COLLEGE OF ENGINEERING CHERTHALA

LAB RECORD

20MCA243

MOBILE APPLICATION DEVELOPMENT LAB

CERTIFICATE

This certifies the bona fide work of Mr./Ms. ……....., Registration Number ……., Second Semester MCA, College of Engineering Cherthala, for the academic year 2024-25.

Teacher In Charge External Examiner

Internal Examiner

INDEX

INTRODUCTION TO ANDROID

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.

Android has come a long way from its humble beginnings, as the product of a small startup, all the way to becoming the leading mobile operating system worldwide. Google's introduction of Project Treble in Android Oreo should make it easier for phone makers to update their devices faster.

One challenge for Android device owners that has been an issue for the OS ever since it launched is updating it with the latest security patches, for major feature updates. Google's supported Nexus and Pixel devices consistently receive regular monthly security updates, and the latest version of the OS.

OPERATING SYSTEMS

Different OS run on different types of hardware and are designed for different types of applications. For example, iOS is designed for iPhones and iPad tablets, while Mac desktops and laptops use macOS.

MICROSOFT WINDOWS:

Initial versions of Windows worked with MS-DOS, providing a modern graphical interface on top of DOS's traditional text-based commands. The Windows Start menu helps users find programs and files on their devices.

APPLE IOS

Apple's iOS is one of the most popular smartphone operating systems, second only to Android. It runs on Apple hardware, including iPhones, iPad tablets and iPod Touch media players.

GOOGLE'S ANDROID OS

Android is the most popular operating system in the world judging by the number of devices installed. Users can download custom versions of the operating system.

APPLE MAC OS

Apple's macOS, successor to the popular OS X operating system, runs on Apple laptops and desktops. MacOS is known for its user-friendly features, which include Siri and FaceTime.

LINUX OPERATING SYSTEM

Linux can be run on a wide variety of hardware and is available free of charge over the internet.

Feature	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.

FEATURES OF ANDROID

Android is a powerful operating system competing with Apple 4GS and support great features. Few of them are listed below:

Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Support single direction and bi-directional text.

THE DEVELOPMENT FRAMEWORK: ANDROID ARCHITECTURE

Android operating system is a stack of software components which is roughly divided into five sections and four main layers.

Android is structured in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices.

THE LINUX KERNEL

Positioned at the bottom of the Android software stack, the Linux Kernel provides a level of abstraction between the device hardware and the upper layers of the Android software stack. Based on Linux version 2.6, the kernel provides pre-emptive multitasking, low-level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio.

ANDROID RUNTIME – ART

When an Android app is built within Android Studio it is compiled into an intermediate byte- code format (DEX format). When the application is subsequently loaded onto the device, the Android Runtime (ART) uses a process referred to as Ahead-of-Time (AOT) compilation to translate the byte-code down to the native instructions required by the device processor. This format is known as Executable and Linkable Format (ELF). Each time the application is subsequently launched, the ELF executable version is run, resulting in faster application performance and improved battery life.

This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi- threading, which is intrinsic in the Java language. The Dalvik VM

enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

ANDROID LIBRARIES

In addition to a set of standard Java development libraries (providing support for such general-purpose tasks as string handling, networking and file manipulation), the Android development environment also includes the Android Libraries. These are a set of Java-based libraries that are specific to Android development.

C/C++ LIBRARIES

The Android runtime core libraries are Java-based and provide the primary

APIs for developers writing Android applications. It is important to note, however, that the core libraries do not perform much of the actual work and are, in fact, essentially Java —wrappersǁ around a set of C/C++ based libraries

APPLICATION FRAMEWORK

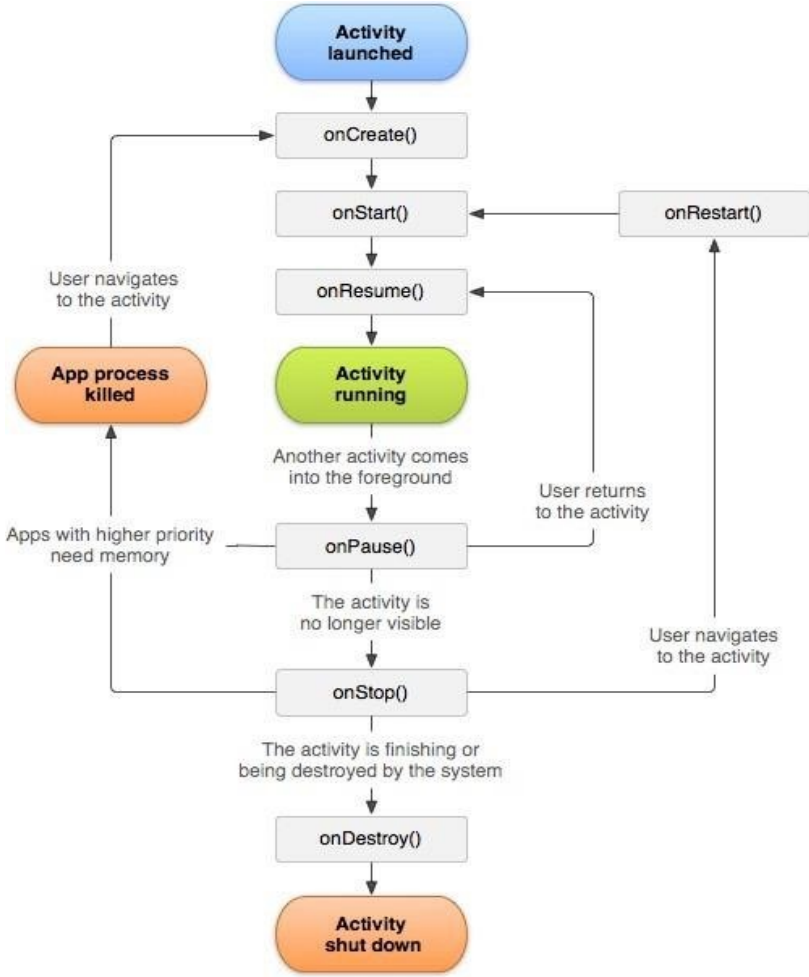
The Application Framework is a set of services that collectively form the environment in which Android applications run and are managed. This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components. This concept is taken a step further in that an application is also able to publish its capabilities along with any corresponding data so that they can be found and reused by other applications.

APPLICATIONS

Located at the top of the Android software stack are the applications. These comprise both the native applications provided with the particular Android implementation (for example web browser and email applications) and the third-party applications installed by the user after purchasing the device.

ANDROID ACTIVITY LIFECYCLE

Let's see the lifecycle methods of android activity.



The Activity class defines the following call backs i.e. events. You don't need to implement all the callbacks methods. However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.

Sr. No.	Callback & Description
1	**onCreate()** This is the first callback and called when the activity is first created.
2	**onStart()** This callback is called when the activity becomes visible to the user.
3	**onResume()** This is called when the user starts interacting with the application.
4	**onPause()** The paused activity does not receive user input and cannot execute any code and call when the c activity is being resumed.
5	**Onstop()** This callback is called when the activity is no longer visible.
6	**onDestroy()** This callback is called before the activity is destroyed by the system.
7	**onRestart()** This callback is called when the activity restarts after stopping it.

EXPERIMENT - I

Aim:

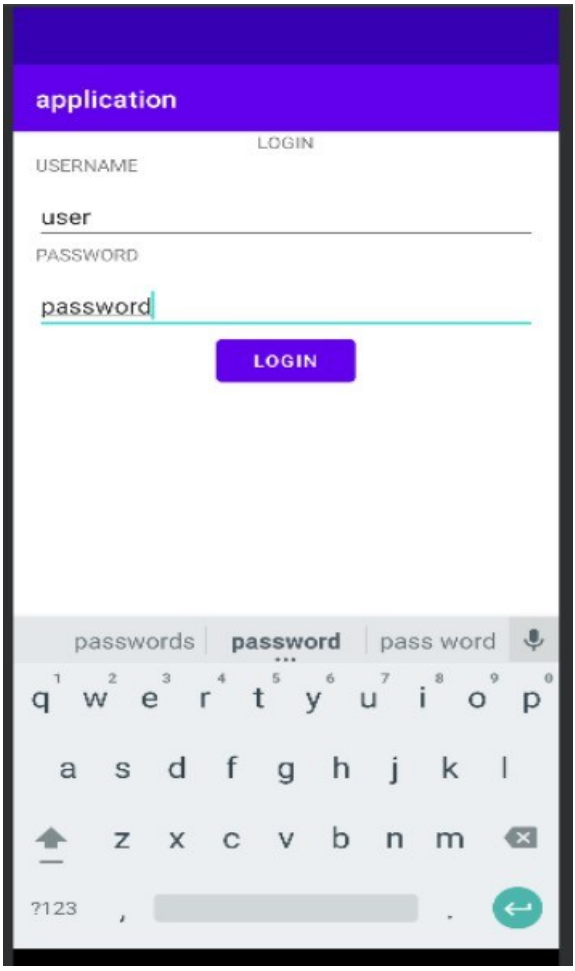
Define a Login Form with username and password using Linear Layout and toast valid Credentials

Source Code:

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <EditText
    android:id="@+id/username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="username" />

  <EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/password"
    android:inputType="textPassword" />

  <Button
    android:id="@+id/login_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/login"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="16dp" />

</LinearLayout>
```

Output:


application
LOGIN
USERNAME
user
PASSWORD
password
LOGIN
passwords
password
pass word
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123

MainActivity.kt

```
package com.example.linearlogin
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity


class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

      val loginButton: Button = findViewById(R.id.login_button)
      val usernameEditText: EditText = findViewById(R.id.username)
      val passwordEditText: EditText = findViewById(R.id.password)

      loginButton.setOnClickListener {
         val username = usernameEditText.text.toString()
         val password = passwordEditText.text.toString()

         if (username == "admin" && password == "password") {
            Toast.makeText(this, "Login successful", Toast.LENGTH_SHORT).show()
         } else {
            Toast.makeText(this, "Invalid credentials", Toast.LENGTH_SHORT).show()
         }
      }
   }
     }
```

Result:

Program is executed successfully and output is obtained

EXPERIMENT - II

Aim:

Implementing basic arithmetic operations of a simple calculator.

Source Code:

Activity_main.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"
   android:id="@+id/main">
   <TextView
      android:id="@+id/input"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textSize="32sp"
      android:padding="16dp"
      android:background="#E0E0E0"
      android:layout_alignParentTop="true"
      android:gravity="end"/>
   <TextView
      android:id="@+id/output"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textSize="24sp"
      android:layout_below="@id/input"
      android:gravity="end"/>
   <GridLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/output"
      android:layout_marginTop="16dp"
      android:rowCount="5"
      android:columnCount="4">
      <Button android:id="@+id/btn1" android:text="1"
```

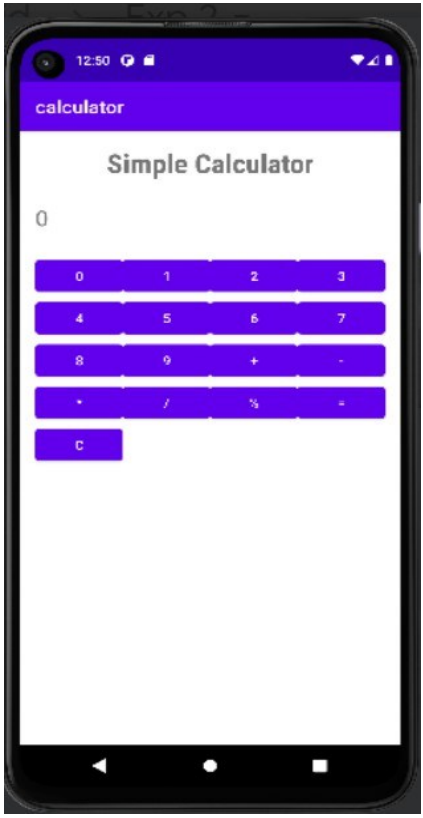
```
        tools:ignore="HardcodedText" />
    <Button android:id="@+id/btn2" android:text="2"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn3" android:text="3"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/add" android:text="+"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn4" android:text="4"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn5" android:text="5"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn6" android:text="6"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/subtract" android:text="-"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn7" android:text="7"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn8" android:text="8"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn9" android:text="9"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/multiply" android:text="*"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btnClear" android:text="C"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/btn0" android:text="0"
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/equal" android:text="="
        tools:ignore="HardcodedText"/>
    <Button android:id="@+id/divide" android:text="/"
        tools:ignore="HardcodedText"/>
  </GridLayout>
</RelativeLayout>
```

MainActivity.kt

```
package com.example.simplecalculator
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {
    private lateinit var inputDisplay: TextView
    private lateinit var outputDisplay: TextView
    private var firstValue: Double = 0.0
    private var secondValue: Double = 0.0
    private var currentOperation: String? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        inputDisplay = findViewById(R.id.input)
        outputDisplay = findViewById(R.id.output)
        val buttons = listOf<Button>(
            findViewById(R.id.btn0),
            findViewById(R.id.btn1),
            findViewById(R.id.btn2),
            findViewById(R.id.btn3),
            findViewById(R.id.btn4),
            findViewById(R.id.btn5),
            findViewById(R.id.btn6),
            findViewById(R.id.btn7),
            findViewById(R.id.btn8),
            findViewById(R.id.btn9),
            findViewById(R.id.add),
            findViewById(R.id.subtract),
            findViewById(R.id.multiply),
            findViewById(R.id.divide),
            findViewById(R.id.equal),
            findViewById(R.id.btnClear)
        )
        for (button in buttons) {
            button.setOnClickListener { onButtonClick(button) }
        }
    }
    private fun onButtonClick(button: Button) {
        when (button.id) {
            R.id.btnClear-> clear()
            R.id.equal-> calculateResult()
            in listOf(R.id.add, R.id.subtract, R.id.multiply, R.id.divide)->
                setOperation(button.text.toString())
            else-> appendToInput(button.text.toString())
        }
```

Output:



```
    }
    private fun clear() {
        inputDisplay.text = ""
        outputDisplay.text = ""
        firstValue = 0.0
        secondValue = 0.0
        currentOperation = null
    }
    private fun appendToInput(value: String) {
        inputDisplay.append(value)
    }
    private fun setOperation(operation: String) {
        if (inputDisplay.text.isNotEmpty()) {
            firstValue = inputDisplay.text.toString().toDouble()
            currentOperation = operation
            inputDisplay.text = ""
        }
    }
    private fun calculateResult() {
        if (inputDisplay.text.isNotEmpty() && currentOperation != null) {
            secondValue = inputDisplay.text.toString().toDouble()
            val result = when (currentOperation) {
                "+"-> firstValue + secondValue
                "-"-> firstValue- secondValue
                "*"-> firstValue * secondValue
                "/"-> if (secondValue != 0.0) firstValue / secondValue else Double.NaN
                else-> 0.0
            }
            outputDisplay.text = result.toString()
            currentOperation = null
        }
    }
}
```

Result:

Program is executed successfully and output is obtained.

Output:


10:44
My Application
Activity Lifecycle
ONCREATE()
ONSTART()
ONPAUSE()
ONSTOP()
ONRESTART()
ONDESTROY()

EXPERIMENT - III

Aim:

Write a program that demonstrates Activity Lifecycle.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:padding="16dp">

   <!-- TextView to display the lifecycle state -->
   <TextView
      android:id="@+id/lifecycleStateText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Activity Lifecycle State"
      android:textSize="18sp"
      android:textAlignment="center"
      android:layout_marginBottom="16dp"/>

   <!-- Button for onCreate -->
   <Button
      android:id="@+id/onCreateButton"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Simulate onCreate" />

   <!-- Button for onStart -->
   <Button
      android:id="@+id/onStartButton"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:text="Simulate onStart" />

    <!-- Button for onResume -->
    <Button
        android:id="@+id/onResumeButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simulate onResume" />

    <!-- Button for onPause -->
    <Button
        android:id="@+id/onPauseButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simulate onPause" />

    <!-- Button for onStop -->
    <Button
        android:id="@+id/onStopButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simulate onStop" />

    <!-- Button for onDestroy -->
    <Button
        android:id="@+id/onDestroyButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simulate onDestroy" />

    <!-- Button for onRestart -->
    <Button
        android:id="@+id/onRestartButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simulate onRestart" />

</LinearLayout>
```

MainActivity.kt

```
package com.example.lifecycleexample

import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

  private lateinit var lifecycleStateText: TextView
  private lateinit var onCreateButton: Button
  private lateinit var onStartButton: Button
  private lateinit var onResumeButton: Button
  private lateinit var onPauseButton: Button
  private lateinit var onStopButton: Button
  private lateinit var onDestroyButton: Button
  private lateinit var onRestartButton: Button

  // onCreate is called when the activity is first created.
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // Initialize views
    lifecycleStateText = findViewById(R.id.lifecycleStateText)
    onCreateButton = findViewById(R.id.onCreateButton)
    onStartButton = findViewById(R.id.onStartButton)
    onResumeButton = findViewById(R.id.onResumeButton)
    onPauseButton = findViewById(R.id.onPauseButton)
    onStopButton = findViewById(R.id.onStopButton)
    onDestroyButton = findViewById(R.id.onDestroyButton)
    onRestartButton = findViewById(R.id.onRestartButton)

    // Default lifecycle state
    lifecycleStateText.text = "Activity Lifecycle State"

    // Set up button listeners to simulate lifecycle methods
    onCreateButton.setOnClickListener {
      lifecycleStateText.text = "onCreate: Activity is created"
```

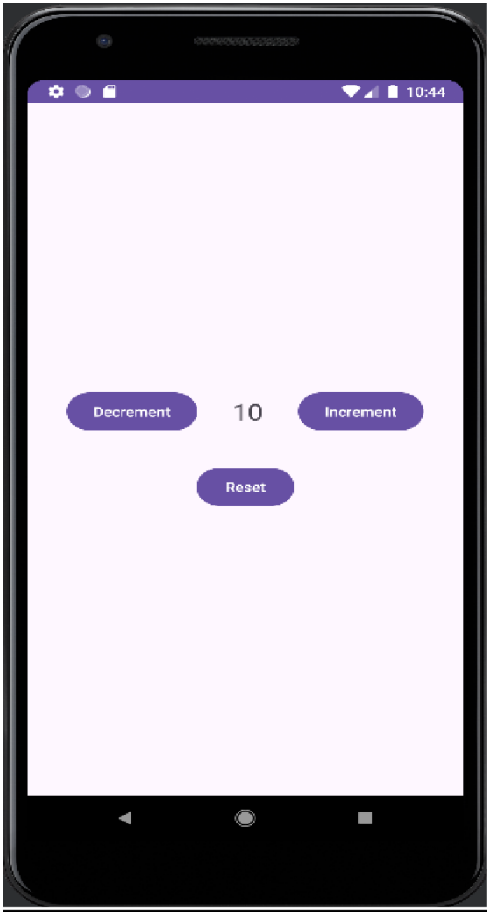
```
    }

    onStartButton.setOnClickListener {
        lifecycleStateText.text = "onStart: Activity is visible"
    }

    onResumeButton.setOnClickListener {
        lifecycleStateText.text = "onResume: Activity is in the foreground"
    }

    onPauseButton.setOnClickListener {
        lifecycleStateText.text = "onPause: Activity is partially hidden"
    }

    onStopButton.setOnClickListener {
        lifecycleStateText.text = "onStop: Activity is no longer visible"
    }

    onDestroyButton.setOnClickListener {
        lifecycleStateText.text = "onDestroy: Activity is being destroyed"
    }

    onRestartButton.setOnClickListener {
        lifecycleStateText.text = "onRestart: Activity is restarting"
    }
}

// onStart is called when the activity becomes visible to the user.
override fun onStart() {
    super.onStart()
    lifecycleStateText.text = "onStart: Activity is visible"
}

// onResume is called when the activity starts interacting with the user.
override fun onResume() {
    super.onResume()
    lifecycleStateText.text = "onResume: Activity is in the foreground"
}

// onPause is called when the system is about to start resuming another activity.
override fun onPause() {
```

```
        super.onPause()
        lifecycleStateText.text = "onPause: Activity is partially hidden"
    }
    override fun onStop() {
        super.onStop()
        lifecycleStateText.text = "onStop: Activity is no longer visible"
    }
    override fun onDestroy() {
        super.onDestroy()
        lifecycleStateText.text = "onDestroy: Activity is being destroyed"
    }
    override fun onRestart() {
        super.onRestart()
        lifecycleStateText.text = "onRestart: Activity is restarting"
    }
}
```

Result:

Program is executed successfully and output is obtained.

Output:



EXPERIMENT - IV

Aim:

Create a counter that increments and decreases.

Source Code:

Activity_main.xml

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

   <TextView
      android:id="@+id/counterTextView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="0"
      android:textSize="40sp"
      android:layout_marginTop="200dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"/>

   <Button
      android:id="@+id/incrementButton"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Increment"
      android:textSize="18sp"
      app:layout_constraintTop_toBottomOf="@id/counterTextView"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      android:layout_marginTop="20dp"/>
```

```
    <Button
        android:id="@+id/decrementButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Decrement"
        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@id/incrementButton"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="20dp"/>

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

MainActivity.kt

```
package com.example.counter

import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    // Declare a variable for the counter
    private var counter = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Get references to the TextView and Buttons
        val counterTextView: TextView = findViewById(R.id.counterTextView)
        val incrementButton: Button = findViewById(R.id.incrementButton)
        val decrementButton: Button = findViewById(R.id.decrementButton)

        // Set an OnClickListener for the increment button
        incrementButton.setOnClickListener {
            // Increment the counter
            counter++
```

```
        // Update the TextView to display the new counter value
        counterTextView.text = counter.toString()
    }

    // Set an OnClickListener for the decrement button
    decrementButton.setOnClickListener {
        // Decrement the counter
        counter--

        // Update the TextView to display the new counter value
        counterTextView.text = counter.toString()
    }
  }
}
```

Result:

Program is executed successfully and output is obtained.

Output:


9:25
My Application
CONSTRAINTLAYOUT
LINEARLAYOUT
GRIDLAYOUT
RELATIVELAYOUT
FRAMELAYOUT
TABLELAYOUT

EXPERIMENT - V

Aim:

Implement validations on various UI controls.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp"
  tools:context=".MainActivity">
  <Button
    android:id="@+id/constraintButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ConstraintLayout" />
  <Button
    android:id="@+id/linearButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="LinearLayout" />
  <Button
    android:id="@+id/gridButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="GridLayout" />
  <Button
    android:id="@+id/relativeButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="RelativeLayout" />
  <Button
    android:id="@+id/frameButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="FrameLayout" />
```

```
  <Button
    android:id="@+id/tableButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
20

    android:text="TableLayout" />
</LinearLayout>
```

MainActivity.kt

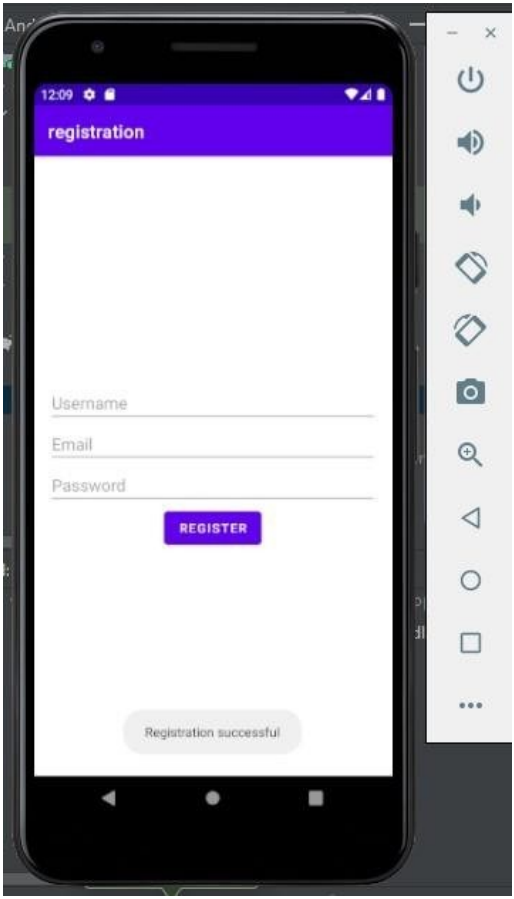
```
package com.example.ui;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button constraintButton = findViewById(R.id.constraintButton);
    Button linearButton = findViewById(R.id.linearButton);
    Button gridButton = findViewById(R.id.gridButton);
    Button relativeButton = findViewById(R.id.relativeButton);
    Button frameButton = findViewById(R.id.frameButton);
    Button tableButton = findViewById(R.id.tableButton);
    View.OnClickListener buttonClickListener = new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String layoutName = ((Button) v).getText().toString();
        displayToken(layoutName);
      }     };

    constraintButton.setOnClickListener(buttonClickListener);
    linearButton.setOnClickListener(buttonClickListener);
    gridButton.setOnClickListener(buttonClickListener);
    relativeButton.setOnClickListener(buttonClickListener);
    frameButton.setOnClickListener(buttonClickListener);
    tableButton.setOnClickListener(buttonClickListener);
```

```
    }

    private void displayToken(String layoutName) {
        Toast.makeText(this, "Token from " + layoutName,
Toast.LENGTH_SHORT).show();
    }
}
```

Result:

Program is executed successfully and output is obtained.

Output:


12:09
registration
Username
Email
Password
REGISTER
Registration successful

EXPERIMENT - VI

Aim:

Design a registration activity and store registration details in local memory of phone using Intents and Shared Preferences.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp"
  android:gravity="center">
  <EditText
    android:id="@+id/usernameEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:inputType="text" />
  <EditText
    android:id="@+id/emailEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress" />
  <EditText
    android:id="@+id/passwordEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword" />
  <Button
    android:id="@+id/registerButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_gravity="center"
    android:text="Register" />
</LinearLayout>
```

MainActivity.kt

```
package com.example.registration;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
  private EditText usernameEditText, emailEditText, passwordEditText;
  private Button registerButton;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    usernameEditText = findViewById(R.id.usernameEditText);
    emailEditText = findViewById(R.id.emailEditText);
    passwordEditText = findViewById(R.id.passwordEditText);
    registerButton = findViewById(R.id.registerButton);
    registerButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String username = usernameEditText.getText().toString();
        String email = emailEditText.getText().toString();
        String password = passwordEditText.getText().toString();
        // Store registration details in SharedPreferences
        SharedPreferences preferences = getSharedPreferences("MyPrefs",
MODE_PRIVATE);
        SharedPreferences.Editor editor = preferences.edit();
        editor.putString("username", username);
        editor.putString("email", email);
        editor.putString("password", password);
        editor.apply();
        Toast.makeText(MainActivity.this, "Registration successful",
Toast.LENGTH_SHORT).show();
```

```
        // Start another activity, e.g., MainActivity, using an Intent
        Intent intent = new Intent(MainActivity.this, MainActivity.class);
        startActivity(intent);
    }    });
}}
```

Result:

Program is executed successfully and output is obtained.

EXPERIMENT - VII

Aim:

Create a Facebook page using Relative Layout; set properties using .xml file.

Source Code:

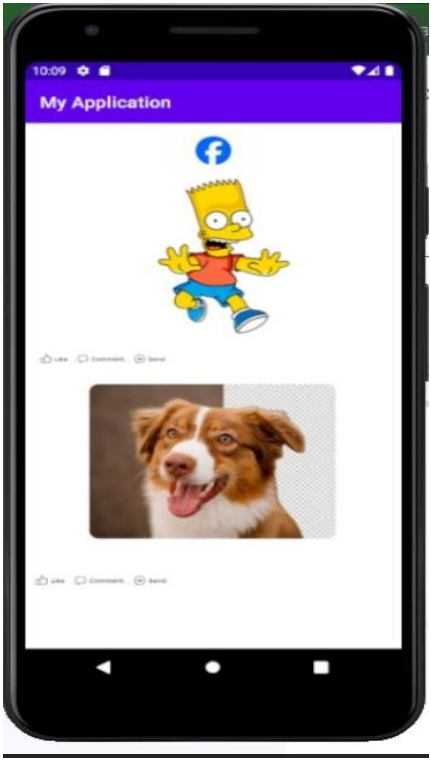
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:paddingLeft="16dp"
   android:paddingRight="16dp" >
   <ScrollView
      android:layout_width="match_parent"
      android:layout_height="match_parent">
      <LinearLayout
         android:layout_width="fill_parent"
         android:layout_height="fill_parent"
         android:orientation="vertical">
         <ImageView
            android:id="@+id/facebookView"
            android:layout_width="200dp"
            android:layout_height="80dp"
            android:layout_gravity="center"
            android:src="@drawable/facebook" />
         <ImageView
            android:id="@+id/imageView4"
            android:layout_width="match_parent"
            android:layout_height="281dp"
            android:src="@drawable/post" />
         <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="40dp"
            android:columnCount="4"
            android:rowCount="4">
```

```
    <!-- Like ImageView -->
    <ImageView
        android:id="@+id/likeImageView"
        android:layout_width="110dp"
        android:layout_height="83dp"
        android:layout_gravity="center"
        android:clickable="true"

        android:onClick="onLikeClick"
        android:src="@drawable/like" />
    <!-- Comment ImageView -->
    <ImageView
        android:id="@+id/commentImageView"
        android:layout_width="111dp"
        android:layout_height="66dp"
        android:layout_row="0"
        android:layout_column="1"
        android:layout_gravity="center"
        android:clickable="true"
        android:onClick="onCommentClick"
        android:src="@drawable/comment" />
    <ImageView
        android:id="@+id/shareImageView"
        android:layout_width="93dp"
        android:layout_height="86dp"
        android:layout_row="0"
        android:layout_column="3"
        android:layout_gravity="center"
        android:clickable="true"
        android:onClick="onShareClick"
        android:src="@drawable/share" />
</GridLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/imageView7"
        android:layout_width="match_parent"
        android:layout_height="281dp"
        android:src="@drawable/dog" />
```

```
<GridLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="40dp"
    android:columnCount="4"
    android:rowCount="4">
    <!-- Like ImageView -->
    <ImageView
        android:id="@+id/likeImageView2"
        android:layout_width="110dp"
        android:layout_height="83dp"
        android:layout_gravity="center"
        android:clickable="true"
        android:onClick="onLikeClick"
        android:src="@drawable/like" />
    <!-- (Your existing ImageView code) -->

    <!-- Comment ImageView -->
    <ImageView
        android:id="@+id/commentImageView2"
        android:layout_width="111dp"
        android:layout_height="66dp"
        android:layout_row="0"
        android:layout_column="1"
        android:layout_gravity="center"
        android:clickable="true"
        android:onClick="onCommentClick"
        android:src="@drawable/comment" />
    <ImageView
        android:id="@+id/shareImageView2"
        android:layout_width="93dp"
        android:layout_height="86dp"
        android:layout_row="0"
        android:layout_column="3"
        android:layout_gravity="center"
        android:clickable="true"
        android:onClick="onShareClick"
        android:src="@drawable/share" />
    <!-- (Your existing ImageView code) -->
</GridLayout>
```

Output:



```
    </LinearLayout>
     </LinearLayout>
  </ScrollView>
</RelativeLayout>
```

MainActivity.kt

```
package com.example.facebook;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;
public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Find the ImageView elements by their IDs
    ImageView facebookView = findViewById(R.id.facebookView );
    ImageView likeImageView = findViewById(R.id.likeImageView);
    ImageView commentImageView = findViewById(R.id.commentImageView);
    ImageView shareImageView = findViewById(R.id.shareImageView);
    likeImageView.setOnClickListener(new View.OnClickListener() {
      public void onClick(View v) {
showToast("You clicked the Like button");
      }    });
    commentImageView.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        showToast("You clicked the Comment button");
      }    });
    shareImageView.setOnClickListener(new View.OnClickListener() {
      public void onClick(View v) {
        showToast("You clicked the Share button");
      }    });  }
  private void showToast(String message) {
    Toast.makeText(this, message, Toast.LENGTH_SHORT).show();  }}
```

Result:

Program is executed successfully and output is obtained.

Output:

EXPERIMENT - VIII

Aim:

Develop an application that toggles image using Frame Layout.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="#BDBABA"
   tools:context=".MainActivity">
   <ImageView
      android:id="@+id/imageView1"
      android:layout_width="427dp"
      android:layout_height="wrap_content"
      android:layout_gravity="left|top"
      android:background="#CACAC8"
      app:srcCompat="@drawable/s1" />
   <ImageView
      android:id="@+id/imageView2"
      android:layout_width="396dp"
      android:layout_height="wrap_content"
      android:layout_gravity="left|top"
      android:visibility="gone"
      app:srcCompat="@drawable/f1" />
</FrameLayout>
```

MainActivity.kt

```
javapackage com.example.frame_layout;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
```

```
ImageView i1,i2;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    i1=(ImageView) findViewById(R.id.imageView1);
    i2=(ImageView) findViewById(R.id.imageView2);
    i1.setOnClickListener(this);
    i2.setOnClickListener(this);
  }
  @Override
  public void onClick(View v) {
    if(v.getId()==R.id.imageView1)
    {
      i1.setVisibility(v.GONE);
      i2.setVisibility(v.VISIBLE);
    }
    else
    {
      i2.setVisibility(v.GONE);
      i1.setVisibility(v.VISIBLE);
    }
  }
}
```

Result:

Program is executed successfully and output is obtained.

EXPERIMENT - IX

Aim:

Implement Options Menu to navigate to activities

Source Code:

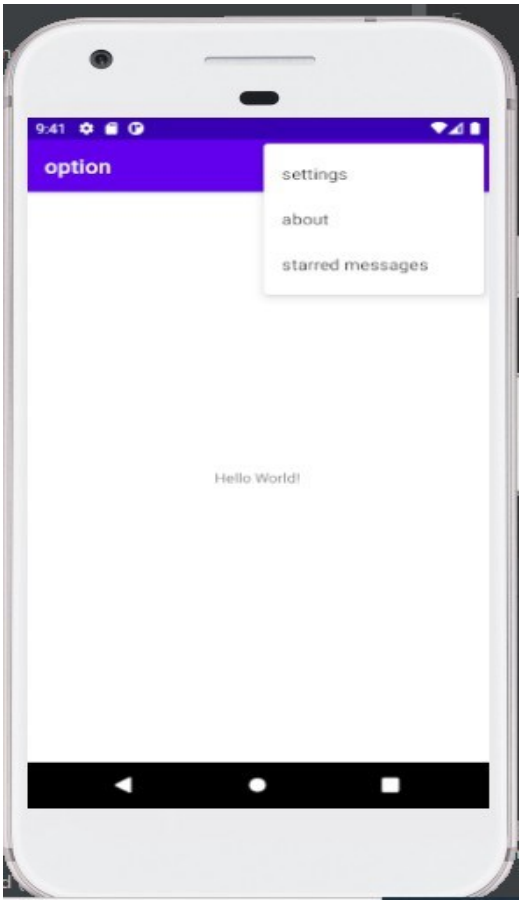
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

menu_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
  android:id="@+id/settings"
  android:title="settings"/>
  <item
    android:id="@+id/about"
    android:title="about"/>
  <item
    android:id="@+id/msgs"
    android:title="starred messages"/>
</menu>
```

Output:



activity_settingspage.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent"
tools:context=".settingspage"> settingspage.java package
com.example.option; import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; public class settingspage extends AppCompatActivity
{ @Override protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_settingspage);
```

MainActivity.kt

```
package com.example.option;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_main,menu);
    return super.onCreateOptionsMenu(menu);
  }
  @Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch(item.getItemId())
```

```
    {
      case R.id.settings:
        Intent intent = new Intent(MainActivity.this,settingspage.class);
        startActivity(intent);
        break;
      case R.id.about:
        Toast.makeText(this,"you clicked
about",Toast.LENGTH_LONG).show();
      break;
      case R.id.msgs:
        Toast.makeText(this,"you clicked starred
messages",Toast.LENGTH_LONG).show();
        break;
    }
    return super.onOptionsItemSelected(item);
  }
}
```

Result:

Program is executed successfully and output is obtained.

EXPERIMENT - X

Aim:
Develop an application that with explicit intent.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="switchActivity"
    android:text="Button"
    app:layout_constraintBottom_toTopOf="@+id/editText1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.5" />
  <EditText
    android:id="@+id/editText1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:text="Enter Your Name"
    app:layout_constraintTop_toBottomOf="@+id/button"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

  <EditText
    android:id="@+id/editText2"
```

Output:


10:10
My Application
BUTTON
activity1
ashwathy
12


10:11
My Application
welcome ashwathy age : 12

```
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:ems="10"
            android:text="Enter age"
            app:layout_constraintTop_toBottomOf="@+id/editText1"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />
        </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
  EditText name;
  EditText age;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    name=findViewById(R.id.editText1);
    age=findViewById(R.id.editText2);
  }
  public void switchActivity(View view){
    Intent intent=new Intent(this,MainActivity2.class);
    intent.putExtra("user",name.getText().toString());
    intent.putExtra("age",age.getText().toString());
    startActivity(intent);
  }}
```

Activity_Main2.xml

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

```
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Activity 2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_margin="16dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_Main2.kt

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity2 extends AppCompatActivity {
TextView tv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        Intent intent=getIntent();
        String user=intent.getStringExtra("user");
        String age =intent.getStringExtra("age");
        tv=findViewById(R.id.textView);
        tv.setText("welcome"+user+"age :"+age);
    }
}
```

Result:

Program is executed successfully and output is obtained.

EXPERIMENT - XI

Aim:

Develop an application that implements Spinner component and perform event Handling.

Source Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <TextView
    android:id="@+id/textview1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:layout_marginTop="50dp"
    android:layout_marginLeft="150dp"/>
  <Spinner
    android:id="@+id/spinner2"
    android:layout_height="50dp"
    android:layout_width="200dp"
    android:layout_marginTop="100dp"
    android:layout_marginLeft="110dp"/>
</RelativeLayout>
```

MainActivity.kt

```
package com.example.spin;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
```

Output:


11:21
spin
values1 text
values1
values2
values3
value4
value5


11:23
spin
values3 text
values3

```
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
   String []names = {"values1","values2","values3","value4","value5"};
   String []text = {"values1 text","values2 text","values3 text","value4
text","value5 text"};
   ArrayAdapter<String> adapter;
   Spinner spinner;
   TextView textView;
   @Override
protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      spinner = findViewById(R.id.spinner2);
      textView = findViewById(R.id.textview1);
      adapter = new ArrayAdapter<String>(getApplicationContext(),
android.R.layout.simple_list_item_1,names);
      spinner.setAdapter(adapter);
      spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
         @Override
         public void onItemSelected(AdapterView<?> adapterView, View
view, int i, long l) {
            switch (i)
            {
               case 0: textView.setText(""+text[i]);
                  break;
               case 1: textView.setText(""+text[i]);
                  break;
               case 2: textView.setText(""+text[i]);
                  break;
               case 3: textView.setText(""+text[i]);
                  break;
               case 4: textView.setText(""+text[i]);
                  break;
            }
         }
         @Override
         public void onNothingSelected(AdapterView<?> adapterView) {
         }      });   }}
```

Result:

Program is executed successfully and output is obtained.

EXPERIMENT - XII

Aim:

Create database using SQLite and perform INSERT and SELECT.

Source Code:

Activity_main.xml

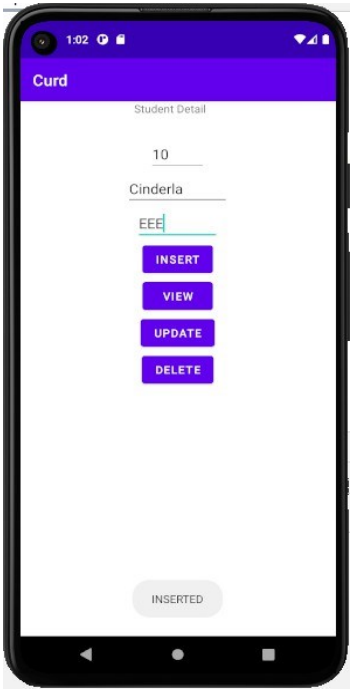
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
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

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="STUDENT DETAILS"
    android:layout_centerHorizontal="true"
    />

  <EditText
    android:id="@+id/edit1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter Rollno"
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/textView"
    />

  <EditText
    android:id="@+id/edit2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter Name"
```

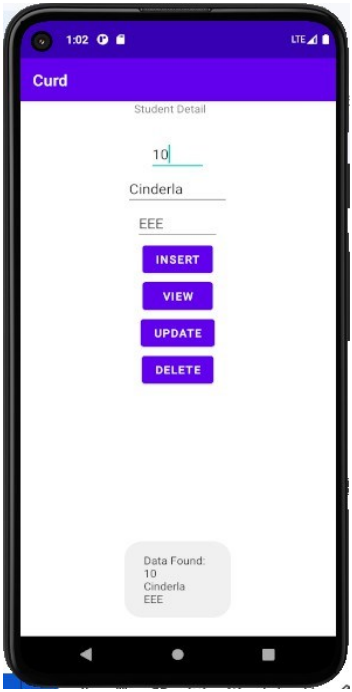
```
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/edit1"
    />

  <EditText
    android:id="@+id/edit3"
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"
    android:hint="Enter Department"
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/edit2"
    />

  <Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="INSERT"
    android:onClick="onInsert"
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/edit3" />

  <Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="READ"
    android:onClick="onRead"
    android:layout_margin="10dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/button2" />

</RelativeLayout>
```

MainActivity.kt

```
package com.example.sql;
import androidx.appcompat.app.AppCompatActivity;
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.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
  TextView textView;
  EditText edit1, edit2, edit3;
  Button button1, button2, button3, button4;
  String rno;
  String name;
  String dept;

  SQLiteDatabase db;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    textView = findViewById(R.id.textView);
    edit1 = findViewById(R.id.edit1);
    edit2 = findViewById(R.id.edit2);
    edit3 = findViewById(R.id.edit3);
    button1 = findViewById(R.id.button1);
    button2 = findViewById(R.id.button2);
    button3 = findViewById(R.id.button3);
    button4 = findViewById(R.id.button4);
    DBHelper dbHelper = new DBHelper(this);
    db = dbHelper.getWritableDatabase();
    db = dbHelper.getReadableDatabase(); }
  public void onInsert(View view) {
    rno = edit1.getText().toString();
    name = edit2.getText().toString();
    dept = edit3.getText().toString();
    if(rno.equals("")  || name.equals("") || dept.equals(""))
```

Output:


1:02
Curd
Student Detail
10
Cinderla
EEE
INSERT
VIEW
UPDATE
DELETE
INSERTED


1:02
LTE
Curd
Student Detail
10
Cinderla
EEE
INSERT
VIEW
UPDATE
DELETE
Data Found:
10
Cinderla
EEE

```
   {
        Toast.makeText(this, "Please Enter Values",
Toast.LENGTH_SHORT).show();}
      else
      {
        ContentValues values = new ContentValues();
        values.put("rollno", rno);
        values.put("name", name);
        values.put("dept", dept);
        db.insert("student", null, values);
        Toast.makeText(this, "Inserted", Toast.LENGTH_SHORT).show();
      }}
   public void onRead(View view) {
 }
 }
```

DBHelper.java

```
package com.example.sql;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;
public class DBHelper extends SQLiteOpenHelper {
   public DBHelper(@Nullable Context context) {
      super(context, "student.db", null , 1 );
   }

   @Override
   public void onCreate(SQLiteDatabase sqLiteDatabase) {
      sqLiteDatabase.execSQL("create table student(rollno int, name
varchar(20), dept
varchar(10))");
}
   @Override
   public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
      sqLiteDatabase.execSQL("drop table if exists student");
      onCreate(sqLiteDatabase);
   }
}
```

Result:

Program is executed successfully and output is obtained.