Program1:Creating “Hello world” Application.

Procedure:

1. Open Android Studio and create a new project.
2. Choose 'Empty Views Activity' and name your project.

`activity_main.xml`code:

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

`MainActivity.java`

```
package com.example.helloworld;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Output:
The application will launch and display the text 'Hello World!' centered on the screen.

Program 2:Creating an application that displays message based on the screen orientation.

Procedure:

Open Android Studio and create a new Empty Views Activity project.

MainActivity.java Code:

```
package com.example.orientation;

import android.content.res.Configuration;
import android.os.Bundle;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

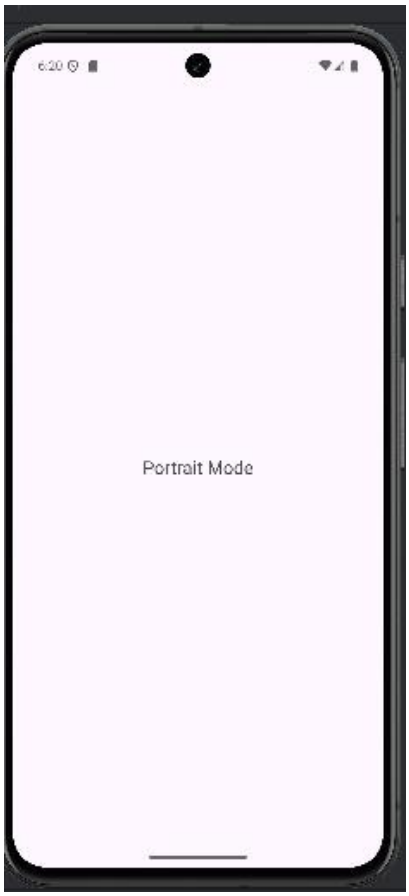
```
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        TextView t = findViewById(R.id.tid);
        if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_LANDSCAPE) {
            t.setText("Landscape Mode");
        } else {
            t.setText("Portrait Mode");
        }
    }
}
```

activity_main.xml Code:

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

    <TextView
        android:id="@+id/tid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Orientation"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

Output:



Program 3:Create an application to develop Login window using UI controls.

Procedure:

1. Open Android Studio and create a new project.
2. Choose an Empty Views Activity and name it MainActivity.
3. Open activity_main.xml.
4. Add two EditText components for username and password.
5. Add one Button component labeled “Login”.
6. Use proper layout structure (e.g., LinearLayout or ConstraintLayout).
7. Set hints and input types for EditText.
8. Apply necessary styling like padding, margin, and alignment.
9. Run the app to verify the layout visually.

XML Layout (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="24dp"
    android:gravity="center"
    android:background="#FFFFFF">

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

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_marginTop="12dp" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_marginTop="18dp" />
</LinearLayout>
```

Program 4: Create a sample application with login module(check user name and password) On successful login change Textview “Login Successful”. On login fail alert using Toast “login fail”

Procedure:

1. Open Android Studio and create a New Project named EmailApp.
2. Choose Empty Views Activity and name the main activity as MainActivity.
3. Open activity_main.xml and design the layout using EditText for username and password, and a Button for login.
4. In MainActivity.java, write code to handle button click and validate user input.
5. Run the app on an emulator or physical device.
6. Enter various combinations of usernames and passwords to check the validation logic.

Source Code:

MainActivity.java

```
package com.example.emailapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
```

```
systemBars.bottom);
            return insets;
        });

        EditText et1 = findViewById(R.id.usr);
        EditText et2 = findViewById(R.id.pwd);
        Button buttonLogin = findViewById(R.id.btn);

        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = et1.getText().toString().trim();
                String password = et2.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Username or Password is empty",
Toast.LENGTH_SHORT).show();
                } else if (username.equals("admin") && password.equals("welcome")) {
                    Toast.makeText(MainActivity.this, "Login Successful",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Incorrect username or password",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="24dp"
    android:gravity="center"
    android:background="#EFEFEF">
```

```
    <EditText
        android:id="@+id/usr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Username"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/pwd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Password"
        android:inputType="textPassword"
        android:layout_marginTop="12dp" />

    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_marginTop="18dp"
        android:backgroundTint="@android:color/holo_blue_dark"
        android:textColor="#FFFFFF"/>
</LinearLayout>
```

Output:

Input: Empty username/password → Output: “Username or Password is empty”
Input: admin / welcome → Output: “Login Successful”
Input: any other values → Output: “Incorrect username or password”

Program 5:Create an application to implement new activity using explicit intent, implicit intent and content provider.

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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="36dp"
        android:text="Explicit Intent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.365"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Implicit Content"
        app:layout_constraintBottom_toTopOf="@+id/btn3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.376"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn1"
        app:layout_constraintVertical_bias="0.471" />
    <Button
        android:id="@+id/btn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="468dp"
        android:text="Open Contacts"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.376"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.intent;
```

```
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 androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
        Button b1=findViewById(R.id.btn1);
        Button b2=findViewById(R.id.btn2);
        Button b3=findViewById(R.id.btn3);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent exintent=new Intent(MainActivity.this, MainActivity2.class);
                startActivity(exintent);
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            Intent impintent=new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.google.com"));
            startActivity(impintent);
        }
    });
    b3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
            startActivity(intent);
        }
    });
  }
}
```

Steps to Create Activity 2 in Android Studio

Steps to Create Activity 2

1. 1. Right-click on the `java` folder in `app > java > your_package_name`.
2. 2. Select New > Activity > Empty view Activity.
3. 3. Enter the Activity Name as `MainActivity2`.
4. 4. Keep the layout name as `activity_main2.xml` (or rename as needed).
5. 5. Click Finish. This creates the Java class, layout XML, and updates the manifest.

Java Code for MainActivity2.java

```
package com.example.intent;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity2 extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2); // Refers to XML file
  }
```

```
}
```

XML Layout for 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.564"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.516" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Program 6:Creating an UI with All Views

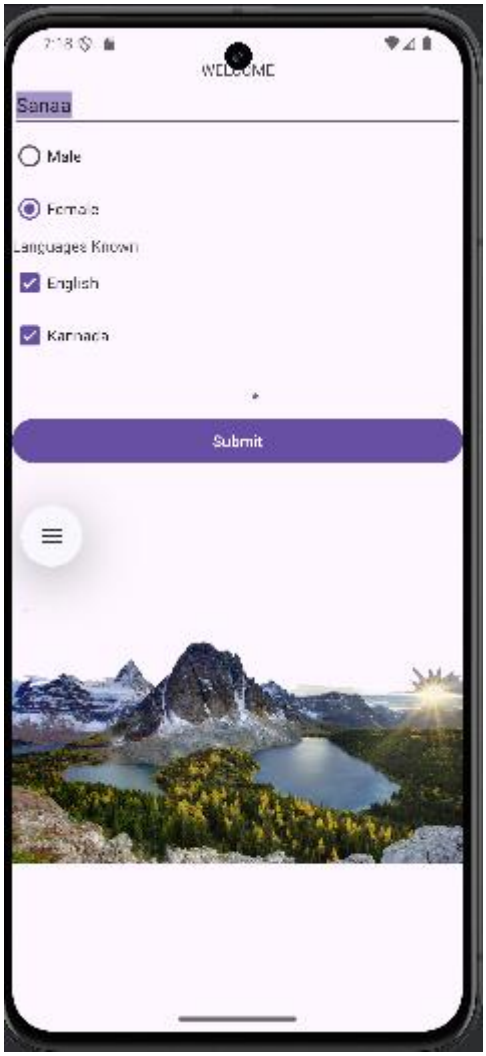
activity_main.xml Code:

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

```
tools:context=".MainActivity">

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

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="WELCOME"
        android:textAlignment="center" />

    <EditText
        android:id="@+id/editTextText3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text"
        android:text="Enter your name here" />

    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <RadioButton
            android:id="@+id/radioButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Male" />

        <RadioButton
            android:id="@+id/radioButton2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Female" />

        <TextView
            android:id="@+id/textView5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Languages Known" />

        <CheckBox
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="English" />

        <CheckBox
            android:id="@+id/checkBox2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Kannada" />

        <ProgressBar
            android:id="@+id/progressBar"
            style="?android:attr/progressBarStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/button4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Submit" />

        <ImageView
            android:id="@+id/imageView2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/img" />

    </RadioGroup>


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

Output:



Program 7:Creating Menu in Application

Step 1: Create menu folder

1. Go to the 'res' directory in Android Studio.
2. Right-click on 'res' > New > Android Resource Directory.
3. In the 'Resource type' dropdown, select 'menu'.
4. Click 'OK'.

Step 2: Create menu_main.xml file

1. Right-click on the newly created 'menu' folder.
2. Select New > Menu Resource File.
3. Name the file as menu_main.xml.

4. Click 'OK'.

Step 3: Add menu items to menu_main.xml
write the following code inside menu_main.xml:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
   <item
      android:id="@+id/action_search"
      android:title="Search"
      android:showAsAction="ifRoom" />
   <item
      android:id="@+id/action_settings"
      android:title="Settings"
       android:showAsAction="ifRoom" />
</menu>
```

Step 4: Add Toolbar using Palette

1. Open activity_main.xml.
2. Go to the 'Palette' on the left.
3. Under 'Widgets', drag and drop 'Toolbar' into your layout.
4. Give it an ID like: android:id="@+id/toolbar".

Step 5: Java Code for Toolbar and Menu
Use the following code in MainActivity.java:

```
package com.example.actionbarr;

import android.os.Bundle;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
```

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        // Initialize and set the toolbar
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

    // Menu creation
    @Override
    public boolean onCreateOptionsMenu(android.view.Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    // Menu item click actions
    @Override
    public boolean onOptionsItemSelected(android.view.MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_search) {
            Toast.makeText(this, "Search clicked", Toast.LENGTH_SHORT).show();
            return true;
        } else if (id == R.id.action_settings) {
            Toast.makeText(this, "Settings clicked", Toast.LENGTH_SHORT).show();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
```

}
Output:

Program 8:Create an Application with Custom Designed Opening Screen

Android Lab Manual – Splash Screen using Explicit Intent

1. Create Two Activities

• SplashActivity (First screen shown when the app launches)

• MainActivity (main content screen)

2. spashActivity.java

```
public class spashActivity extends AppCompatActivity {

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

```
        setContentView(R.layout.activity_spash);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v,
insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(spashActivity.this, MainActivity.class);
startActivity(intent);
                finish();
            }
        },3000);
    }
}
```

3. activity_spash.xml (Splash Screen Layout with ConstraintLayout)

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".spashActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="172dp"
        android:layout_height="20dp"
        android:layout_marginBottom="176dp"
        android:text="Welcome to app"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.389"
        app:layout_constraintStart_toStartOf="parent" />
```

```
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/img" />

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

4. MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

5. AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SpashScreenApp"
        tools:targetApi="31">
```

```
        <activity
            android:name=".MainActivity"
            android:exported="false" />
        <activity
            android:name=".spashActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Program 9:Read/Write the Local data

activity_main.xml Code:

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

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

        <EditText
            android:id="@+id/editName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="text"
            android:text="Enter Name" />
```

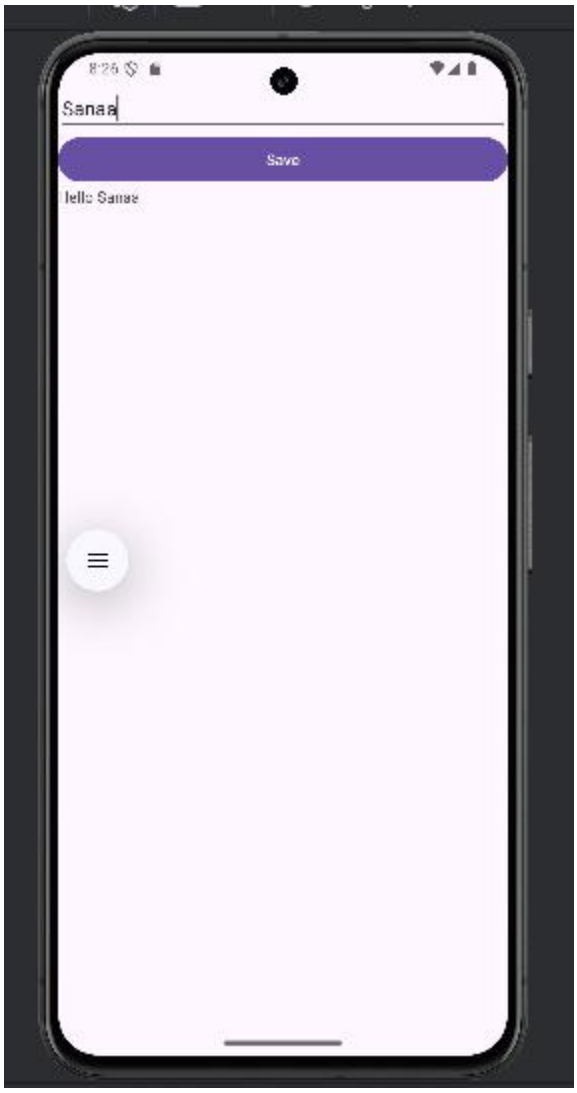
```
    <Button
        android:id="@+id/btnSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save Preferences" />

    <TextView
        android:id="@+id/txtResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello" />

  </LinearLayout>

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

MainActivity.java:

```
package com.example.userprefapp;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

```
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets)
-> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
        EditText editName = findViewById(R.id.editName);
        Button btnSave = findViewById(R.id.btnSave);
        TextView txtResult = findViewById(R.id.txtResult);
        SharedPreferences prefs = getPreferences(MODE_PRIVATE);
        txtResult.setText("Hello " + prefs.getString("name", ""));
        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                prefs.edit().putString("name", editName.getText().toString()).apply();
                txtResult.setText("Hello " + editName.getText().toString());
            }
        });

    }
}
```

Output:


8:26
Sanaa
Save
hello Sanaa

Program 10:Create / Read / Write data with database (SQLite).

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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

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

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

        <Button android:id="@+id/save"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Save" />

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

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

MainActivity.java

```
package com.example.sqldbapp;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    TextView output;
    SQLiteDatabase db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets)
-> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

       EditText input = findViewById(R.id.input);
       output= findViewById(R.id.output);
        Button save = findViewById(R.id.save);

        // Create/open database and table
```

```
        db = openOrCreateDatabase("MyDB", MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(name TEXT)");

        // Save name to database
        save.setOnClickListener(v -> {
            String name = input.getText().toString();
            db.execSQL("INSERT INTO student VALUES('" + name + "')");
            showNames();
        });

        showNames(); // Show existing names
    }

    void showNames() {
        Cursor c = db.rawQuery("SELECT * FROM student", null);
        StringBuilder names = new StringBuilder();
        while (c.moveToNext()) names.append(c.getString(0)).append("\n");
        output.setText(names.toString());

    }
}
```

Output:

Program11:Create an application to send SMS and receive SMS

Permissions Required

Add the following permissions inside the AndroidManifest.xml file before <application:

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"
/>
```

Layout File: activity_main.xml

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

```
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp">

    <EditText
        android:id="@+id/e1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Mobile Number"
        android:inputType="phone" />

    <EditText
        android:id="@+id/e2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Message"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send SMS" />
</LinearLayout>
```

Java Code: MainActivity.java

```
package com.example.smsapp;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
```

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

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets)
-> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        EditText e1 = findViewById(R.id.e1);
        EditText e2 = findViewById(R.id.e2);
        Button btn = findViewById(R.id.btn);

        btn.setOnClickListener(view -> {
            String mobile_no = e1.getText().toString();
            String message = e2.getText().toString();

            Intent my_intent = new Intent(getApplicationContext(), MainActivity.class);
            PendingIntent my_pen = PendingIntent.getActivity(getApplicationContext(), 0,
my_intent, PendingIntent.FLAG_IMMUTABLE);

            SmsManager mymessage = SmsManager.getDefault();
            mymessage.sendTextMessage(mobile_no, null, message, my_pen, null);

            Toast.makeText(MainActivity.this, "Message sent",
Toast.LENGTH_SHORT).show();
        });
    }
}
```

Steps to Test the App on a Physical Device

Enable USB Debugging on Android Phone:

1. Open Settings → About Phone.
2. Tap Build Number 7 times to enable Developer Options.
3. Go back to Settings → Developer Options.
4. Enable USB Debugging.

Connect to Android Studio:

1. Connect your phone to the computer using USB.
2. Go to **Settings** > **Apps**
3. Scroll to find **SMSApp**
4. **In SMSApp->permissions->click on SMS->Select Allow**

Output

• User enters the mobile number and message.
• On clicking Send SMS, the message is sent.
• A toast message "Message sent" is displayed.

Program12:Create an application to send an email

Layout File (activity_main.xml):

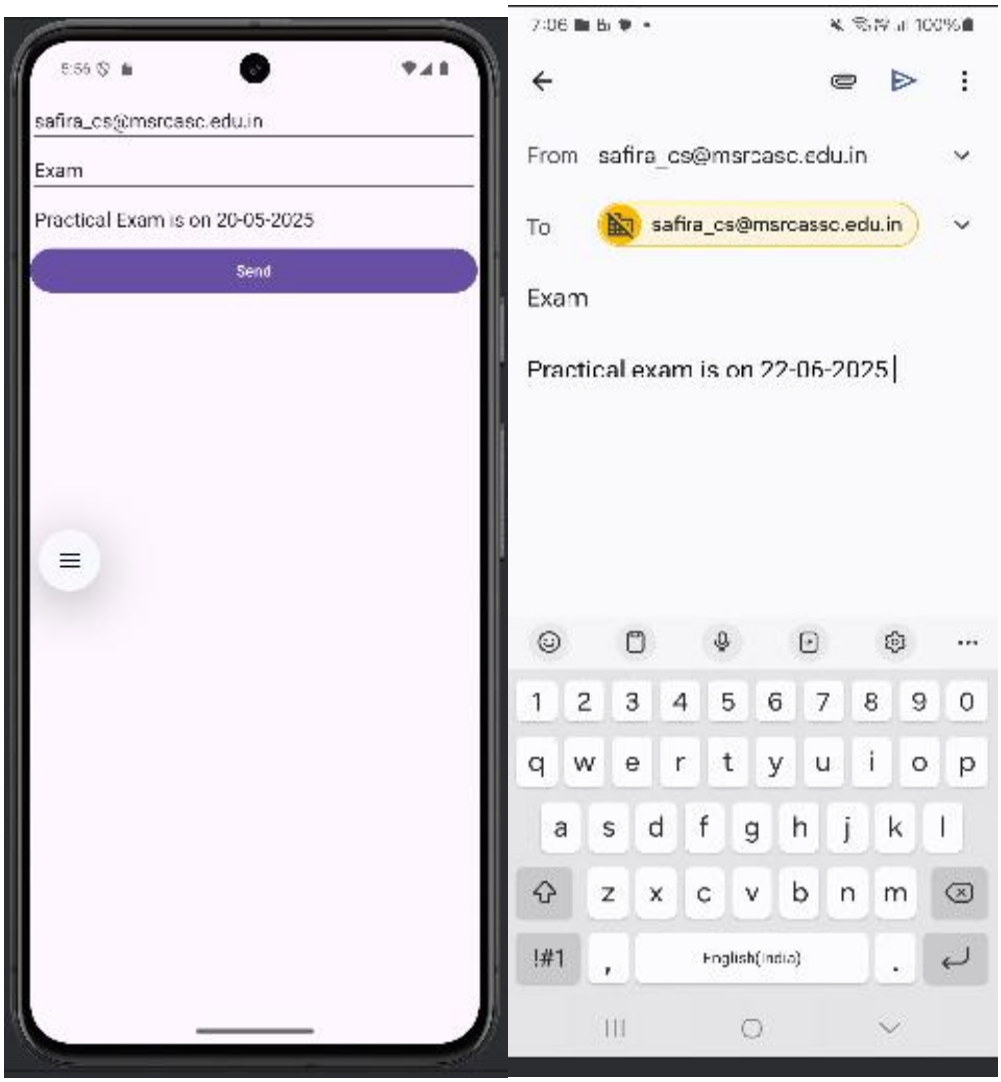
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"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

    <LinearLayout
        android:layout_width="409dp"
        android:layout_height="729dp"
        android:orientation="vertical"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="1dp">

        <EditText
            android:id="@+id/et1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="text"
            android:text="Email Address" />

        <EditText
            android:id="@+id/et2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="text"
            android:text="Subject" />

        <EditText
            android:id="@+id/et3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="text"
```

```
        android:text="Message" />

    <Button
        android:id="@+id/btn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send" />
  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java File (MainActivity.java):

```
package com.example.emailapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

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

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
      Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
      v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
      return insets;
    });

    EditText et1 = findViewById(R.id.et1);
    EditText et2 = findViewById(R.id.et2);
    EditText et3 = findViewById(R.id.et3);
```

```
        Button btn = findViewById(R.id.btn);

        btn.setOnClickListener(view -> {
            String emailAddress = et1.getText().toString();
            String subject = et2.getText().toString();
            String message = et3.getText().toString();

            Intent intent = new Intent(Intent.ACTION_SEND);
            intent.setType("text/plain");
            intent.putExtra(Intent.EXTRA_EMAIL, new String[]{emailAddress});
            intent.putExtra(Intent.EXTRA_SUBJECT, subject);
            intent.putExtra(Intent.EXTRA_TEXT, message);
            startActivity(Intent.createChooser(intent, "Choose an Email app"));
        });
    }
}
```



Program 13:Display Map based on the current/given Location

Add Required Permissions in -AndroidManifest.xml

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

Update Dependencies in-Build.gradle.kts(module:app)

```
dependencies {

implementation("com.google.android.gms:play-services-location:21.3.0")
}
```

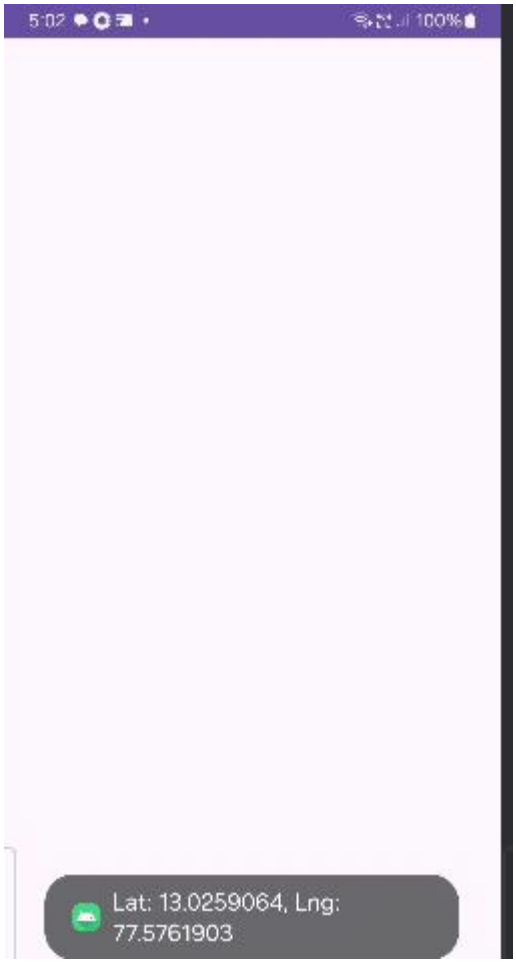
Design Layout-In res/layout/activity_main.xml:

```
<?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:gravity="center"
    android:padding="16dp">
</LinearLayout>
```

MainActivity.java

```
package com.example.mapapp;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.Priority;

public class MainActivity extends AppCompatActivity {

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

        FusedLocationProviderClient client =
LocationServices.getFusedLocationProviderClient(this);

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
```

```
PackageManager.PERMISSION_GRANTED) {
        client.getCurrentLocation(Priority.PRIORITY_HIGH_ACCURACY, null)
                .addOnSuccessListener(location -> {
                    if (location != null) {
                        double lat = location.getLatitude();
                        double lng = location.getLongitude();
                        Toast.makeText(this, "Lat: " + lat + ", Lng: " + lng,
Toast.LENGTH_LONG).show();
                    } else {
                        Toast.makeText(this, "Location is null", Toast.LENGTH_SHORT).show();
                    }
                });
      } else {
          ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
      }
   }
}
```


5:02
100%
Lat: 13.0259064, Lng:
77.5761903