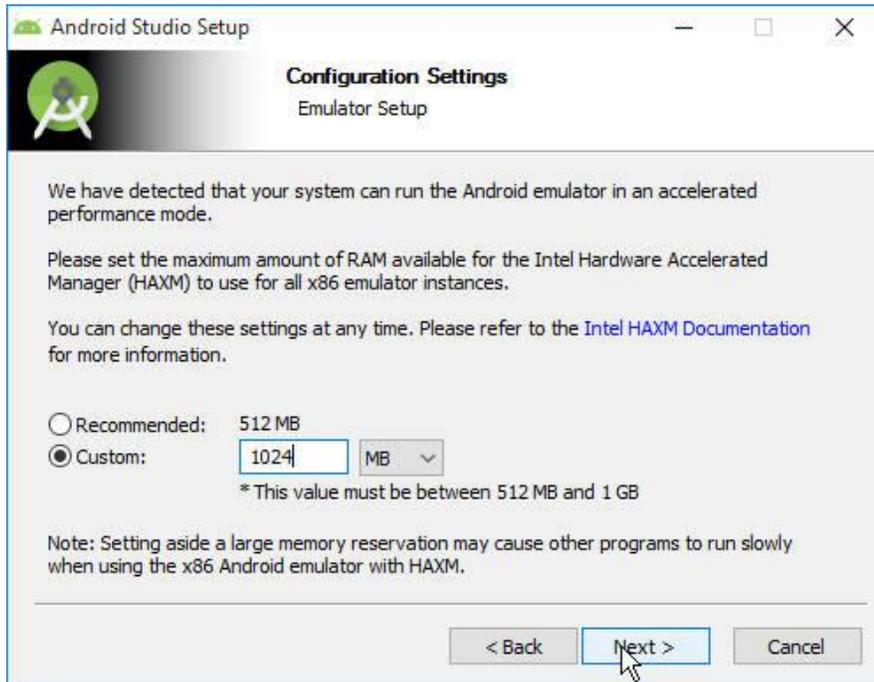
Step8: Now choose setting for shortcut of android studio on your computer.
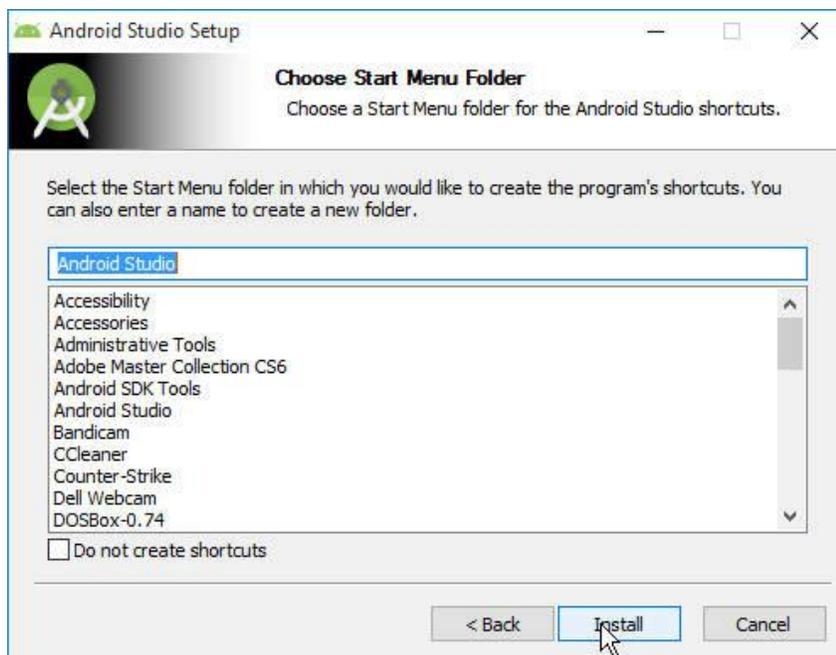
Step9: After that it will take lots of time to copy all the files to the storage location you

have selected before.
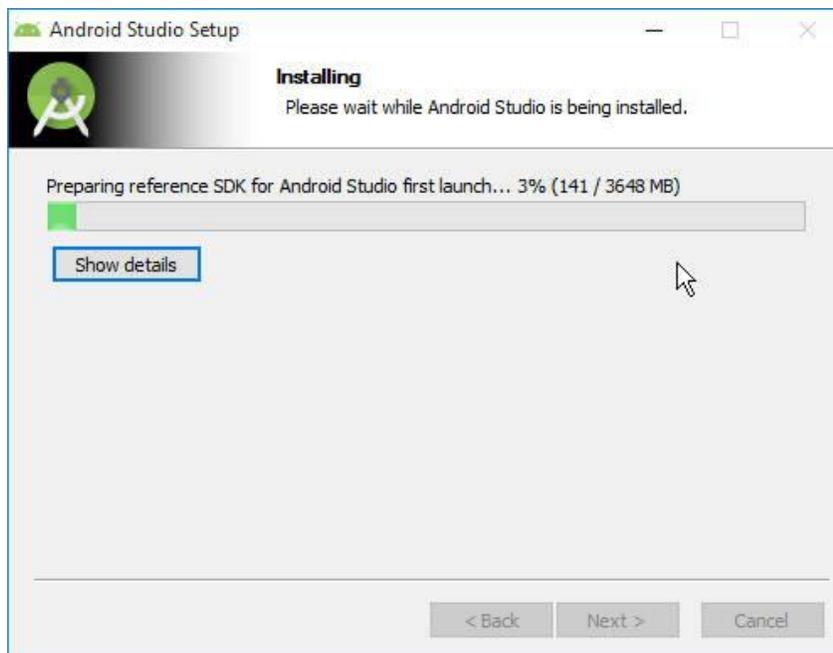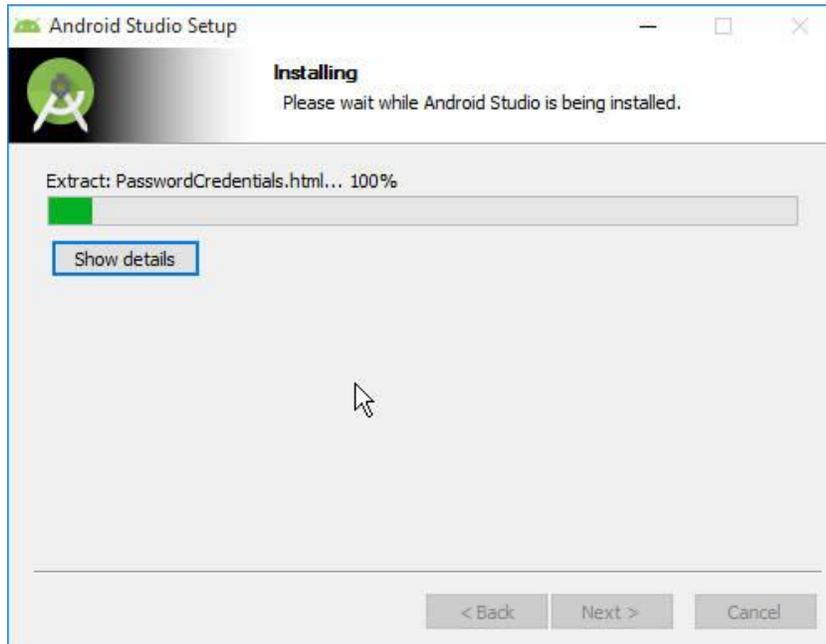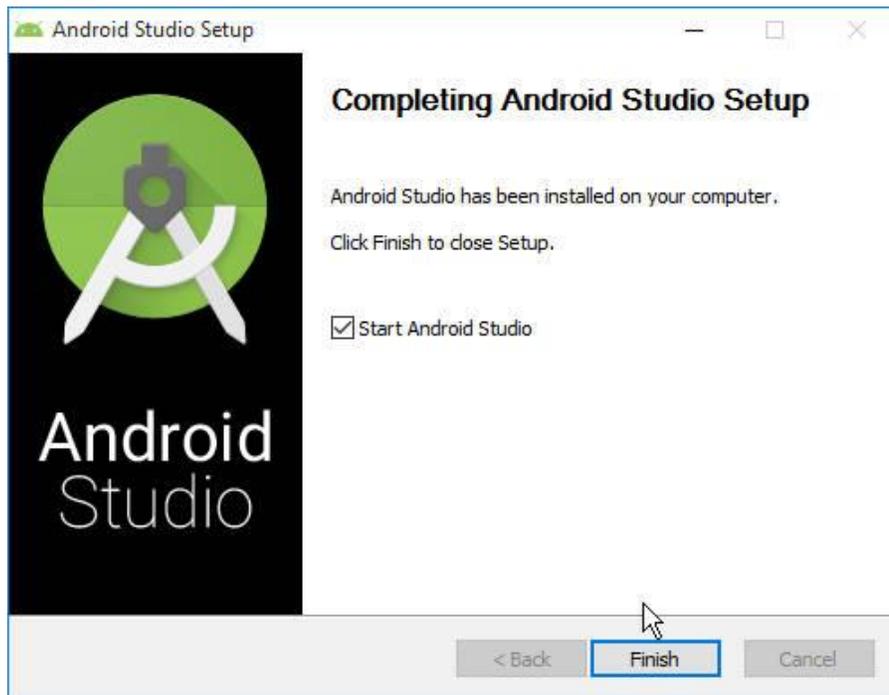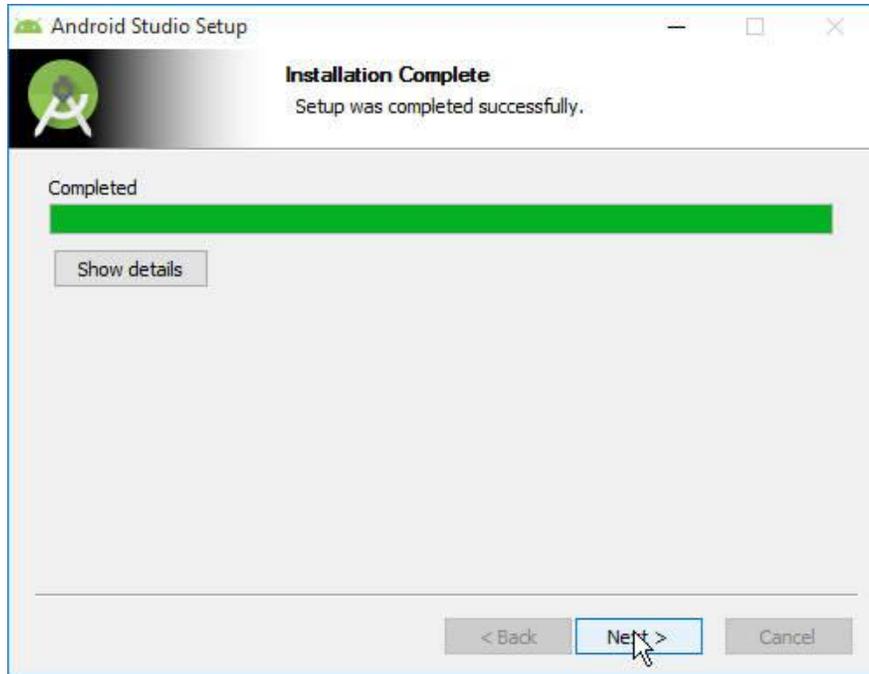
Chapter-3

Creating New Project on Android Studio

Android Studio is one of the best IDE (Integrated Development Environment) for Android Developers with rich functionality and control mechanism. It was announced on 2013 and today it is one of the most popular IDE in the world for android development purpose.

In this chapter we will learn how to create new project on android studio.

Step1: First of all you need to start your **Android Studio.** Then it will show you the default start window on your screen, where you can see various options.
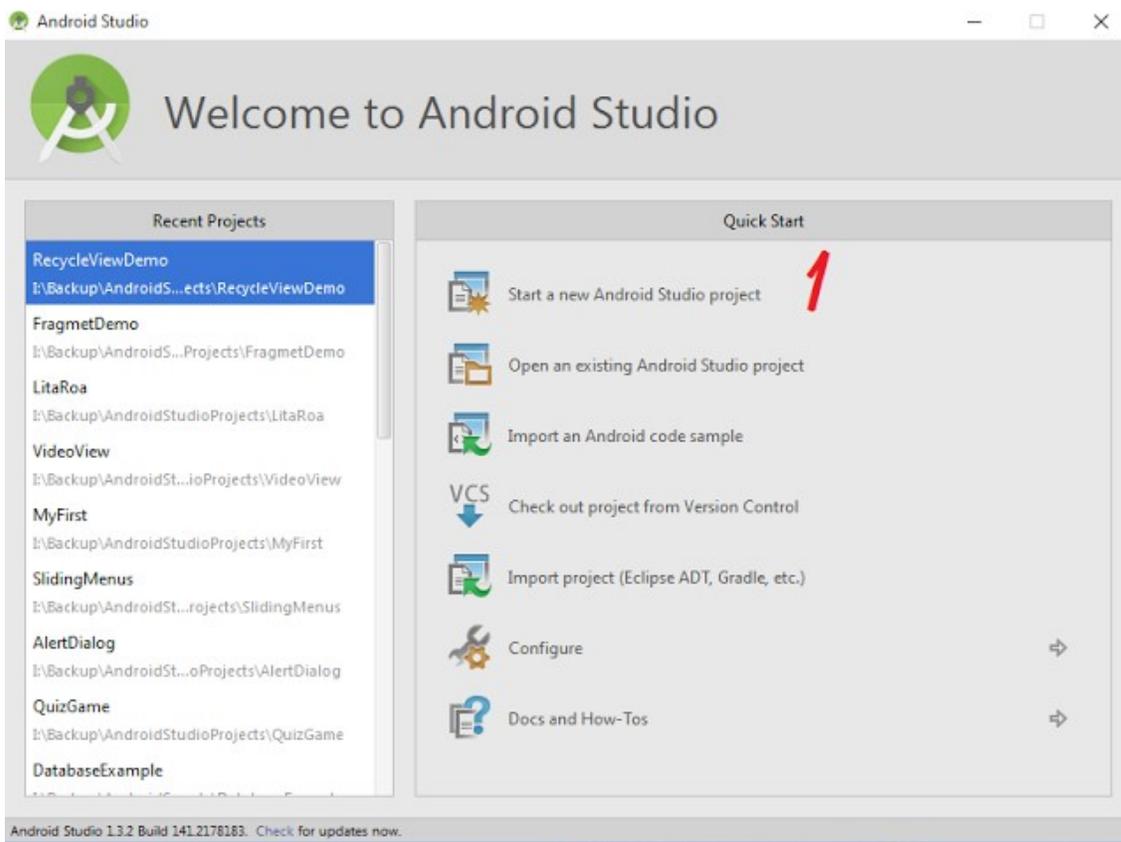
In this window you have to click on **Start a new Android Studio project,** that will let you create new project on android studio.

Step2: After clicking on new project will get another window on your screen which will ask for project information or configuration.
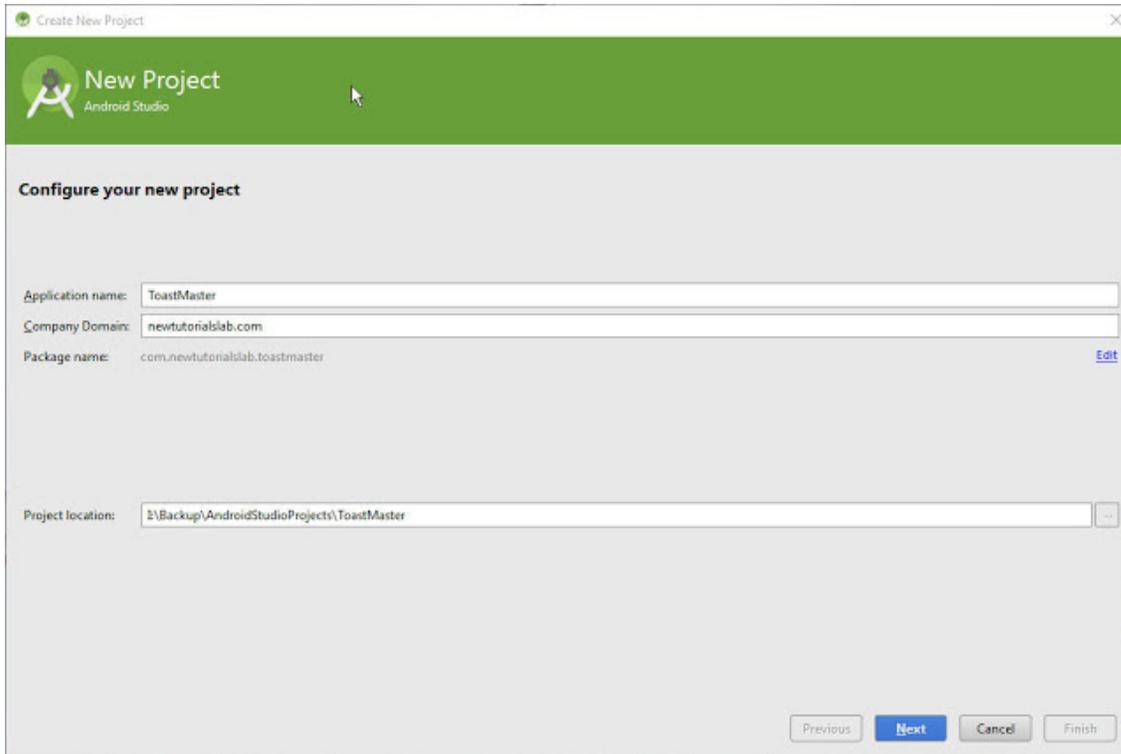
As you can see on the above windows, you need to put **Application Name,**

Application Name: Without Space, Capitalized letter format

1. **ToastMaster - True - Good**

2. **Toastmaster - false - still need modification**

3. **toastmaster - false - not good**

4. **Toast Master - false - poor**

5. **toast maSter - false critical**

Company Domain,

You must write your company name like we have newtutorialslab.com, or ansmachine.net, satyajoshi.com.np etc.

Package Name,

Package naming convention is reverse notation as you write company name it will automatically change the package name in the form of reverse notation.

Project Location,

You must select appropriate location for your project. That might be default location of C drive or you can select another drive location to save your special project.

Step3: After doing all the above tasks you must have to choose platform where you want to run your android application. Run able factors for your project. You can choose as per your need.
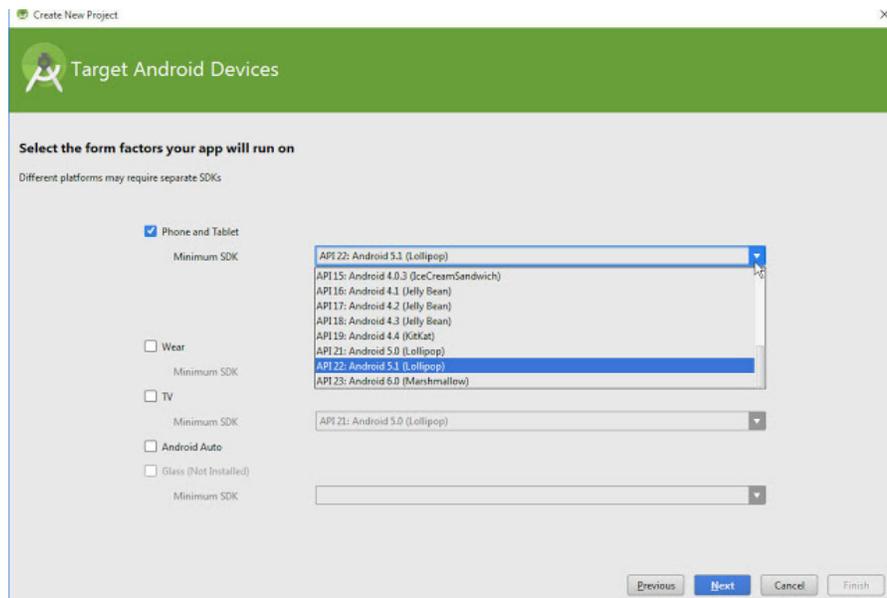
Step4: Now here you have to choose the activity where you will get java classes and xml files for designing and coding purpose. You can even choose no activity format. If you choose that there will be no auto generated codes on your project. If you select **Blank Activity,** you will get auto generated codes and that will save your lots of time.
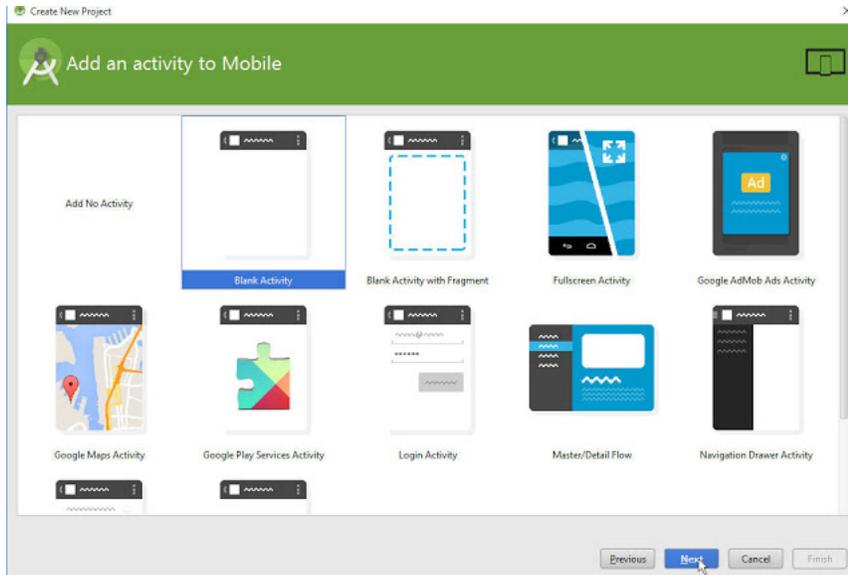
Step5: In this you have to select layout name, activity name etc...
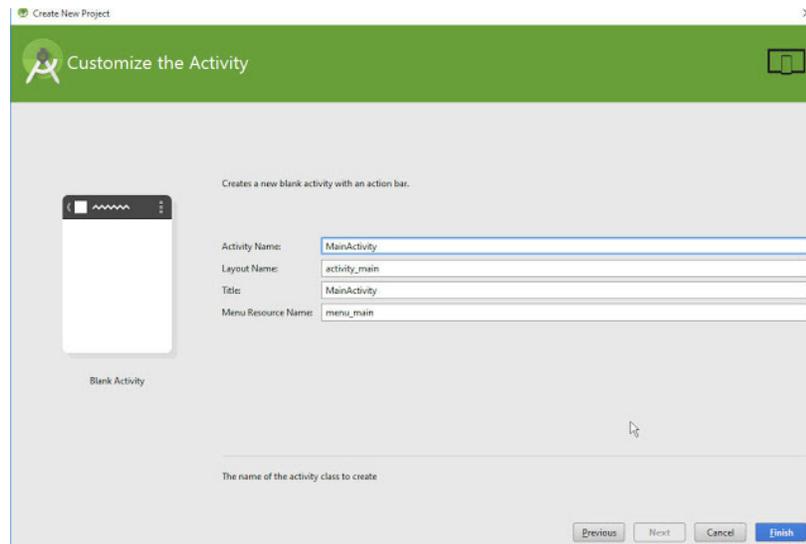
Than after you just have to click on **Finish** button. Now it will take little time to create your new project on Android Studio

Chapter-4

Open, Re-open and Close an Existing Project on Android Studio

Opening an existing project

Steps1: After opening your android studio you will get start screen on your computer screen.

Step2: Now click on **Open an Existing Android Studio project** option where you have to choose the location and open that specific project.
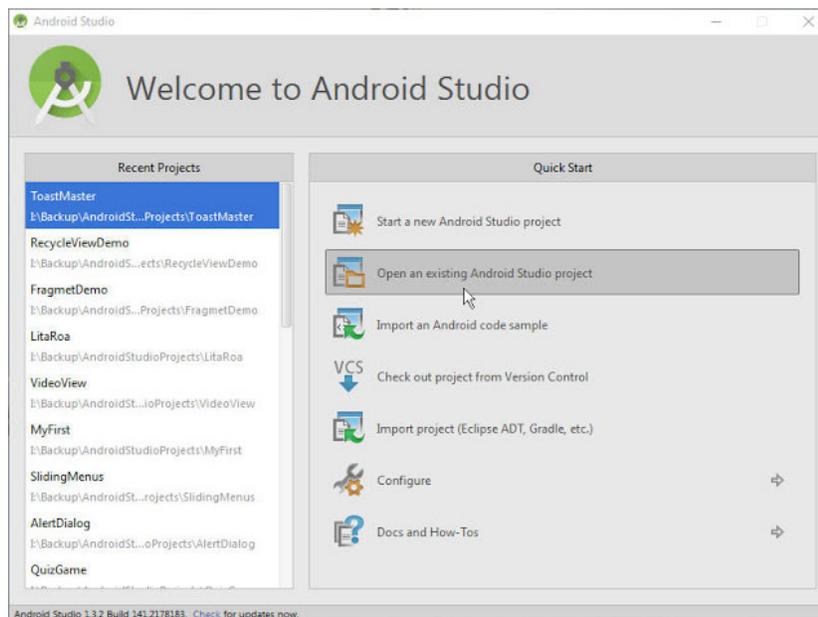
Step3: Now you have to select your project form your specific location. Choose the location and open that project which you want to edit on android studio.
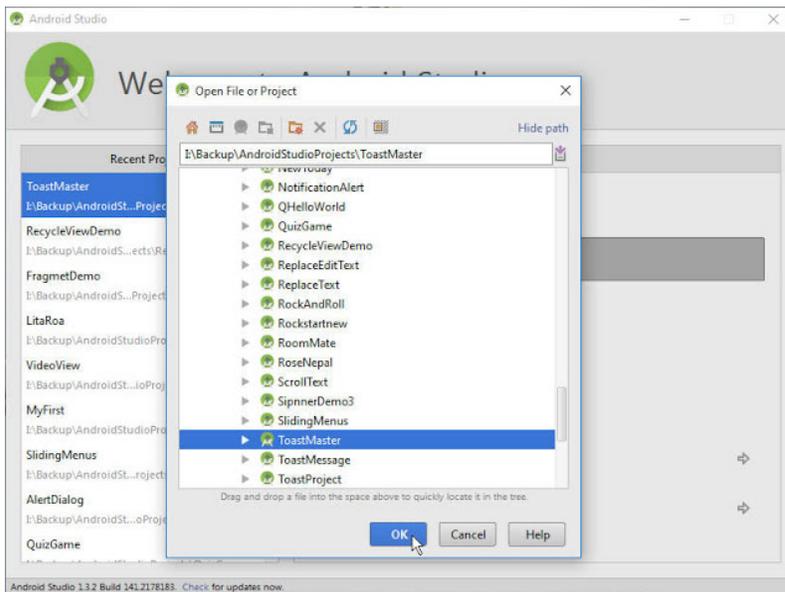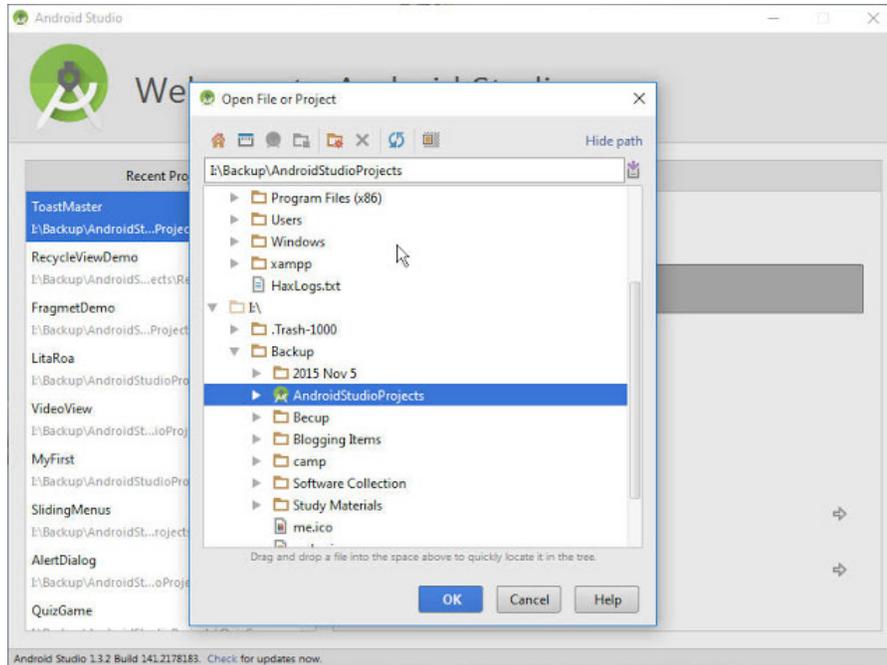
Re-opening project on Android studio

Step1: Start your android studio and go for file menu

Step2: After that you need to click or point on re-open projects option and you have to select project that you want to re-open. After clicking on project you will see option to open project on new window and this window that depends on you how you want to re-open your new project.

ToastMaster - [I:\Backup\AndroidStudioProjects\ToastMaster] - [app] - ...\app\src\main\java\com\newtutorialslab\to

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

New
Import Sample...
Open...
Reopen Project   RecycleViewDemo
Close Project   FragmetDemo
                              LitaRoa
Settings...   Ctrl+Alt+S VideoView
Project Structure... Ctrl+Alt+Shift+S MyFirst
Other Settings   SlidingMenus
                              AlertDialog
Import Settings...   QuizGame
Export Settings...   DatabaseExample
                              GridLayout
Save All   Ctrl+S   MyUsersInfo
Synchronize   Ctrl+Alt+Y MyFragmentDemo
Invalidate Caches / Restart... MyFragment
                              MyForm
Export to HTML...   MyDatabase
Print...   MyCalculator
Add to Favorites   MyCalc
File Encoding   Menu
Line Separators   LoyalFeeder
Make File Read-only   LoadSheding
Power Save Mode   ListViewDemo
                              ListViewControls
Exit

java > com > newtutorialslab > toastmaster > MainAc
nActivity.java ×   activity_main.xml ×

port ...

blic class MainActivity extends Act

    private Button ntlab;

    @Override
    protected void onCreate(Bundle sa
        super.onCreate(savedInstanceS
        setContentView(R.layout.activ

        ntlab = (Button) findViewById

        ntlab.setOnClickListener(new
            @Override
            public void onClick(View
                Toast.makeText(getApp
                    //you can put
                    ntlab.getText
                    ,Toast.LENGTH

Emulator Nexus_4_API_22 Androi   ewtutorialslab.toastmaster (3059)

---

app ▶

com > newtutorialslab > toastmaster > MainActivity

MainActivity.java ×   activity_main.xml ×   strings.xml ×

import ...

public class MainActivity extends Activity {

    private Button ntlab;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

Open Project   ×

?   New projects can either be opened in a new window or replace the project in the existing window.
    How would you like to open the project?

☐ Remember, don't ask again   This Window   New Window   Cancel

                ntlab.getText()//This will show text of your button
                ,Toast.LENGTH_LONG).show();

com.newtutorialslab.toastmaster (3059)

Log level: Verbose   Q▾

.newtutorialslab.toastmaster W/OpenGLRenderer: railed to set EGL_SWAP_BEHAVIOR on surface Ux/3eb/a25ucu, error
.newtutorialslab.toastmaster W/art: Suspending all threads took: 10ms

Closing of project

This closing of project can help you to close your opening project, after closing project it will leads you to the main window of android studio where you will find recent projects and various options.

Step1: Go for file menu and click on close project button that will directly lead you to the another start window of your android studio.
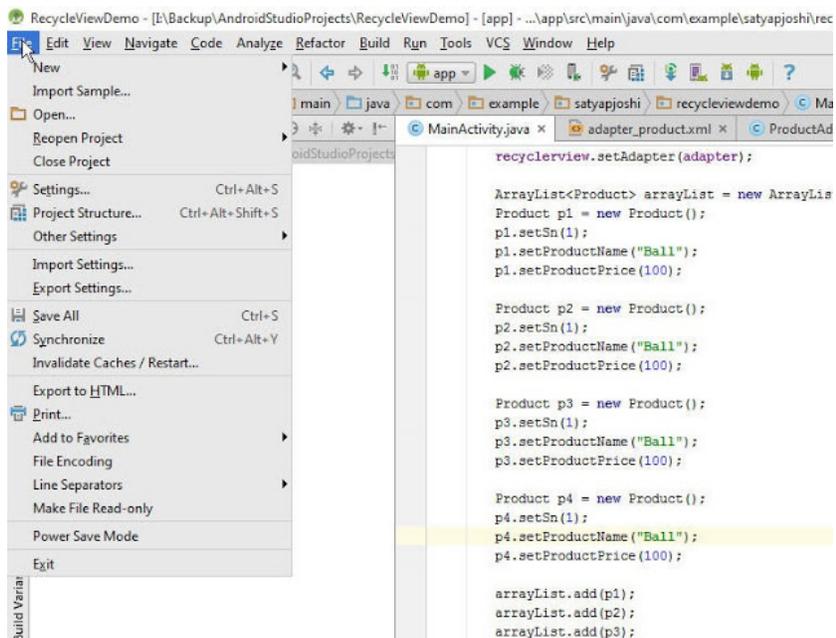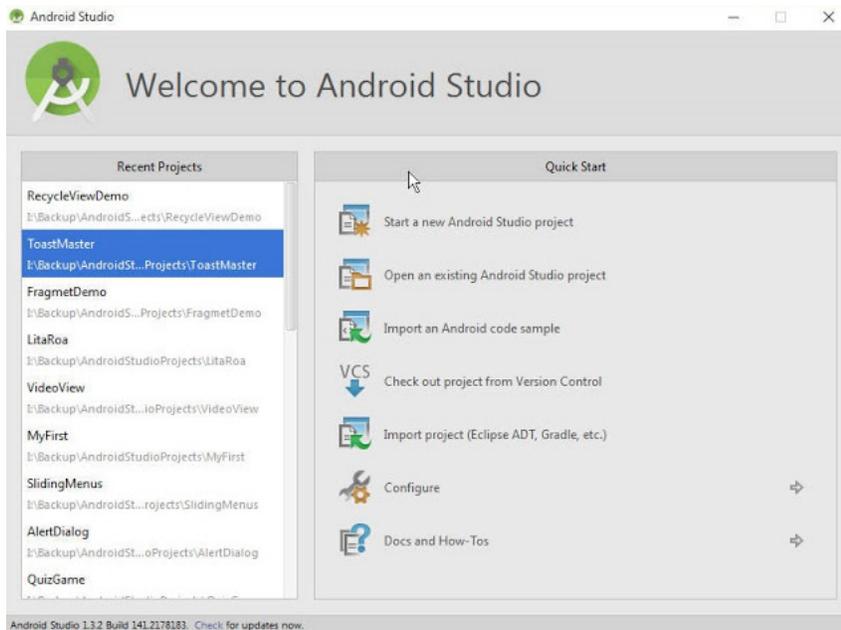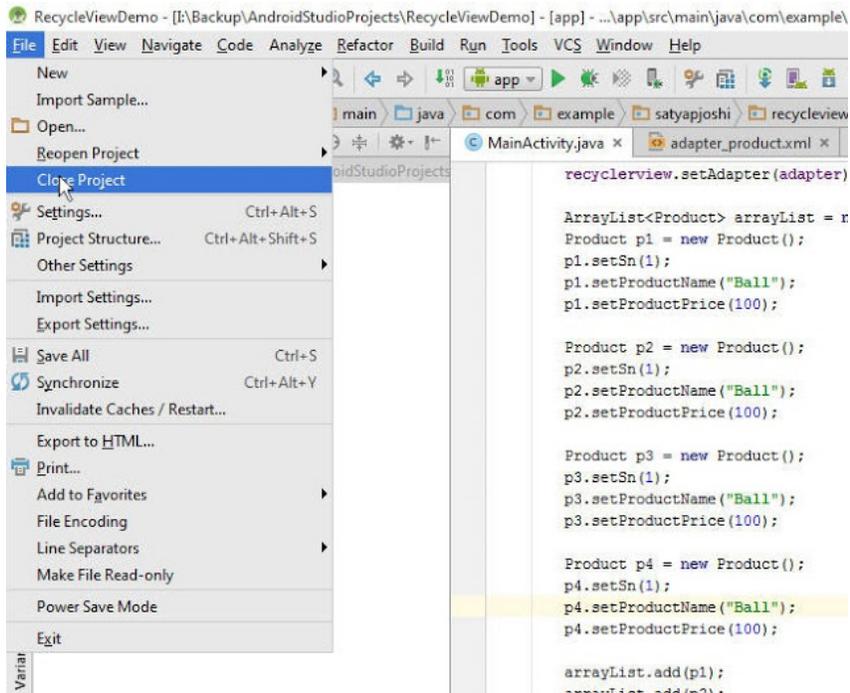
Chapter-4

Creating Android Virtual Devices

The AVD Manager is a tool you can use to create and manage Android virtual devices (AVDs), which define device configurations for the Android Emulator.

To launch the AVD Manager:

- In Android Studio, select **Tools > Android > AVD Manager**, or click the AVD Manager icon in the toolbar.

- Or, use the command line to navigate to your SDK's tools/ directory and execute:

- $ android avd

The AVD Manager main screen shows your current virtual devices, as shown in figure 1.

Figure 1. The AVD Manager main screen shows your current virtual devices.

Note: If you launch the AVD Manager from the command line, the UI is different than how it appears in Android Studio, as documented here. Most of the same functionality is available, but the command-line version of the AVD Manager is currently not documented.

Creating an AVD

You can create as many AVDs as you would like to use with the Android Emulator. To effectively test your app, you should create an AVD that models each device type for which you have designed your app to support. For instance, you should create an AVD for each

API level equal to and higher than the minimum version you've specified in your manifest <uses-sdk> tag.

To create an AVD based on an existing device definition:

1. From the main screen (figure 1), click **Create Virtual Device**.
2. In the Select Hardware window, select a device configuration, such as Nexus 6, then click **Next**.
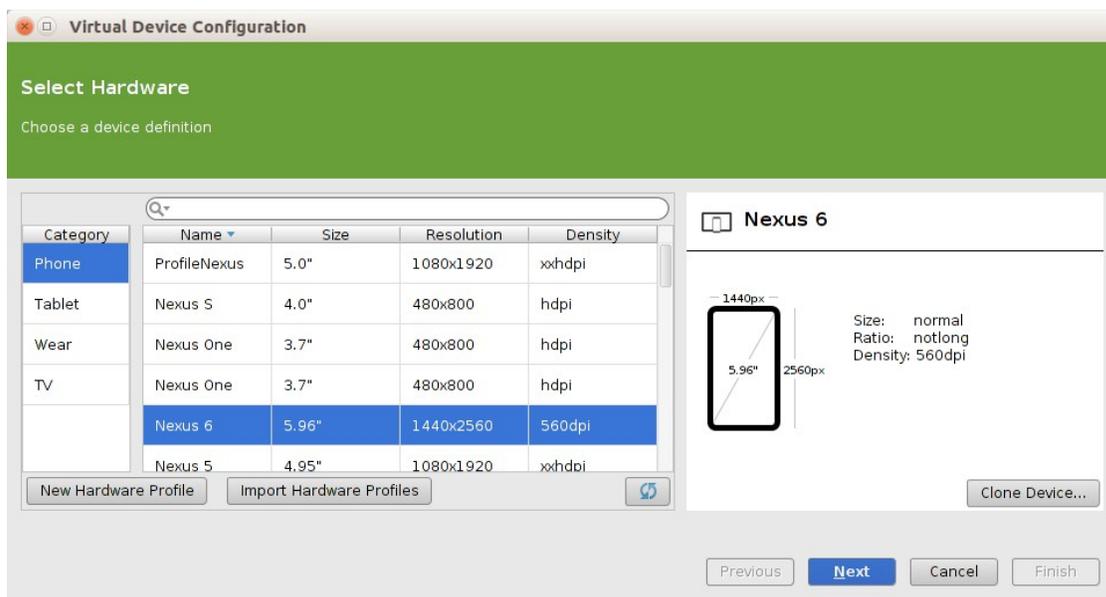
Figure 2. The Select Hardware window.

3. Select the desired system version for the AVD and click **Next**.
4. Verify the configuration settings, then click **Finish**.
5. If necessary, click **Show Advanced Settings** to select a custom skin for the hardware profile and adjust other hardware settings.

To launch the AVD in the Android Emulator, click the launch button in the list of AVDs.

Creating a device definition

In case the available device definitions do not match the device type you'd like to emulate, you can create a custom device definition for your AVD:

1. From the main screen (figure 1), click **Create Virtual Device**.

2. To begin you custom device by using an existing device profile as a template, select a device profile then click **Clone Device**.

3. Or, to start from scratch, click **New Hardware Profile**.

4. The following Configure Hardware Profile window (figure 3) allows you to specify various configurations such as the screen size, memory options, input type, and sensors.

5. When you're done configuring the device, click **Finish**.
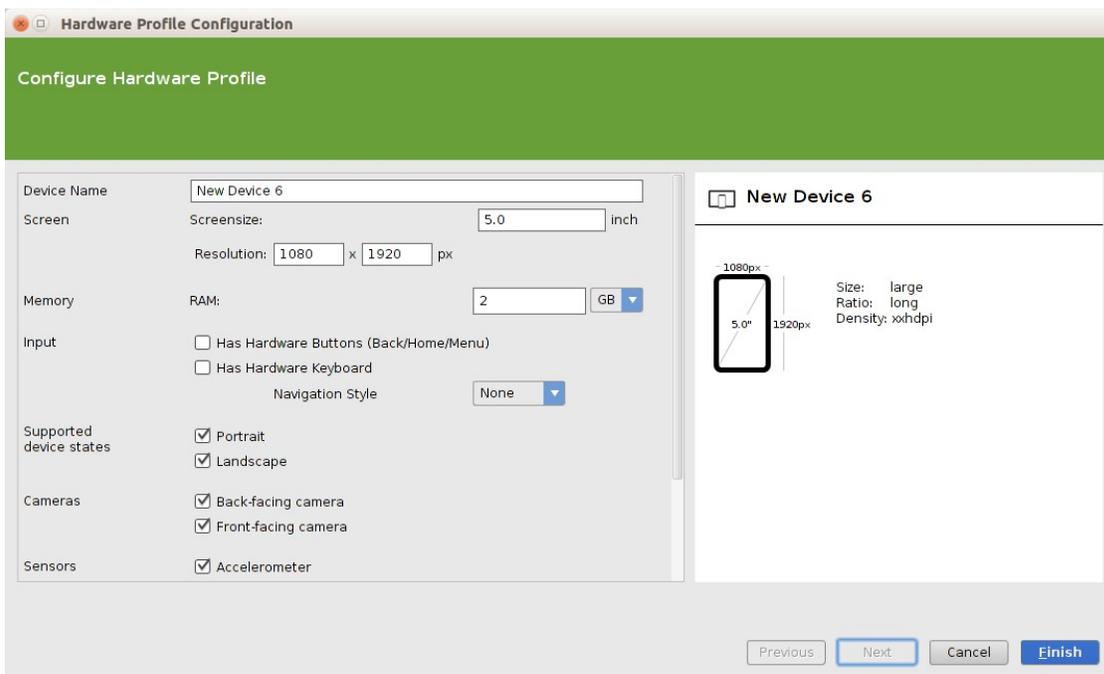
Figure 3. The Configure Hardware window when creating a custom device configuration.

1. Your custom device configuration is now available in the list of device definitions (shown after you click **Create Virtual Device**). To continue preparing an AVD with your custom device configuration, select the new configuration and follow the instructions above to create an AVD with an existing device definition (and select your new definition).

Hardware options

If you are creating a new AVD, you can specify the following hardware options for the AVD to emulate:

Characteristic	Description	Property
Device ram size	The amount of physical RAM on the device, in megabytes. Default value is "96".	hw.ramSize
Touch-screen support	Whether there is a touch screen or not on the device. Default value is "yes".	hw.touchScreen
Trackball support	Whether there is a trackball on the device. Default value is "yes".	hw.trackBall
Keyboard support	Whether the device has a QWERTY keyboard. Default value is "yes".	hw.keyboard
DPad support	Whether the device has DPad keys. Default value is "yes".	hw.dPad
GSM modem support	Whether there is a GSM modem in the device. Default value is "yes".	hw.gsmModem
Camera support	Whether the device has a camera. Default value is "no".	hw.camera
Maximum horizontal camera pixels	Default value is "640".	hw.camera.maxHorizontalPixels
Maximum vertical camera pixels	Default value is "480".	hw.camera.maxVerticalPixels
GPS support	Whether there is a GPS in the device. Default value is "yes".	hw.gps
Battery support	Whether the device can run on a battery. Default value is "yes".	hw.battery

Accelerometer	Whether there is an accelerometer in the device. Default value is "yes".	hw.accelerometer
Audio recording support	Whether the device can record audio. Default value is "yes".	hw.audioInput
Audio playback support	Whether the device can play audio. Default value is "yes".	hw.audioOutput
SD Card support	Whether the device supports insertion/removal of virtual SD Cards. Default value is "yes".	hw.sdCard
Cache partition support	Whether we use a /cache partition on the device. Default value is "yes".	disk.cachePartition
Cache partition size	Default value is "66MB".	disk.cachePartition.size
Abstracted LCD density	Sets the generalized density characteristic used by the AVD's screen. Default value is "160".	hw.lcd.density

Android Applications

1. **Create a Hello World android application**

Aim

To create an "Hello World" application in android studio

Objectives

To familiarise simple android application

Problem Statement

Create a Hello world android application using android studio and compile it using android sdk and run it on Android Virtual Device (AVD).

Apparatus/Tools/Equipments/Components

Android studio

Principle/Theory/Flow-chart/Algorithm

Procedure

Step 1: In Android Studio, create a new project

Step 2: Fill out the fields on the screen, and click **Next**.

Step 3: Select the form factors your app will run on, check the box for Phone and Tablet.

> ➤ For **Minimum SDK**, select **API 8: Android 2.2 (Froyo)**.
>
> ➤ The Minimum Required SDK is the earliest version of Android that your app
>
> supports, indicated using the API level. To support as many devices as possible, you

should set this to the lowest version available that allows your app to provide its core feature set.

➢ Leave all of the other options (TV, Wear, and Glass) unchecked and click **Next**
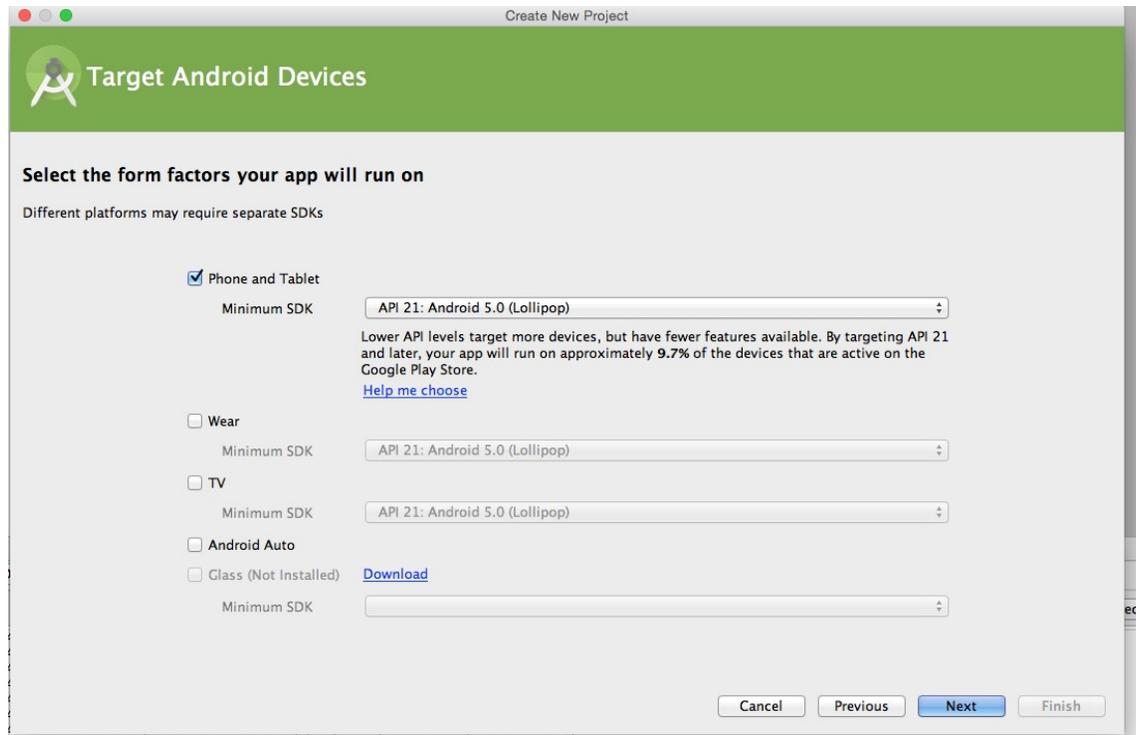
Step 4: Under **Add an activity to *<template>***, select **Blank Activity** and click **Next**.

➢ An activity is one of the distinguishing features of the Android framework. Activities provide the user with access to your app, and there may be many activities. An application will usually have a main activity for when the user launches the application, another activity for when she selects some content to view
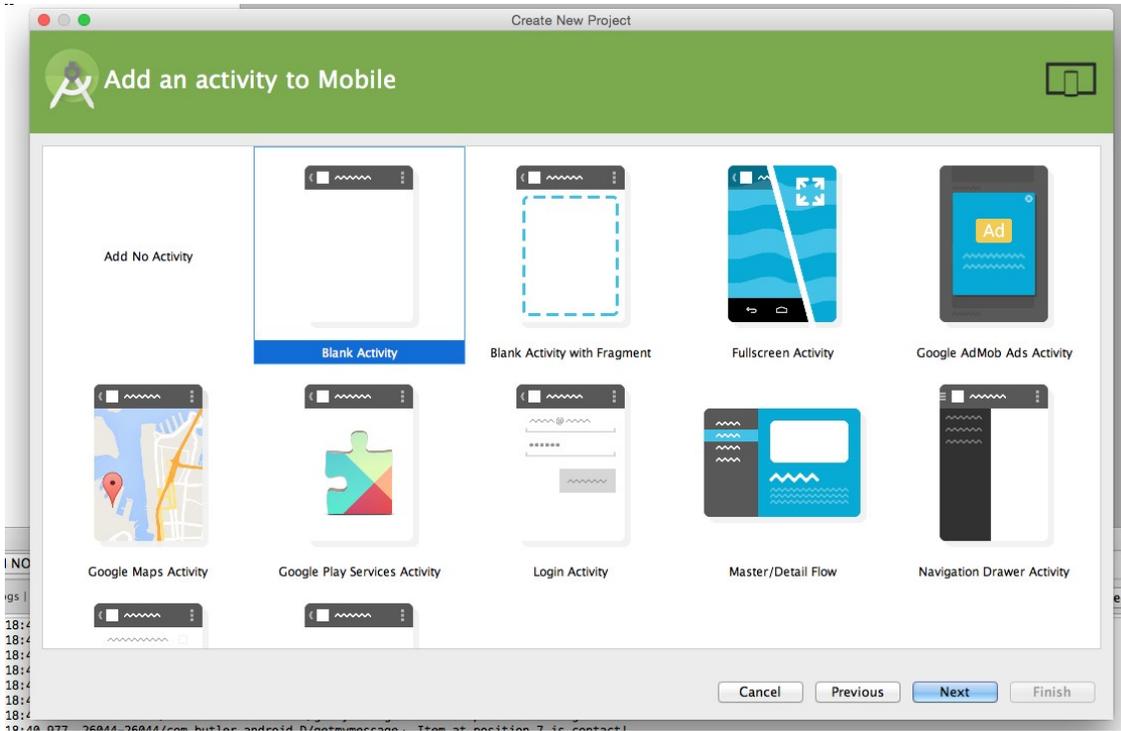
1. Under **Customize the Activity**, change the **Activity Name** to MainActivity. The **Layout Name** changes to *activity_main*, and the **Title** to *MainActivity*. The **Menu Resource Name** is *menu_main*.

2. Click the **Finish** button to create the project.
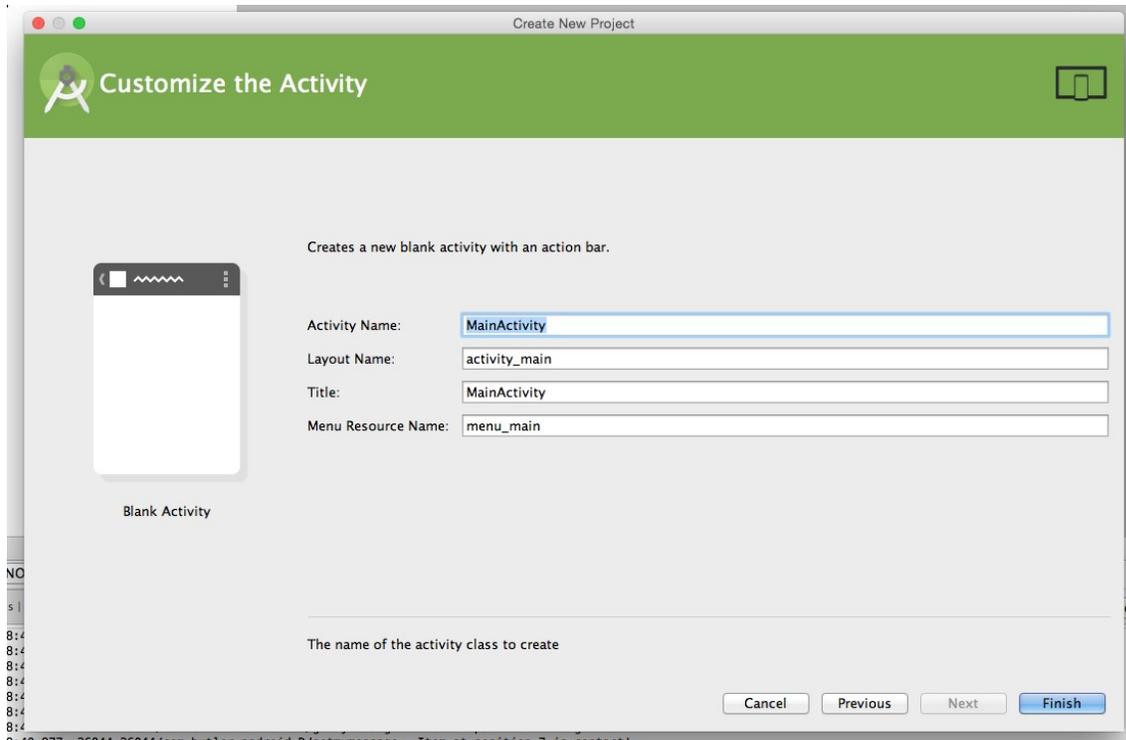
Android project is now a basic "Hello World" app that contains the following default files.

app/src/main/res/layout/activity_main.xml

> This XML layout file is for the activity you added when you created the project with Android Studio. Following the New Project workflow, Android Studio presents this file with both a text view and a preview of the screen UI.

app/src/main/java/com.mycompany.myfirstapp/MainActivity.java

> A tab for this file appears in Android Studio when the New Project workflow finishes. When you select the file you see the class definition for the activity you created. When you build and run the app, the Activity class starts the activity and loads the layout file that says "Hello World!"

app/src/main/AndroidManifest.xml

> The manifest file describes the fundamental characteristics of the app and defines
> each of its components. You'll revisit this file as you follow these lessons and add
> more components to your app.

app/build.gradle

> Android Studio uses Gradle to compile and build your app. There is a build.gradle file
> for each module of your project, as well as a build.gradle file for the entire project.
> Usually, you're only interested in thebuild.gradle file for the module, in this case the
> app or application module

Note also the /res subdirectories that contain the resources for your application:

drawable-<*density*>/

> Directories for drawable resources, other than launcher icons, designed for various
> densities.

layout/

> Directory for files that define your app's user interface like activity_my.xml, which
> describes a basic layout for the MyActivity class.

menu/

> Directory for files that define your app's menu items.

mipmap/

> Launcher icons reside in the mipmap/ folder rather than the drawable/ folders. This
> folder contains the ic_launcher.png image that appears when you run the default app.

values/

> Directory for other XML files that contain a collection of resources, such as string
> and color definitions.

MainActivity.java - HelloWorld - [~/Documents/vishnu/testing/HelloWorld]

HelloWorld

```java
package test.sample.com.helloworld;

import ...

public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Gradle build finished in 9s 269ms (a minute ago)

activity_main.xml - HelloWorld - [~/Documents/vishnu/testing/HelloWorld]

HelloWorld

Palette

Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout

Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal)
SeekBar
RatingBar
Spinner
WebView

Text Fields
Plain Text
Person Name
Password
Password (Numeric)
E-mail
Phone
Postal Address
Multiline Text
Time
Date

HelloWorld

Hello world!

Component Tree
Device Screen
RelativeLayout
TextView - @string/hello_world

Properties

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
accessibilityTraversalAft	
accessibilityTraversalBel	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	
contentDescription	
contextClickable	
elevation	
focusable	
focusableInTouchMode	
foreground	
foregroundGravity	[]
foregroundTint	

Source Code

MainActivity.java

```java
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
  }

  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
     // Inflate the menu; this adds items to the action bar if it is present.
     getMenuInflater().inflate(R.menu.menu_main, menu);
     return true;
  }

  @Override
  public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
     if (id == R.id.action_settings) {
        return true;
     }

     return super.onOptionsItemSelected(item);
  }
}
```

activity_main.xml

```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"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  tools:context=".MainActivity">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

android:text="@string/hello_world" />

</RelativeLayout>

Observations
Input

Steps in running the project
1. Start the android virtual device(Emulator)

➢ In Android Studio, select Tools > Android > AVD Manager, or click

the AVD Manager icon [image] in the toolbar.

➢ The AVD Manager main screen shows your current virtual devices, as
shown in figure 1.

Figure 1. The AVD Manager main screen shows your current virtual devices.

➢ To launch the AVD in the Android Emulator, click the launch button [image]
in the list of AVDs.

➢ RUN the android application

OR

2. Run the app from Android Studio

1. Select one of your project's files and click **Run** [image] from the toolbar.
2. In the **Choose Device** window that appears, select the **Choose a running device**
radio button, select your device, and click **OK** .

Android Studio installs the app on your connected device and starts it.

<u>Output</u>

2. Create an android application to understand the Life Cycle of an Activity

Aim:

To create an android application to understand the life cycle of an activity.

Objectives:

Create an android application and understand the application life cycle in different scenarios.

Apparatus/Tools/Equipments/Components

Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure

How to open logcat for view the program output

Displaying Android Monitor

Android Monitor is integrated into the Android Studio main window:

- To display Android Monitor, click ![6: Android Monitor], which by default is at the bottom of the main window.
- To hide Android Monitor, click ![6: Android Monitor] again.
- Click the "Logcat" tab to view the log output.
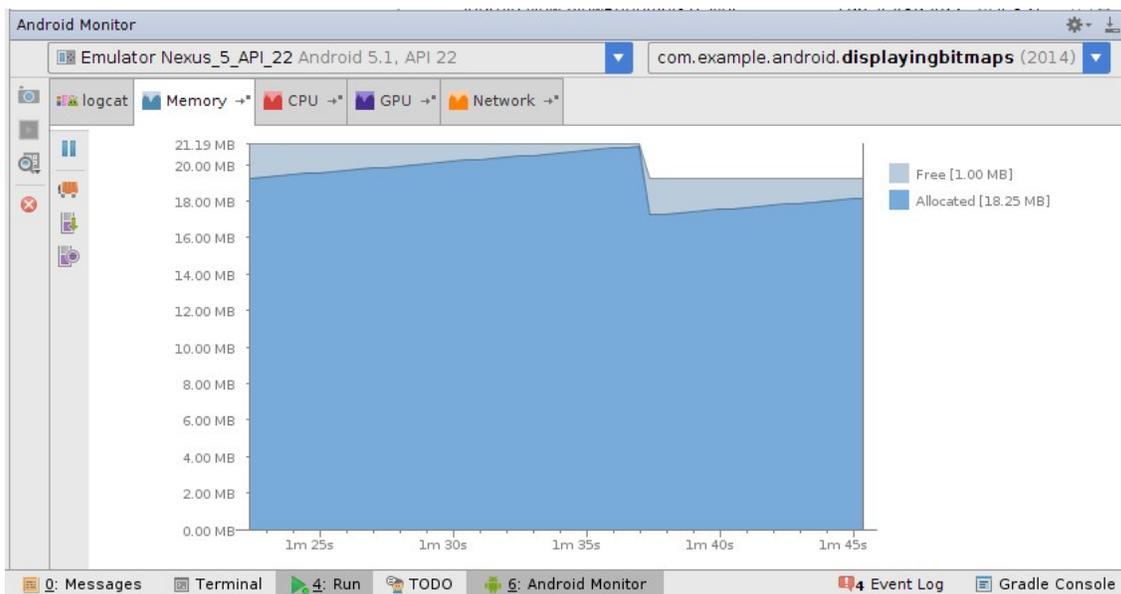
Source Code

MainActivity.java

```
package test.sample.com.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends Activity {
```

```java
    String tag = "Activity Life Cycle";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
      Log.d(tag, "In the onCreate() event");

    }
    public void onStart()
    {
        super.onStart();
        Log.d(tag, "In the onStart() event");
    }
    public void onRestart()
    {
        super.onRestart();
        Log.d(tag, "In the onRestart() event");
    }
    public void onResume()
    {
        super.onResume();
        Log.d(tag, "In the onResume() event");
    }
    public void onPause()
    {
        super.onPause();
        Log.d(tag, "In the onPause() event");
    }
    public void onStop()
    {
        super.onStop();
        Log.d(tag, "In the onStop() event");
    }
    public void onDestroy()
    {
        super.onDestroy();
        Log.d(tag, "In the onDestroy()event");
    }


}
```

Result

Logcat output

Scenario

1.Opening App

D/Activity Life Cycle: In the onCreate() event

D/Activity Life Cycle: In the onStart() event

D/Activity Life Cycle: In the onResume() event

2.Minimize the application

D/Activity Life Cycle: In the onPause() event

D/Activity Life Cycle: In the onStop() event

3. Open Application from background

D/Activity Life Cycle: In the onRestart() event

D/Activity Life Cycle: In the onStart() event

D/Activity Life Cycle: In the onResume() event

4. close application

D/Activity Life Cycle: In the onPause() event

D/Activity Life Cycle: In the onStop() event

D/Activity Life Cycle: In the onDestroy()event

5. Remove Application from minimized list

D/Activity Life Cycle: In the onDestroy()event

3. Linking Activities using Intent

Aim :
To create a android application to understand the use of linking activities using Intent.

Objective :
Create an android application to understand the use of Intent.

Problem statement
Create an android application with two activities. First activity with a "ENTER" button and second screen with a welcome message. By clicking "ENTER" it should bring the use to the second screen.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure

Source Code

MainActivity.java

```java
public class MainActivity extends Activity {
  Button mButton;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mButton = (Button) findViewById(R.id.button);
    mButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

        Intent _Intent = new Intent(MainActivity.this, SecondActivity.class);
        startActivity(_Intent);
      }
    });

  }
}
```

activity_main.xml

```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/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="ENTER" />
</RelativeLayout>
```

SecondActivity.java

```java
public class SecondActivity extends Activity {

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

activity_second.xml

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

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Welcome "
    android:id="@+id/textView"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```
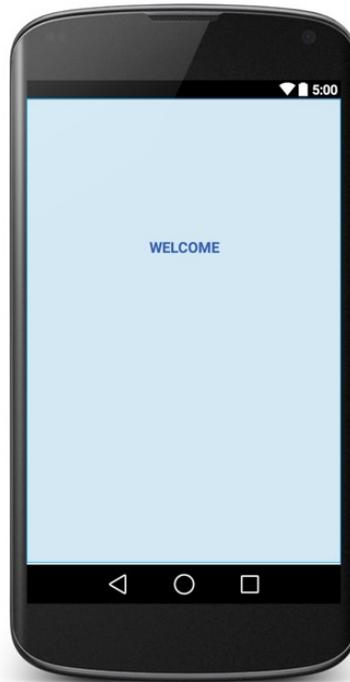
Observations

Input-Output

Screenshots

4. Passing Data Using an Intent Object

Aim :
To create an android application to understand passing data through Intent.

Objective :
Create an android application for authentication process.

Problem Statement
Create an android application for authentication process. First activity should accept the user email address and pass the email id to the second activity. In the second activity it should show the user email id and ask for the password.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure

Source Code

MainActivity.java

```
public class MainActivity extends Activity {

  Button mButton;
  EditText mEditText;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mButton = (Button) findViewById(R.id.button);
    mEditText = (EditText) findViewById(R.id.editText);
    mButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

        String _username = mEditText.getText().toString().trim();
        Bundle bundle = new Bundle();
        bundle.putString("USERNAME", _username);
        Intent _Intent = new Intent(MainActivity.this, SecondActivity.class);
        _Intent.putExtras(bundle);
        startActivity(_Intent);
```

```
        }
    });

    }
}
```

activity_main.xml

```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:padding="20dp">

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_marginTop="150dp"
    android:hint="Username"
    android:inputType="textEmailAddress"/>
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="NEXT"
    android:textColor="#ffffff"
    android:background="@color/colorPrimaryDark"
    android:layout_below="@+id/editText"/>

</RelativeLayout>
```

SecondActivity.java

```java
public class SecondActivity extends Activity {

    TextView mTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        mTextView = (TextView) findViewById(R.id.textView);

        Bundle _bundle = getIntent().getExtras();
```

```java
        String _username = _bundle.getString("USERNAME");
        mTextView.setText("Hi "+_username+",");

    }
}
```

activity_second.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<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"
    android:padding="20dp">

<TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="150dp"
        android:gravity="center"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="@color/colorPrimaryDark"
        android:textStyle="bold" />

<EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView"
        android:layout_marginTop="20dp"
        android:hint="Password"
        android:inputType="textPassword" />

<Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText"
        android:layout_marginTop="10dp"
        android:background="@color/colorPrimaryDark"
        android:text="SUBMIT"
        android:textColor="#ffffff" />
</RelativeLayout>
```
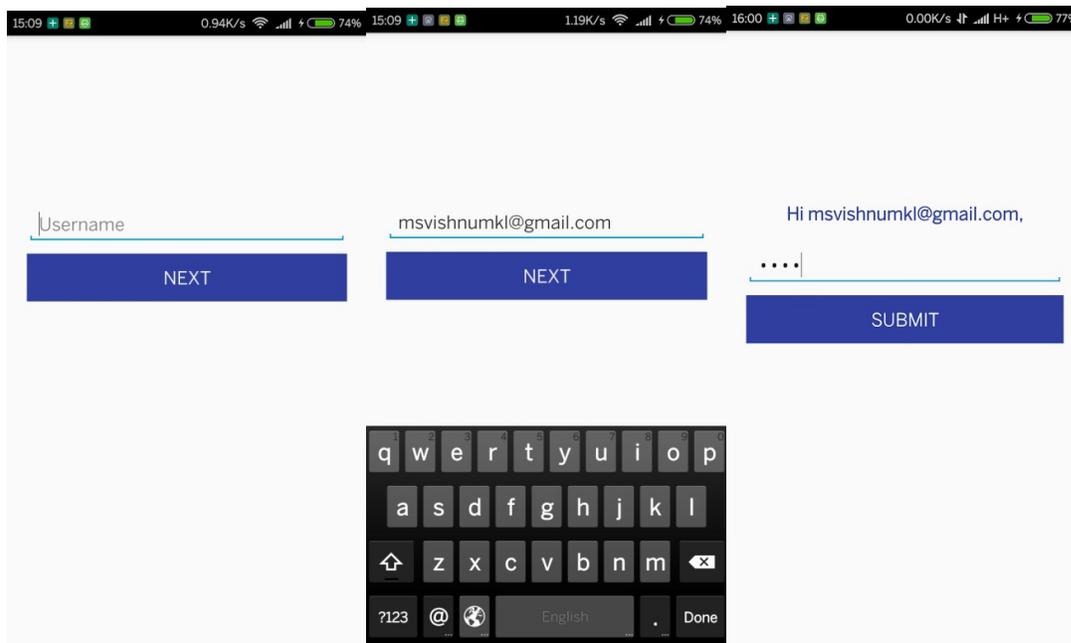
Input-Output

screenshots

5. Implicit intents Familiarization

Aim:

To create an android application to invoke built in application like contact,browser,google map and dialer.

Objectives:
Create an android application to familiarize implicit intent

Problem Statement
Create an android application with four buttons like web browser,make calls,show map and choose contact

Operations are
1. By clicking on we browser button , it should open "www.google.com" in android built in browser application
2. By Clicking on make call button, it should open the phone dialer with a predefined number.
3. By Clicking on show map button, it should plot a specified location co ordinates in the built in google map application.
4. By clicking on choose contact button, it should allow the user to select a particular contact and show the details as a toast message.

Apparatus/Tools/Equipments/Components
Android studio

Principle/Theory/Flow-chart/Algorithm

Procedure

Source Code
MainActivity.java

```java
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {

  Button mBtnBrowser, mBtnCall, mBtnMaps, mBtnContact;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

```java
        setContentView(R.layout.activity_main);
        mBtnBrowser = (Button) findViewById(R.id.btn_web_browser);
        mBtnCall = (Button) findViewById(R.id.btn_make_calls);
        mBtnMaps = (Button) findViewById(R.id.btn_show_map);
        mBtnContact = (Button) findViewById(R.id.btn_choose_contact);

        mBtnBrowser.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse("http://www.google.com"));
                startActivity(i);
            }
        });
        mBtnCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:+651234567"));
                startActivity(i);
            }
        });
        mBtnMaps.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse("geo:37.827500,-122.481670"));
                startActivity(i);
            }
        });
        mBtnContact.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(android.content.Intent.ACTION_PICK);
                i.setType(ContactsContract.Contacts.CONTENT_TYPE);
                startActivityForObservations(i, 1);
            }
        });
    }
    public void onActivityObservations(int requestCode, int ObservationsCode, Intent data) {
        if (requestCode == 1) {
            if (ObservationsCode == OBSERVATIONS_OK) {
                Toast.makeText(this, data.getData().toString(), Toast.LENGTH_SHORT).show();
                Intent i = new Intent(android.content.Intent.ACTION_VIEW,
Uri.parse(data.getData().toString()));
                startActivity(i);
            }
        }
```

```
    }
}
```

activity_main.xml

```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:padding="20dp"
    android:orientation="vertical">
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Web Browser"
        android:id="@+id/btn_web_browser" />
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Make Calls"
        android:id="@+id/btn_make_calls" />
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Show Map"
        android:id="@+id/btn_show_map" />
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Choose Contact"
        android:id="@+id/btn_choose_contact" />
</LinearLayout>
```

Observations

Input-Output
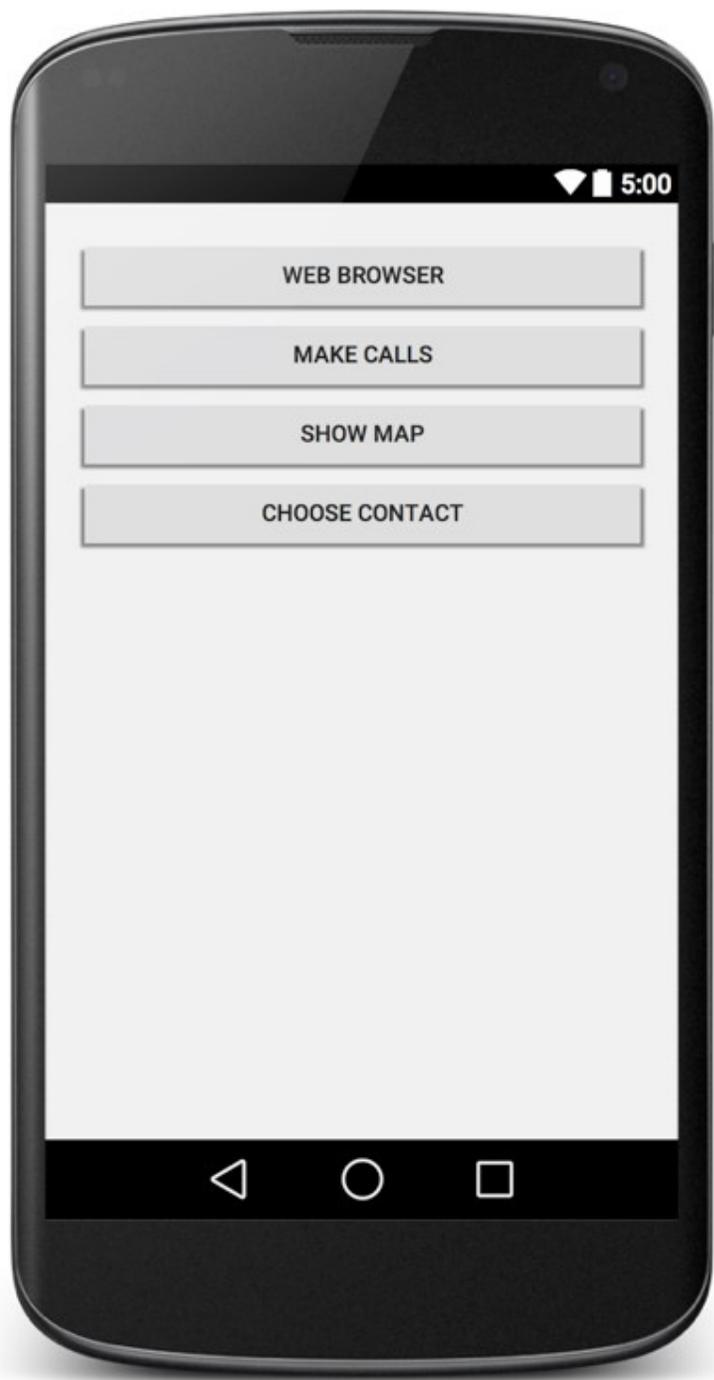
Screenshots

6. Understanding different layouts in android

Aim :
To create an android application with multiple screen. Each screen should use different layout like LinearLayout, RelativeLayout, FrameLayout and TableLayout

Objective
Create an android application to understand the different layouts in android.

Problem Statement
Create an android application with four screen.
First screen should have three buttons such as relativeLayout,TableLayout and Frame Layout. All three should be wrapped under a LinearLayout.

1. By clicking the "RelativeLayout", it should bring the user to the "RelativeLayout" screen.In this screen a sample login page should be created using RelativeLayout.
2. By clicking "Table Layout", it should bring the user to the Table Layout screen. Here also a login page should be designed under a table layout.
3. By Clicking "FrameLayout", it should bring the user to the framelayout screen. There should be an image and a button ("Set as wallpaper") above that image.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

HomeActivity.java

```java
public class HomePage extends Activity {

  Button mBtnRelative, mBtnTable, mBtnFrame;

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

    mBtnRelative = (Button) findViewById(R.id.btn_relative);
    mBtnTable = (Button) findViewById(R.id.btn_table);
    mBtnFrame = (Button) findViewById(R.id.btn_frame);
```

```java
        mBtnRelative.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent _Intent = new Intent(HomePage.this,RelativeLogin.class);
                startActivity(_Intent);

            }
        });

        mBtnTable.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent _Intent = new Intent(HomePage.this,TableLogin.class);
                startActivity(_Intent);

            }
        });

        mBtnFrame.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent _Intent = new Intent(HomePage.this,FrameLayout.class);
                startActivity(_Intent);

            }
        });

    }

}
```

activity_home_page.xml

```xml
<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="com.example.vishnums.myapplication.HomePage">


<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
```

```xml
        android:background="@color/colorPrimary"
        android:gravity="center_vertical"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Layout Example"
        android:textColor="#ffffff"
        android:textSize="20sp"
    />

<Button
        android:id="@+id/btn_relative"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Relative Layout"
        android:layout_marginTop="100dp"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"/>

<Button
        android:id="@+id/btn_table"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Table Layout"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"/>

<Button
        android:id="@+id/btn_frame"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Frame Layout"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"/>
</LinearLayout>
```

RelativeLogin.java

```java
public class RelativeLogin extends Activity {

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

}

activity_relative.xml

```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"
    android:orientation="vertical"
    tools:context="com.example.vishnums.myapplication.HomePage">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Relative Layout"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="80dp"
    android:ems="10"
    android:hint="User Name"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/editText3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/editText2"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="10dp"
```

```
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />

<Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/editText3"
        android:layout_below="@+id/editText3"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="30dp"
        android:background="@color/colorPrimary"
        android:text="Login"
        android:textColor="#ffffff" />
</RelativeLayout>
```

TableLogin.java

```java
public class TableLogin extends Activity {

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

}
```

activity_table.xml

```xml
<TableLayout 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"
    android:padding="20dp">

<TableRow android:layout_marginTop="60dp">
<TextView
        android:layout_width="150dp"
        android:text="User Name" />
<EditText android:layout_width="200dp" />
</TableRow>

<TableRow>
<TextView android:text="Password" />
```

```xml
<EditText android:password="true" />
</TableRow>

<TableRow>
<TextView android:text="" />
<CheckBox
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Remember Password" />

</TableRow>

<TableRow>
<Button android:text="Log In" />
</TableRow>

</TableLayout>
```

FrameLayout.java

```java
public class FrameLayout extends Activity {

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

}
```

activity_frame.xml

```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"
  android:orientation="vertical"
  tools:context="com.example.vishnums.myapplication.HomePage">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
```

```xml
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Layout Example"
        android:textColor="#ffffff"
        android:textSize="20sp" />

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textView2">

    <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            android:src="@drawable/sample" />

    <Button
            android:id="@+id/button3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal|bottom"
            android:layout_margin="20dp"
            android:text="Set as wallpaper" />
    </FrameLayout>

</RelativeLayout>
```
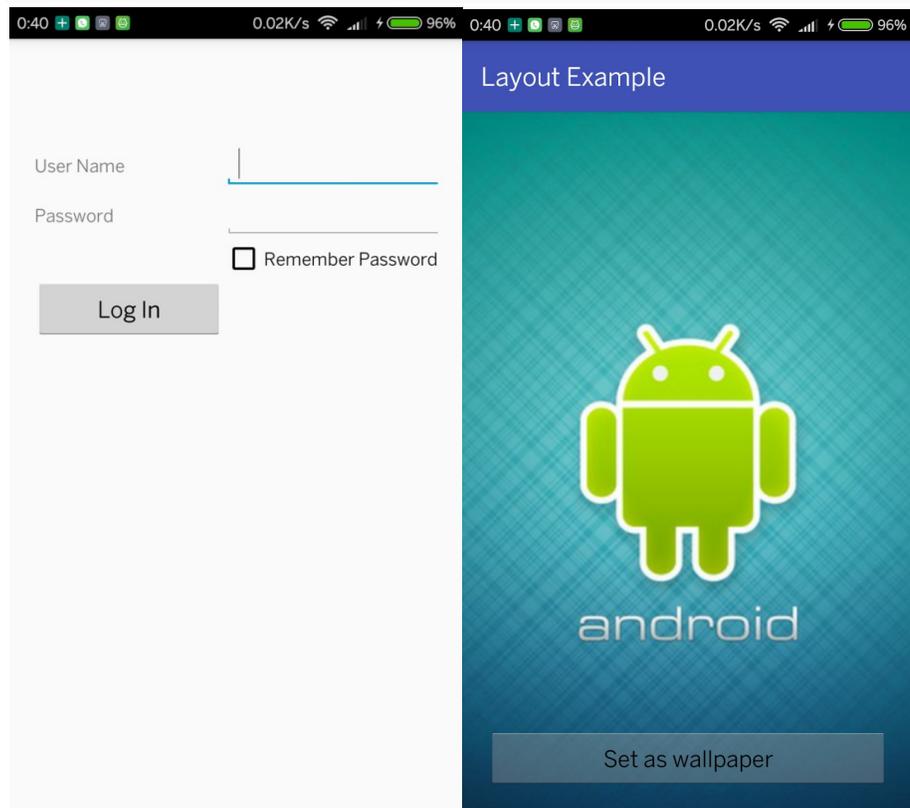
Observations

Input-Output

Screenshots

Relative Layout
Table Layout
Frame Layout

Relative Layout

User Name

Password

Login

User Name

Password

Remember Password

Log In

Layout Example

android

Set as wallpaper
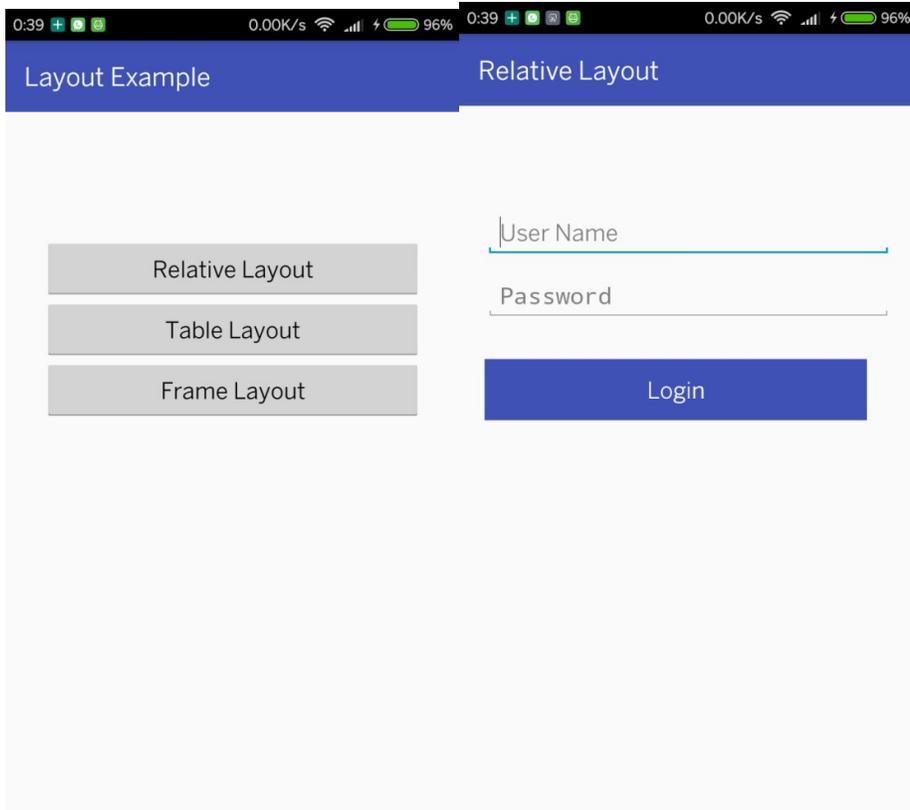
07. Create a simple listview

Aim :

To create an android application with simple listview to list down india's former presidents.

Objective

Create an android application to familiarise listview.

Problem Statement

List should contains the former presidents. By clicking on the listview should show the corresponding president's name in a Toast message.

Apparatus/Tools/Equipments/Components

Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

ListViewActivity.java

```java
public class ListViewActivity extends Activity {

  ListView mListView;
  String[] presidents = {
       "Dr. Rajendra Prasad (1884-1963)",
       "Dr. Sarvepalli Radhakrishnan (1888-1975)",
       "Dr. Zakir Husain (1897-1969)",
       "Shri Varahagiri Venkata Giri (1894-1980)",
       "Dr. Fakhruddin Ali Ahmed (1905-1977)",
       "Shri Neelam Sanjiva Reddy (1913-1996)",
       "Giani Zail Singh (1916-1994)",
       "Shri R Venkataraman (1910-2009)",
       "Dr Shankar Dayal Sharma (1918-1999)",
       "Shri K. R. Narayanan (1920 - 2005)",
       "DR. A.P.J. Abdul Kalam (1931-2015)",
       "Smt Pratibha Devisingh Patil (birth - 1934)"
  };

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

    mListView = (ListView) findViewById(R.id.listView);
```

```
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, presidents);
    mListView.setAdapter(adapter);

    AdapterView.OnItemClickListener itemClickListener = new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View container, int position, long
id) {
            Toast.makeText(getBaseContext(), presidents[position],
Toast.LENGTH_SHORT).show();
        }
    };

    mListView.setOnItemClickListener(itemClickListener);
  }
}
```

activity_list_item.xml

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

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="22sp"
    android:text="Former Presidents"
    android:padding="15dp"
    android:background="@color/colorPrimary"
    android:textColor="#ffffff"
    android:textStyle="bold"/>

<ListView
    android:id="@+id/listView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView3" />

</RelativeLayout>
```

Observations

Input-Output

Screenshots

08. Create a listview with image and text

<u>**Aim**</u>
To create an android application with listview to list down country flag, name and currency.

<u>**Objective**</u>
Create an android applcation to familiarise list view with image and text.

<u>**Problem Statement**</u>
Create an android application with custom listview with image and text. List country image , country name and currency.

<u>**Apparatus/Tools/Equipments/Components**</u>
Android Studio

<u>**Principle/Theory/Flow-chart/Algorithm**</u>

<u>**Procedure**</u>

<u>**CustomListview.java**</u>
import android.app.Activity;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class CustomListview extends Activity {

// Array of strings storing country names
String[] countries = new String[]{
        "India",
        "Pakistan",
        "Sri Lanka",
        "China",
        "Bangladesh",
        "Nepal",
        "Afghanistan",
        "North Korea",
        "South Korea",
        "Japan",
        "India",
        "Pakistan",
        "Sri Lanka",

```java
        "China",
        "Bangladesh",
        "Nepal",
        "Afghanistan",
        "North Korea",
        "South Korea",
        "Japan"
    };

// Array of integers points to images stored in /res/drawable-ldpi/
int[] flags = new int[]{
        R.drawable.india,
        R.drawable.pakistan,
        R.drawable.srilanka,
        R.drawable.china,
        R.drawable.bangladesh,
        R.drawable.nepal,
        R.drawable.afghanistan,
        R.drawable.nkorea,
        R.drawable.skorea,
        R.drawable.japan,
        R.drawable.india,
        R.drawable.pakistan,
        R.drawable.srilanka,
        R.drawable.china,
        R.drawable.bangladesh,
        R.drawable.nepal,
        R.drawable.afghanistan,
        R.drawable.nkorea,
        R.drawable.skorea,
        R.drawable.japan
};

// Array of strings to store currencies
String[] currency = new String[]{
        "Indian Rupee",
        "Pakistani Rupee",
        "Sri Lankan Rupee",
        "Renminbi",
        "Bangladeshi Taka",
        "Nepalese Rupee",
        "Afghani",
        "North Korean Won",
        "South Korean Won",
        "Japanese Yen",
        "Indian Rupee",
        "Pakistani Rupee",
```

```java
            "Sri Lankan Rupee",
            "Renminbi",
            "Bangladeshi Taka",
            "Nepalese Rupee",
            "Afghani",
            "North Korean Won",
            "South Korean Won",
            "Japanese Yen"
    };

/**
    * Called when the activity is first created.
    */
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

// Each row in the list stores country name, currency and flag
List<HashMap<String, String>> aList = new ArrayList<HashMap<String, String>>();

        for (int i = 0; i < 20; i++) {
            HashMap<String, String> hm = new HashMap<String, String>();
            hm.put("name", "Country : " + countries[i]);
            hm.put("currency", "Currency : " + currency[i]);
            hm.put("flag", Integer.toString(flags[i]));
            aList.add(hm);
        }

// Keys used in Hashmap
String[] from = {"name", "currency", "flag"};

// Ids of views in listview_layout
int[] to = {R.id.txt, R.id.cur, R.id.flag};

// Instantiating an adapter to store each items
    // R.layout.listview_layout defines the layout of each item
SimpleAdapter adapter = new SimpleAdapter(getBaseContext(), aList,
R.layout.listview_layout, from, to);

// Getting a reference to listview of main.xml layout file
ListView listView = (ListView) findViewById(R.id.listview);

// Setting the adapter to the listView
listView.setAdapter(adapter);
    }
}
```

main.xml

```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">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Country List"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<ListView
    android:id="@+id/listview"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

    />
</LinearLayout>
```

listview_layout.xml

```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:background="#80d8ff"
  android:orientation="horizontal">

<ImageView
    android:id="@+id/flag"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="img"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="10dp"
```

```
        android:paddingTop="10dp"

    />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

<TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp" />

<TextView
        android:id="@+id/cur"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="10dp" />

</LinearLayout>
</LinearLayout>
```

Observations

Input-Output

Screenshots

Country List

Country : India
Currency : Indian Rupee

Country : Pakistan
Currency : Pakistani Rupee

Country : Sri Lanka
Currency : Sri Lankan Rupee

Country : China
Currency : Renminbi

Country : Bangladesh
Currency : Bangladeshi Taka

Country : Nepal
Currency : Nepalese Rupee

Country : Afghanistan
Currency : Afghani

Country : North Korea
Currency : North Korean Won

Country : South Korea
Currency : South Korean Won

Country : Japan
Currency : Japanese Yen

Country : India
Currency : Indian Rupee

09.Integrate a website inside the application using Webview

Aim
To integrate the webpage inside our android application using Webview.

Objective
To understand the usage of Webview for integrate the webpage inside our own application.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

WebviewActivity.java

```java
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class WebviewActivity extends Activity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_webview);
    WebView webView = (WebView) findViewById(R.id.webView);
    WebSettings webSettings = webView.getSettings();
    webSettings.setBuiltInZoomControls(true);
    webView.loadUrl("http://www.amazon.com");
  }
}
```

activity_webview.xml

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

```xml
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Webview Example"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<WebView
    android:id="@+id/webView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/textView2" />
</RelativeLayout>
```

Observations

Input-Output
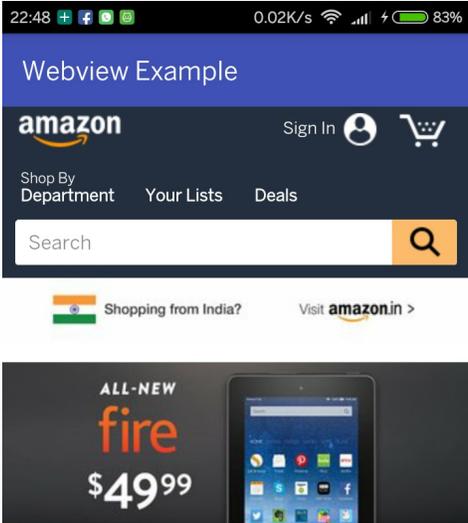
Screenshots

10. Saving Data Using the SharedPreferences Object

Aim

To create an android application to store the details of a student permanently and display the details in another activity using shared preference.

Objective

To understand the use of Shared Preference in android.

Problem statement

1. Create an activity to store Student Roll No,name and branch in Shared preference.
2. Display the details in the next screen.

Apparatus/Tools/Equipments/Components

Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

DetailsInsert.java

```java
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class DetailsInsert extends Activity {

    EditText mEtRollNo, mEtName, mEtBranch;
    Button mBtnSave, mBtnSkip;

    SharedPreferences prefs;

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

        mEtRollNo = (EditText) findViewById(R.id.et_rollno);
```

```java
mEtName = (EditText) findViewById(R.id.et_name);
mEtBranch = (EditText) findViewById(R.id.et_branch);
mBtnSave = (Button) findViewById(R.id.btn_save);
mBtnSkip = (Button) findViewById(R.id.btn_skip);

mBtnSave.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String rllNo = mEtRollNo.getText().toString();
        String name = mEtName.getText().toString();
        String branch = mEtBranch.getText().toString();

        prefs = getSharedPreferences("MyPref", MODE_PRIVATE);
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("rollno", rllNo);
        editor.putString("name", name);
        editor.putString("branch", branch);
        editor.commit();
        Intent intent = new Intent(DetailsInsert.this, DetailsView.class);
        startActivity(intent);

    }
});

mBtnSkip.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent intent = new Intent(DetailsInsert.this, DetailsView.class);
        startActivity(intent);

    }
});

    }

}
```
activity_details_insert.xml

```xml
<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">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
```

```xml
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:background="@color/colorPrimary"
        android:gravity="center_vertical"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Student Details"
        android:textColor="#ffffff"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/et_rollno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:ems="10"
        android:hint="Roll No"
        android:inputType="number"
        android:paddingLeft="10dp" />

    <EditText
        android:id="@+id/et_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:hint="Name"
        android:paddingLeft="10dp" />

    <EditText
        android:id="@+id/et_branch"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:hint="Branch"
        android:paddingLeft="10dp" />

    <Button
        android:id="@+id/btn_save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:background="@color/colorPrimary"
        android:text="Save and Continue"
        android:textColor="#ffffff" />
      android:paddingLeft="10dp" />
```

```xml
<Button
    android:id="@+id/btn_skip"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_margin="20dp"
    android:background="@color/colorPrimary"
    android:text="Skip"
    android:textColor="#ffffff" />
</LinearLayout>
```

DetailsView.java

```java
import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.TextView;

public class DetailsView extends Activity {

  TextView tvObservations;
  SharedPreferences prefs;

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

    tvObservations = (TextView) findViewById(R.id.tv);
    prefs = getSharedPreferences("MyPref", MODE_PRIVATE);
    String rlno = prefs.getString("rollno", "no values");
    String name = prefs.getString("name", "no values");
    String branch = prefs.getString("branch", "no values");

    String Observations = "Roll No : " + rlno
        + "\nName : " + name
        + "\nBanch : " + branch;

    tvObservations.setText(Observations);

  }
}
```

activity_details_view.xml

```xml
<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="com.example.vishnums.myapplication.DetailsView">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Student Details"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<TextView
    android:id="@+id/tv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp"
    android:lineSpacingExtra="15sp"
    android:paddingLeft="20dp"
    android:text=""
    android:textSize="24sp"
    android:textStyle="bold" />

</LinearLayout>
```

Observations

Input-Output

Screenshots

11. Saving Data to Internal Storage as files

Aim
To create an android application to save data in a text file.

Objective
Create an android application to save data in a text file. Load file from memory and show in the view.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure

FileActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;

/**
* Written by Vishnu on 01/02/16.
*/
public class FileActivity extends Activity {

    EditText mEtText;
    Button mBtnSave, mBtnLoad;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_file);
        mEtText = (EditText) findViewById(R.id.et_message);
        mBtnSave = (Button) findViewById(R.id.btn_save);
        mBtnLoad = (Button) findViewById(R.id.btn_load);
```

```java
mBtnSave.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String message = mEtText.getText().toString();
        try {

            FileOutputStream fout = openFileOutput("textfile.txt", MODE_PRIVATE);
            OutputStreamWriter osw = new OutputStreamWriter(fout);


            osw.write(message);
            osw.flush();
            osw.close();

            Toast.makeText(getApplicationContext(), "File saved successfully",

             Toast.LENGTH_LONG).show();
            mEtText.setText("");

        } catch (Exception e) {
            e.printStackTrace();
        }


    }
});

mBtnLoad.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {

            FileInputStream fin = openFileInput("textfile.txt");
            InputStreamReader isr = new InputStreamReader(fin);

            char[] inputBuffer = new char[100];
            String s = "";
            int charRead;
            while ((charRead = isr.read(inputBuffer)) > 0) {
                String readString = String.copyValueOf(inputBuffer, 0, charRead);
                s += readString;
                inputBuffer = new char[100];
            }

            mEtText.setText(s);
```

```
                Toast.makeText(getApplicationContext(), "File loaded successfully",
Toast.LENGTH_LONG).show();


            } catch (Exception e) {
                e.printStackTrace();
            }


        }
    });


    }
}
```

activity_file.xml

```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">

<TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:background="@color/colorPrimary"
        android:gravity="center_vertical"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Save text in Files"
        android:textColor="#ffffff"
        android:textSize="20sp" />

<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:paddingLeft="15dp"
        android:text="Enter some text to save" />

<EditText
        android:id="@+id/et_message"
        android:layout_width="match_parent"
        android:layout_height="100dp"
```

```
        android:ems="10"
        android:hint="Message to save"
        android:paddingLeft="15dp" />

<Button
        android:id="@+id/btn_save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:background="@color/colorPrimary"
        android:text="Save"
        android:textColor="#ffffff" />

<Button
        android:id="@+id/btn_load"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:background="@color/colorPrimary"
        android:text="Load"
        android:textColor="#ffffff" />

</LinearLayout>
```
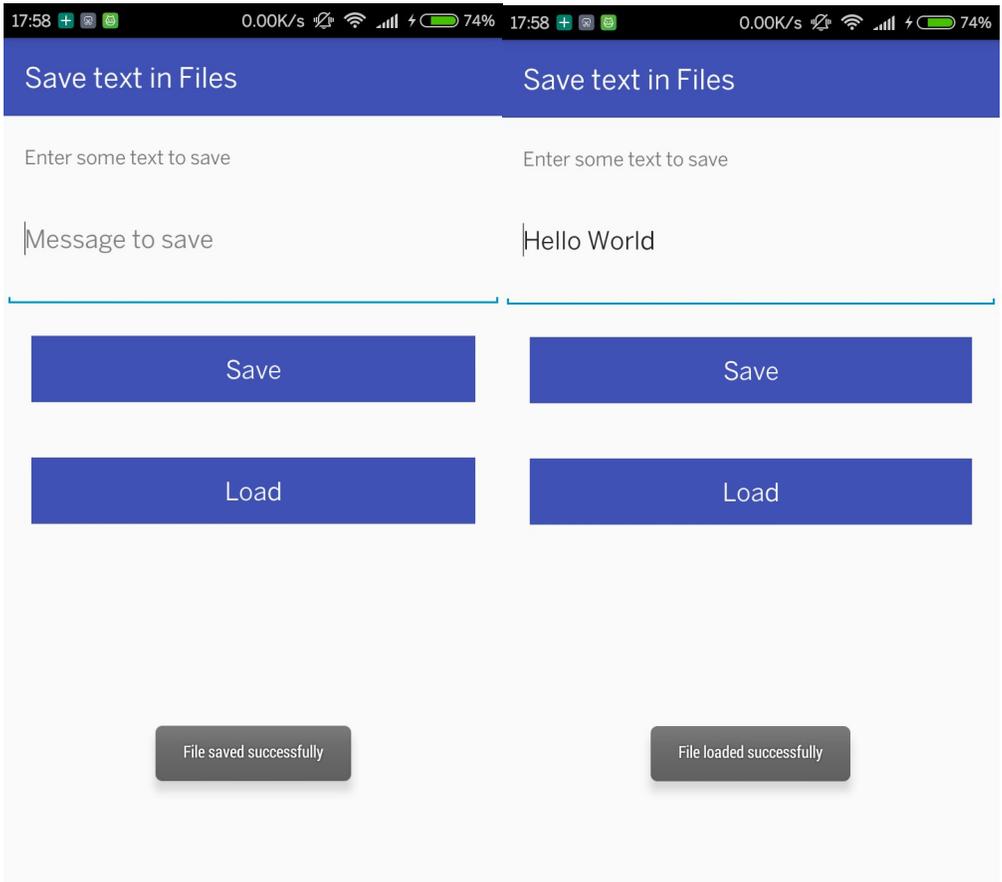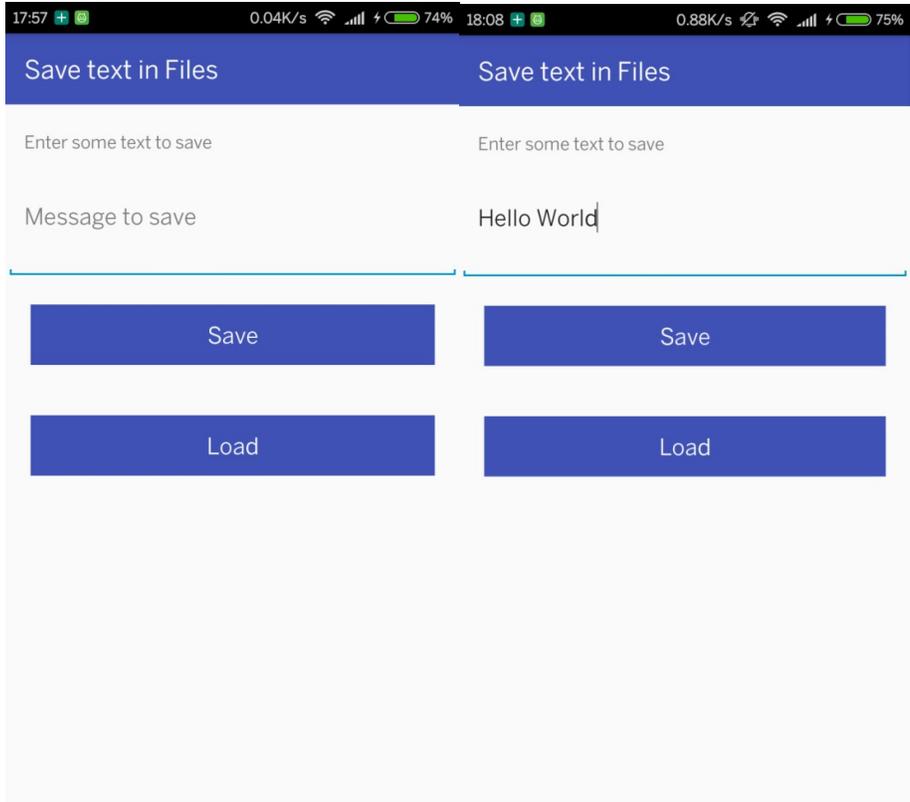
Observations

Input-Output

Screenshots

12. SQLite program for database operations

Aim :
To create an android application to store data in the SQLite database.

Objective
Create an android application to understand database operations

Problem Statement
Create an android application to store user details into a database table, and display the total no of records while inserting.
1. Create a database with name "MyDb" with version number "1".
2. Create a table "userdetails" with field (id,username,email and password).
3. Create a screen to allow the user to enter the details .
4. Display the last inserted record's id in a toast message.

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

DataBase.java

```java
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DataBase extends SQLiteOpenHelper {
  private static final int DATABASE_VERSION = 1;
  private static final String DATABASE_NAME = "Test1";

  public DataBase(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }

  @Override
  public void onCreate(SQLiteDatabase db) {
    String CREATE_USER_TABLE = "CREATE TABLE userdetails(id INTEGER
PRIMARY KEY AUTOINCREMENT ,username TEXT NOT NULL,password TEXT
NOT NULL,email TEXT NOT NULL" + ")";
    db.execSQL(CREATE_USER_TABLE);
  }
```

```java
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }

}
```

```java
import android.app.Activity;
import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class DBAddDetailsOne extends Activity {

    private EditText edUserName, edUserEmail, edUserPassword;
    private Button btnRegister;
    SQLiteDatabase db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        DataBase objdatabase = new DataBase(this);
        db = objdatabase.getWritableDatabase();
        setContentView(R.layout.activity_db_add_one);

        edUserEmail = (EditText) findViewById(R.id.id_useremail);
        edUserName = (EditText) findViewById(R.id.id_username);
        edUserPassword = (EditText) findViewById(R.id.id_user_password);
        btnRegister = (Button) findViewById(R.id.id_btn_register);
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                registerUser();
            }
        });

    }
```

```java
public void registerUser() {
    String _userName, _userPword, _userEmail;
    _userName = edUserName.getText().toString();
    _userPword = edUserPassword.getText().toString();
    _userEmail = edUserEmail.getText().toString();
    ContentValues values = new ContentValues();
    values.put("username", _userName);
    values.put("email", _userEmail);
    values.put("password", _userPword);
    long id = db.insert("userdetails", null, values);
    Toast.makeText(DBAddDetailsOne.this, "No. of Records : " + id,
Toast.LENGTH_SHORT).show();

    }
}
```

activity_db_add_one.xml

```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">

  <TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Add Details"
    android:textColor="#ffffff"
    android:textSize="20sp" />

  <EditText
    android:id="@+id/id_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:ems="10"
    android:hint="Username"
```

```xml
        android:inputType="textPersonName"
        android:paddingLeft="10dp" />

    <EditText
        android:id="@+id/id_useremail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:paddingLeft="10dp" />

    <EditText
        android:id="@+id/id_user_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:paddingLeft="10dp" />


    <Button
        android:id="@+id/id_btn_register"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:background="@color/colorPrimary"
        android:text="Save"
        android:textColor="#ffffff" />
</LinearLayout>
```
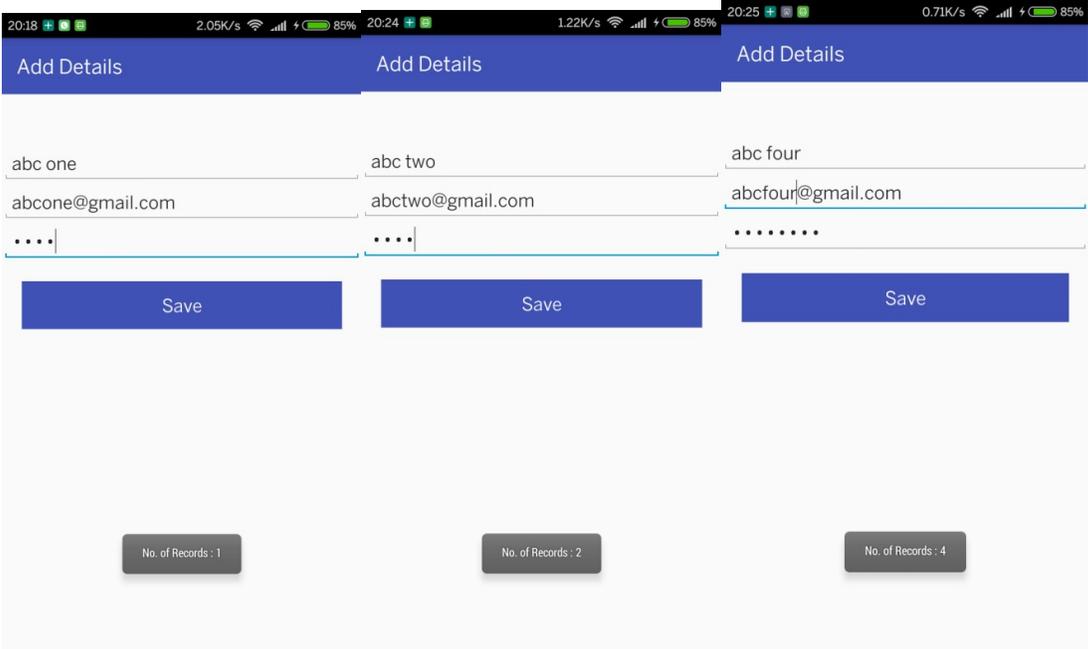
Observations

Screenshots

13. Student database using SQlite

Aim :
To create an android application to manage student database using SQlite database.

Objective
Create an android application to store Students details and manipulate different database operation.

Problem statement

1. Create a database with name "MyDb" with version "1"
2. Create a student table with id,roll number, name, branch, total Score.
3. Create a screen to allow user to input student details, and store the details to table.
4. List the inserted student details using android listview.
5. Allow user to view the complete details by selecting the item in the list
6. Allow the user to edit or remove the details

Apparatus/Tools/Equipments/Components
Android Studio

Principle/Theory/Flow-chart/Algorithm

Procedure
Source Code

DataBaseStudent.java

```java
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

  public class DataBaseStudent extends SQLiteOpenHelper {
  private static final int DATABASE_VERSION = 1;
  private static final String DATABASE_NAME = "student_details";
  public static final String TABLE_STUDENT = "student";
  public static final String KEY_ID = "id";
  public static final String KEY_ROLL_NUMBER = "rollno";
  public static final String KEY_CLASS = "pass";
  public static final String KEY_NAME = "name";
  public static final String KEY_MARK = "mark";

  public DataBaseStudent(Context context) {
     super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }
```

```java
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_USER_TABLE = "CREATE TABLE " + TABLE_STUDENT + "("
            + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT ," +
KEY_NAME + " TEXT," + KEY_CLASS + " TEXT,"
            + KEY_ROLL_NUMBER + " TEXT," + KEY_MARK + " FLOAT)";
        db.execSQL(CREATE_USER_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }
}
```

```java
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class StudentHomePage extends Activity {
    Button btnAdd, btnView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_student_home);
        btnAdd = (Button) findViewById(R.id.btnAdd);
        btnView = (Button) findViewById(R.id.btnview);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent toRegister = new Intent(StudentHomePage.this, StudentRegister.class);
                startActivity(toRegister);
            }
        });
        btnView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent toDetails = new Intent(StudentHomePage.this, ListAllStudentActivity.class);
                startActivity(toDetails);
            }
        });
```

```
    }
}
```

activity_student_home.xml

```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">

<TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:background="@color/colorPrimary"
        android:gravity="center_vertical"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Student Manager"
        android:textColor="#ffffff"
        android:textSize="20sp" />
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical"
        android:padding="30dp">

<Button
            android:id="@+id/btnAdd"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="15dp"
            android:background="@color/colorPrimary"
            android:padding="20dp"
            android:text="ADD STUDENT"
            android:textColor="#ffffff"
            android:textSize="20sp" />
<Button
            android:id="@+id/btnview"
            style="?android:attr/buttonStyleSmall"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="15dp"
```

```
                android:background="@color/colorPrimary"
                android:padding="20dp"
                android:text="VIEW DETAILS"
                android:textColor="#ffffff"
                android:textSize="20sp" />
    </LinearLayout>
</LinearLayout>
```

StudentRegister.java

```java
import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class StudentRegister extends Activity implements View.OnClickListener {
    private EditText edStRollNo, edStName, edStClass, edStMark;
    private Button btnRegister;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        edStName = (EditText) findViewById(R.id.id_student_name);
        edStRollNo = (EditText) findViewById(R.id.id_rollno);
        edStClass = (EditText) findViewById(R.id.id_class);
        edStMark = (EditText) findViewById(R.id.id_student_mark);
        btnRegister = (Button) findViewById(R.id.id_btn_register);
        btnRegister.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        registerUser();
    }

    public void registerUser() {
        String _stRollNo, _stName, _stMark, _stClass;
        _stRollNo = edStRollNo.getText().toString();
        _stClass = edStClass.getText().toString();
        _stName = edStName.getText().toString();
        _stMark = edStMark.getText().toString();
```

```java
DataBaseStudent objdatabase = new DataBaseStudent(this);
SQLiteDatabase db = objdatabase.getWritableDatabase();
Cursor c = db.query(DataBaseStudent.TABLE_STUDENT, null,
DataBaseStudent.KEY_ROLL_NUMBER + "=?", new String[]{_stRollNo}, null, null,
null);
if (c.getCount() != 0) {
    Toast.makeText(StudentRegister.this, "Roll number already exist.",
     Toast.LENGTH_SHORT).show();

  }
   else {
   ContentValues values = new ContentValues();

   values.put(DataBaseStudent.KEY_NAME, _stName);
   values.put(DataBaseStudent.KEY_ROLL_NUMBER, _stRollNo);
   values.put(DataBaseStudent.KEY_CLASS, _stClass);
   values.put(DataBaseStudent.KEY_CLASS, _stClass);
   values.put(DataBaseStudent.KEY_MARK, Float.parseFloat(_stMark));

   long id = db.insert(DataBaseStudent.TABLE_STUDENT, null, values);
   Toast.makeText(StudentRegister.this, "Student details submitted. Count - " + id,
   Toast.LENGTH_SHORT).show();
   edStClass.setText("");
   edStMark.setText("");
   edStName.setText("");
   edStRollNo.setText("");

  }
  c.close();
 }
}
```

activity_register.xml

```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">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
```

```xml
        android:background="@color/colorPrimary"
        android:gravity="center_vertical"
        android:paddingBottom="15dp"
        android:paddingLeft="15dp"
        android:paddingTop="15dp"
        android:text="Add Details"
        android:textColor="#ffffff"
        android:textSize="20sp" />
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp">

<EditText
            android:id="@+id/id_rollno"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Roll"
            android:inputType="number" />

<EditText
            android:id="@+id/id_student_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Username"
            android:inputType="textEmailAddress" />

<EditText
            android:id="@+id/id_class"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Branch"
            android:inputType="text" />

<EditText
            android:id="@+id/id_student_mark"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Score"
            android:inputType="number" />

</LinearLayout>
```

```xml
<Button
    android:id="@+id/id_btn_register"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_margin="20dp"
    android:background="@color/colorPrimary"
    android:text="Save"
    android:textColor="#ffffff" />
</LinearLayout>
```

ListAllStudentActivity.java

```java
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.HashMap;

public class ListAllStudentActivity extends Activity {

    private ListView mListView;
    ListAdapter mAdapter;
    ArrayList<HashMap<String, String>> students = new ArrayList<HashMap<String,
String>>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_all_student);
        mListView = (ListView) findViewById(R.id.listView);
        mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                HashMap<String, String> student = students.get(i);
                String rollNumber = student.get("rollno");
```

```java
            Intent toEdit = new Intent(ListAllStudentActivity.this, EditStudentActivity.class);
            toEdit.putExtra("rollNumber", rollNumber);
            startActivity(toEdit);
        }
    });
  }

  @Override
  protected void onResume() {
    super.onResume();
    getAllDetails();
  }
  private void getAllDetails() {
    students.clear();
    DataBaseStudent objdatabase = new DataBaseStudent(this);
    SQLiteDatabase db = objdatabase.getReadableDatabase();
    Cursor c = db.query(DataBaseStudent.TABLE_STUDENT, null, null, null, null, null,
     null);
    if (c.getCount() != 0) {

        while (c.moveToNext()) {
            HashMap<String, String> _student = new HashMap<String, String>();
            String studentName =
            c.getString(c.getColumnIndex(DataBaseStudent.KEY_NAME));
                    String studentRollNo =
            c.getString(c.getColumnIndex(DataBaseStudent.KEY_ROLL_NUMBER));
            _student.put("name", studentName);
            students.add(_student);
        }

        String[] from = {"name"};
        int[] to = {R.id.id_student_name};
        mAdapter = new SimpleAdapter(ListAllStudentActivity.this, students,
            R.layout.list_items, from, to);
        mListView.setAdapter(mAdapter);


    } else {
        Toast.makeText(ListAllStudentActivity.this, "No Details found",
Toast.LENGTH_SHORT).show();
    }
    c.close();
  }

}
```

activity_list_all_student.xml

```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">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Students List"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"></ListView>

</LinearLayout>
```

EditStudentActivity.java

```java
import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class EditStudentActivity extends Activity implements View.OnClickListener {
```

```java
    private EditText edStRollNo, edStName, edStClass, edStMark;
    private Button btnRegister, btnDelte;
    String rollNumber;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit);
        rollNumber = getIntent().getExtras().getString("rollNumber");

        edStName = (EditText) findViewById(R.id.id_student_name);
        edStRollNo = (EditText) findViewById(R.id.id_rollno);
        edStClass = (EditText) findViewById(R.id.id_class);
        edStMark = (EditText) findViewById(R.id.id_student_mark);
        btnRegister = (Button) findViewById(R.id.id_btn_register);
        btnDelte = (Button) findViewById(R.id.id_btn_delete);
        btnDelte.setOnClickListener(this);
        btnRegister.setOnClickListener(this);
        getDetails();

    }

    private void getDetails() {
        DataBaseStudent objdatabase = new DataBaseStudent(this);
        SQLiteDatabase db = objdatabase.getWritableDatabase();
        Cursor c = db.query(DataBaseStudent.TABLE_STUDENT, null,
        DataBaseStudent.KEY_ROLL_NUMBER + "=?", new String[]{rollNumber}, null,
        null, null);
        if (c.getCount() != 0) {
        String Name, Class;
        Float Mark;
        c.moveToFirst();
        Name = c.getString(c.getColumnIndex(DataBaseStudent.KEY_NAME));
        Class = c.getString(c.getColumnIndex(DataBaseStudent.KEY_CLASS));
        Mark = c.getFloat(c.getColumnIndex(DataBaseStudent.KEY_MARK));
        edStName.setText(Name);
        edStMark.setText(Mark + "");
        edStClass.setText(Class);
        edStRollNo.setText(rollNumber);
        edStRollNo.setFocusable(false);
        }

        c.close();
    }

    @Override
```

```java
    public void onClick(View view) {
        if (view == btnRegister) {
            registerUser();

        } else if (view == btnDelte) {
            deleteUser();
        }
    }

    private void deleteUser() {
        DataBaseStudent objdatabase = new DataBaseStudent(this);
        SQLiteDatabase db = objdatabase.getWritableDatabase();
        db.delete(DataBaseStudent.TABLE_STUDENT,
DataBaseStudent.KEY_ROLL_NUMBER + "=?", new String[]{rollNumber});
        finish();
    }
    public void registerUser() {
        String _stRollNo, _stName, _stMark, _stClass;
        _stRollNo = edStRollNo.getText().toString();
        _stClass = edStClass.getText().toString();
        _stName = edStName.getText().toString();
        _stMark = edStName.getText().toString();

        DataBaseStudent objdatabase = new DataBaseStudent(this);
        SQLiteDatabase db = objdatabase.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(DataBaseStudent.KEY_NAME, _stName);
        values.put(DataBaseStudent.KEY_ROLL_NUMBER, _stRollNo);
        values.put(DataBaseStudent.KEY_CLASS, _stClass);
        values.put(DataBaseStudent.KEY_CLASS, _stClass);
        values.put(DataBaseStudent.KEY_MARK, _stMark);

        long id = db.update(DataBaseStudent.TABLE_STUDENT, values,
DataBaseStudent.KEY_ROLL_NUMBER + "=?", new String[]{rollNumber});
        Toast.makeText(EditStudentActivity.this, "User Updated" + id,
Toast.LENGTH_SHORT).show();
        finish();

    }
}
```

activity_edit.xml

```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"
```

```xml
    android:orientation="vertical">

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@color/colorPrimary"
    android:gravity="center_vertical"
    android:paddingBottom="15dp"
    android:paddingLeft="15dp"
    android:paddingTop="15dp"
    android:text="Update Details"
    android:textColor="#ffffff"
    android:textSize="20sp" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:orientation="vertical"
    android:padding="16dp">

<EditText
    android:id="@+id/id_rollno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Roll No"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/id_student_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Name"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/id_class"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Branch"
    android:inputType="text" />
```

```xml
<EditText
        android:id="@+id/id_student_mark"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Score" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="20dp"
    android:weightSum="2">

<Button
        android:id="@+id/id_btn_register"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginRight="5dp"
        android:layout_weight="1"
        android:background="@color/colorPrimary"
        android:text="Update"
        android:textColor="#ffffff" />

<Button
        android:id="@+id/id_btn_delete"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:background="@color/colorPrimary"
        android:text="Delete"
        android:textColor="#ffffff" />


</LinearLayout>
</LinearLayout>
```
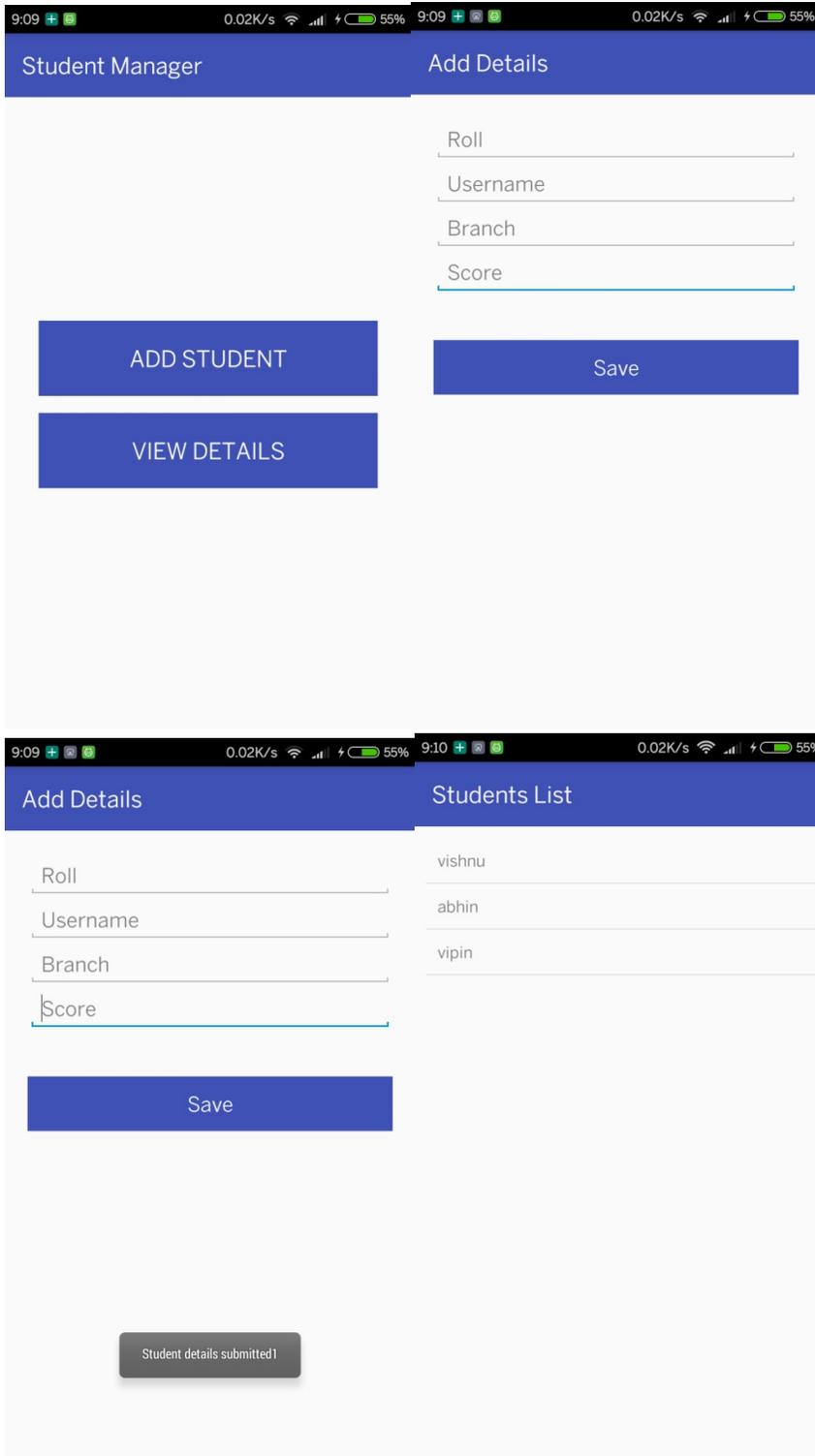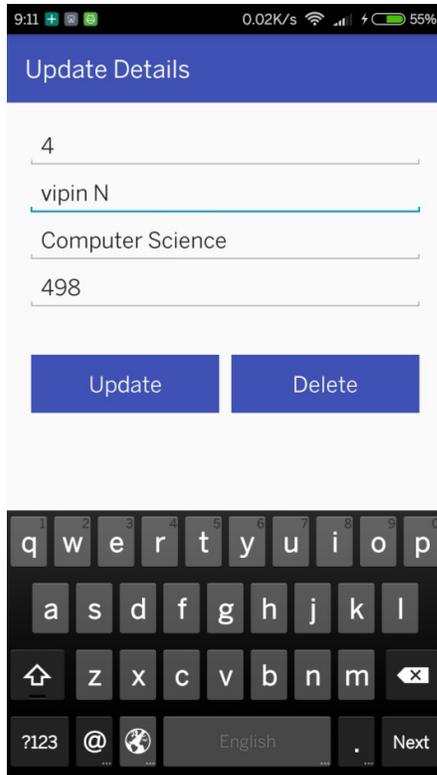
Observations

Input-Output

Screenshots

Sample Questions

1. Create a library database using SQLite
   a. Create a screen to allow user to input book details, and store the details to table.
   b. List the inserted book details using android listview.
   c. Allow user to view the complete details by selecting the item in the list
   d. Allow the user to edit or remove the details

2. Create a Bank database using SQLite
   a. Create a screen to allow user to input account holder details, and store the details to table.
   b. List the inserted details using android listview.
   c. Allow user to view the complete details by selecting the item in the list
   d. Allow the user to edit or remove the details

14. Create a hello world html5 application

<u>Aim :</u>
To create an android application to understand integration of html using webview.

<u>Objective</u>
Create an android application to integrate html using webview.

<u>Apparatus/Tools/Equipments/Components</u>
Android Studio

<u>Principle/Theory/Flow-chart/Algorithm</u>

<u>Steps to include html and javascript file in the android project</u>.

- ➢ html,javascript,css and other asset files are included in the "assets" folder in the project.
- ➢ The path to the assets folder should be:<project>/src/main/assets
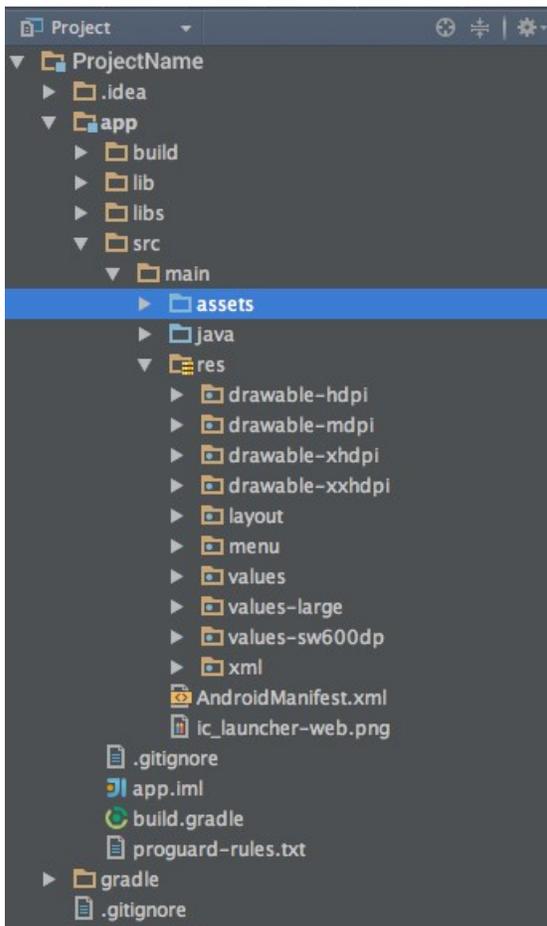
Procedure

helloworld.html

```html
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<b>Helllo World</b>
<br>
<br>
Welcome to html5 application development
</body>
</html>
```

WebviewSample.java

```java
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class WebviewSample extends Activity {

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

    WebView webView = (WebView) findViewById(R.id.webView2);
    WebSettings webSettings = webView.getSettings();
    webSettings.setBuiltInZoomControls(true);
    webSettings.setJavaScriptEnabled(true);
    webView.loadUrl("file:///android_asset/helloworld.html");
  }
}
```

activity_webview_sample.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<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"
  android:padding="16dp">
```

```xml
<WebView
    android:id="@+id/webView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true" />
</RelativeLayout>
```

Observations

Input-Output

Screenshots
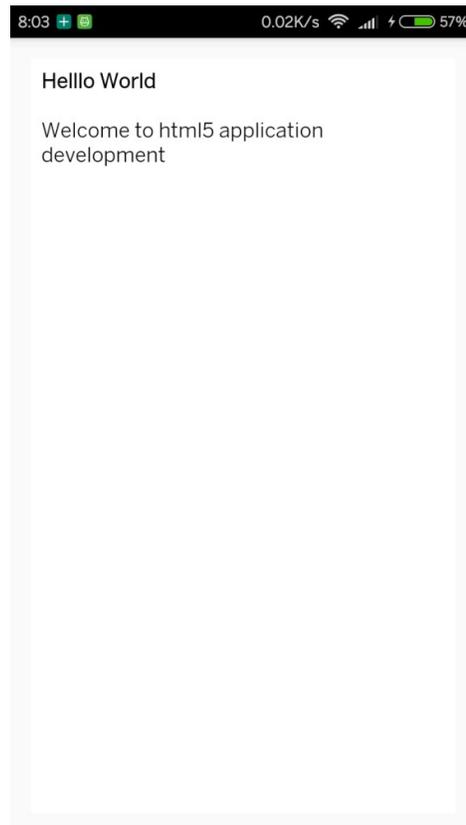
15. Digital Bio Data Application using HTML5

Aim:

To create a digital bio data android application using html5.

Objective

Create an android application using html5 for storing and displaying the Biodata of the user.

Problem statement

1. Create a home screen with two options "Add Details " and "View Details".
2. Add the following input fields in the add details page first name, last name,email,date of birth dnd blood group.
3. Save all the details using html5 local storage technology.
4. Read all details from local storage and display in 'View Details' screen.

Apparatus/Tools/Equipments/Components

Android Studio

Principle/Theory/Flowchart/Algorithm

Procedure

Source Code

WebviewMain.java

```
import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings; import
android.webkit.WebView;

public class WebviewMain extends Activity {

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

    WebView webView = (WebView) findViewById(R.id.webView2);
    WebSettings webSettings = webView.getSettings();
    webSettings.setJavaScriptEnabled(true);//enable javascript
    webSettings.setDomStorageEnabled(true);//Enable local Storage
```

```
      webSettings.setBuiltInZoomControls(false);//Disable zoom
      webView.setWebChromeClient(new WebChromeClient());//To enable the alert
      webView.loadUrl("file:///android_asset/index.html");//Set the initial html file
   }
}
```

activity_webview_sample.xml

```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">

  <WebView android:id="@+id/webView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true" />
</RelativeLayout>
```

index.html

```html
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript" src="script.js" ></script>
</head>
<body bgcolor="#F39B9B" style="text-align:center;margin-top:50px">
<div id="main_screen" style="margin-top:40%">
  <b>DIGITAL BIO DATA</b>
  </br>
  </br>
  <button style="width:100px;70px" onclick="openRegister()">ADD
</br>DETAILS</button>
  <button style="width:100px;70px" onclick="showHomepage()">VIEW
</br>DETAILS</button>
</div>
```

```html
<div id="register_screen" style="display:none">
  <b>ADD DETAILS</b>
  <br>
  <br>Firstname: <br><input id="fname" type="text" name="fname"><br>
  <br>Lastname:<br><input id="lname" type="text" name="lname"><br>
  <br>Email :<br><input id="email" type="email" name="lname"><br>
  <br>Date of Birth:<br><input id="dob" type="date" name="lname"><br>
  <br>Blood Group:<br><input id="blood" type="text" name="lname"><br>
  <br><input type="submit" value="Cancel" onclick="closeRegistration()">
  <input type="submit" value="Submit" onclick="register()">
  <br>
  <br>
</div>
<div id="home_screen" style="display:none">
  hi
</div>

</body>
</html>
```

script.js

```javascript
function openRegister() {
  var div = document.getElementById('register_screen');
  div.style.display = 'block';
  var div = document.getElementById('main_screen');
  div.style.display = 'none';
}
 function closeRegistration() {
  var div = document.getElementById('register_screen');
  div.style.display = 'none';
  var div = document.getElementById('main_screen');
  div.style.display = 'block';
}

function register() {
  var fname = document.getElementById("fname").value;
```

```
var lname = document.getElementById("lname").value; var
email = document.getElementById("email").value; var dob =
document.getElementById("dob").value;
var blood = document.getElementById("blood").value;

localStorage.setItem("fname",fname);
localStorage.setItem("lname",lname);
localStorage.setItem("email",email);
localStorage.setItem("dob",dob);
localStorage.setItem("blood",blood);
closeRegistration();
alert("Details added Successfull")
}
  function showHomepage() {
var div = document.getElementById('main_screen');
div.style.display = 'none';
var div = document.getElementById('home_screen');
div.style.display = 'block';
div.innerHTML = "<br></br><b>DIGITAL BIO DATA</b>"
   +"</br></br><b>First Name :</b> "+localStorage.getItem("fname")
   +"</br><b>Last Name : </b>"+localStorage.getItem("fname")
   +"</br><b>Email : </b>"+localStorage.getItem("email")
   +"</br><b>Date of Birth : </b>"+localStorage.getItem("dob")
   +"</br><b>Blood Group : </b>"+localStorage.getItem("blood");
}
```
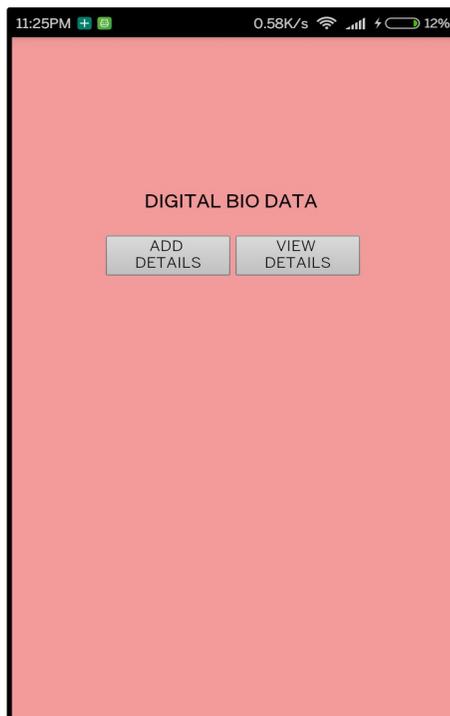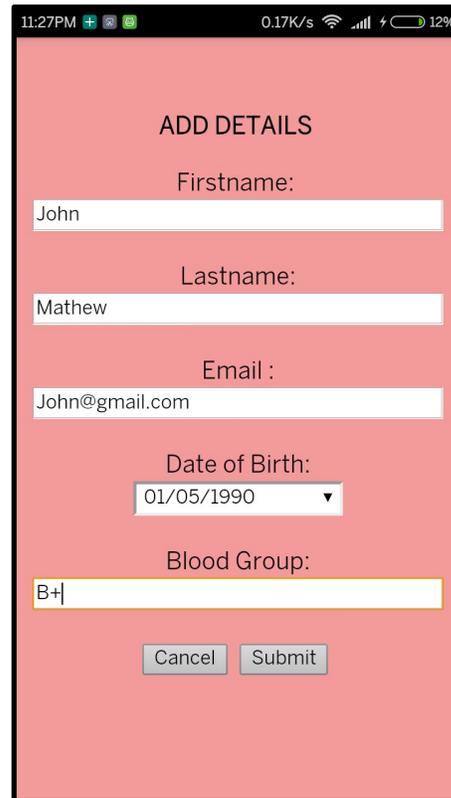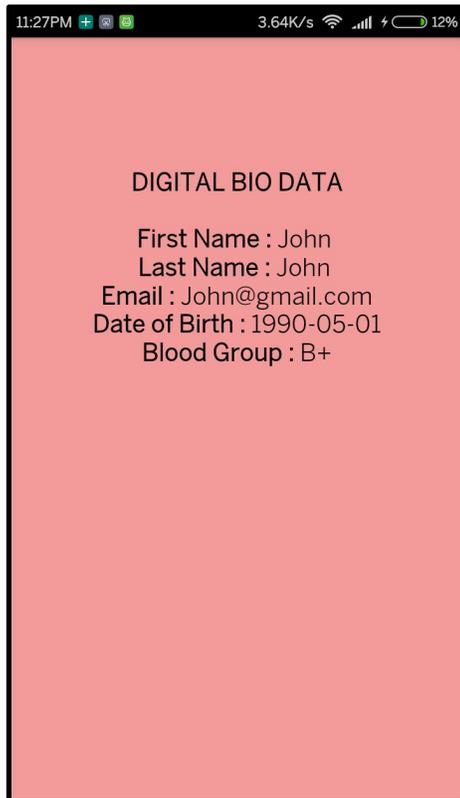
Observations

Input-Output   **Screenshots**

Digital Bio Data (Screen 1)

DIGITAL BIO DATA

First Name : John
Last Name : John
Email : John@gmail.com
Date of Birth : 1990-05-01
Blood Group : B+

Add Details (Screen 2)

ADD DETAILS

Firstname:
John

Lastname:
Mathew

Email :
John@gmail.com
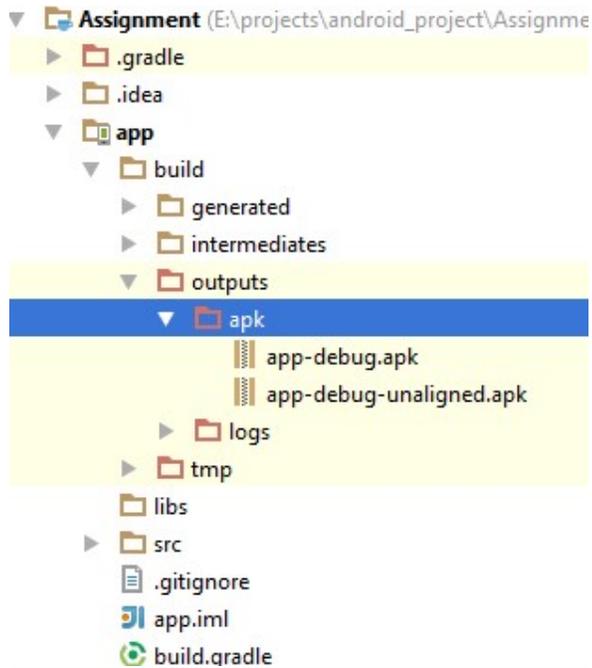
Date of Birth:
01/05/1990 ▼

Blood Group:
B+

Cancel Submit

Sample Questions

1. Create an android application using html5 for storing and dispalying Employee details
2. Create an android application using html5 for storing and displaying customers details of a telephone company

Chapter-7

Installing an application on a mobile device

Locating Apk in Android Studio

Signing Your App in Android Studio

To sign your app in release mode in Android Studio, follow these steps:

1. On the menu bar, click **Build** > **Generate Signed APK**.

2. On the *Generate Signed APK Wizard* window, click **Create new** to create a new keystore.

3. If you already have a keystore, go to step 4.

4. On the *New Key Store* window, provide the required information as shown in figure 1.

5. Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.

Figure 1. Create a new keystore in Android Studio.

6. On the *Generate Signed APK Wizard* window, select a keystore, a private key, and enter the passwords for both. Then click **Next**.

Figure 2. Select a private key in Android Studio.

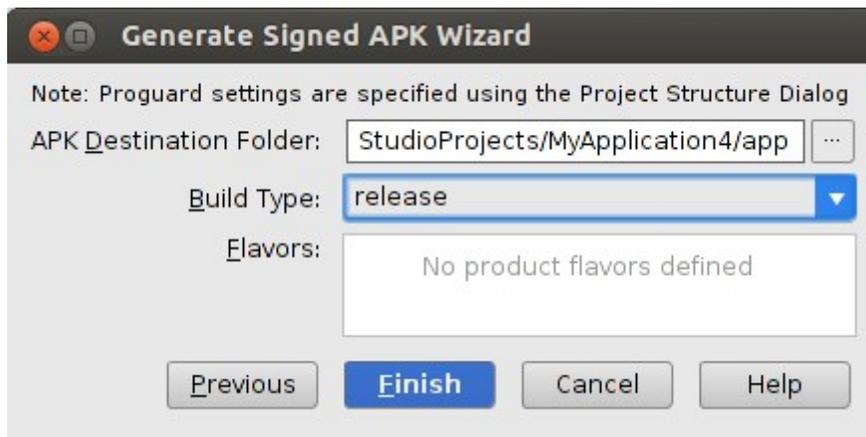7. On the next window, select a destination for the signed APK and click **Finish**.

Figure 3. Generate a signed APK in Android Studio.

COURSE TITLE : SMART DEVICE PROGRAMMING LAB
COURSE CODE : 6138
COURSE CATEGORY : A
PERIODS/WEEK : 5
PERIODS/SEMESTER : 75
CREDITS : 5

<u>**General Outcomes:**</u>
1. To install software tools required for mobile technologies
2. To develop simple android applications
3. To design and implement user interfaces
4. To develop applications using data storage
5. To develop mobile applications using HTML 5.0 and JavaScript

Specific Outcomes:
1.1 Install Android studio
1.2 Install android sdk
1.3 Configure Android Studio for android
1.4 Create android virtual device(Emulator)

2.1 Create Hello World application
2.2 Create activity based applications
2.3 Implement programs for linking Activities with intents
2.4 Implement for Passing Data to the Target Activity
2.5 Implement program for Calling Built-In Applications Using Intents

3.1 Implement program using UI components
      1. To Create three pages using layout system - Linear Layout, Relative Layout and Table Layout respectively.
      2. To Navigate to these screen from the home page that you created In module 1
      3. Implement country listing simple list view.
      4. Include custom list view.
      5. Include a website inside android application using web view.
4.1 Write applications utilising Shared Preferences and file system
      1. Save data using shared preference
      2. To create a application to save data to a file in internal storage.
4.1 Write applications utilising database and SQLite commands
      1. To Create a SQLite database and perform query operations.
      2. To Insert data in database table
      3. To Retrieve and display details in screen
4.2 Create an android application with login page and a home page
4.3 Compile and debug the application.
4.4 Install application in emulator .
5.1 Design and Develop mobile application using HTML 5.0 and JavaScript
5.2 Build HTML 5.0 – android application.
   ➤ Create HTML5 helloworld application.
   ➤ To include use of local storage in HTML5 applications.
5.3 Install an application on a mobile device
5.4 Generate signed apk for play store submission.

References

1. http://developer.android.com/training/index.html

2. Beginning Android Application Development- Wei-Meng Lee- Wrox-First Edition.

3. HTML 5 Mobile Application Development- SAMS publications- Jennifer Kyrnin Jennifer Kyrnin.-First Edition