

1. First Launch Setup
 - o Start Android Studio.
 - o It will install the SDK, Emulator, and build tools.
 - o Choose Standard setup (recommended).
 - o Wait until setup completes.
 - a. Create and run a basic “Hello World” Android app.

App Name to use: “whello”

Code: MainActivity.java

```
package com.example.whello;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {

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

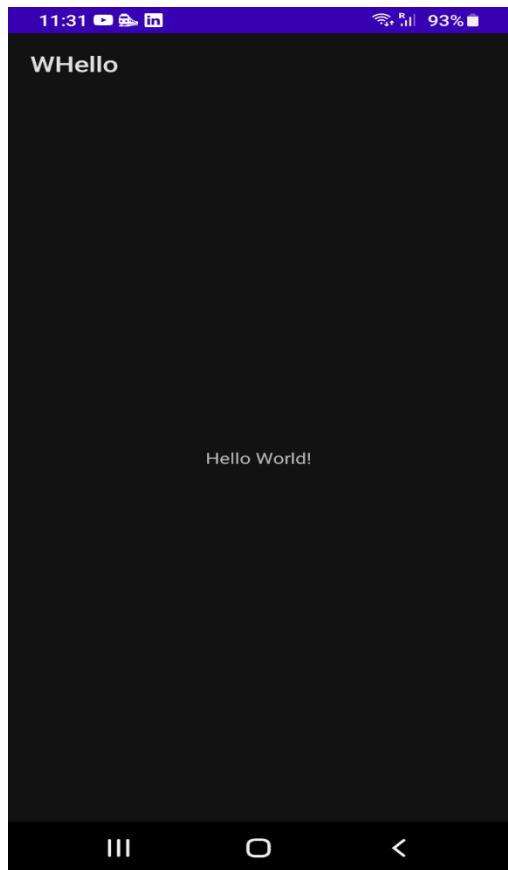
1. activity_main.xml

```
<?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>
```

Output Screen:



2. Explore Android Project Structure.

1. AndroidManifest.xml

The Android Manifest file is the core configuration file for every Android application. It acts as a bridge between the Android system and the app. It defines the overall structure and behavior of the app.

Functions:

- Declares app components such as Activities, Services, Broadcast Receivers, and Content Providers.
- Specifies the app's package name, icon, label (app name), and theme.
- Declares required permissions (like Internet, Camera, Storage, etc.).
- Defines the launcher activity — the first screen of the app that runs when the app starts.
- Sets minimum and target SDK versions to ensure device compatibility.

Code:

```
<application
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

2. Java Folder

This folder contains all the Java (or Kotlin) source files that define the functionality and logic of the application. It's the “brain” of the app.

Key components:

- MainActivity.java – The main entry point that controls what happens on the screen.
- Other classes, adapters, and helper files for managing app logic and events.
- May also contain test folders (androidTest and test) used for unit testing and instrumented testing.

3. Res (Resources) Folder

The res folder stores all non-code resources used in the application such as layouts, images, and strings. This helps separate the design from the logic.

Subfolders:

- layout/ – Contains XML files that define the UI layout for each activity or fragment. Example: activity_main.xml.
- drawable/ – Stores image resources (PNG, JPG, XML drawables like shapes or gradients).
- values/ – Holds important XML files for reusable values such as colors (colors.xml), strings (strings.xml), and styles (styles.xml).
- mipmap/ – Stores launcher icons in different resolutions (hdpi, mdpi, xhdpi, etc.) for various screen densities.
- menu/ – Contains XML files for defining menus and navigation options.
- raw/ – Used to store raw files such as audio, video, or other media files.

4. Gradle Files

Gradle is the build automation system used in Android Studio to compile, build, and package the app efficiently.

There are two main Gradle files:

- Project-level build.gradle:
 - Manages settings common to all modules in the project.

- Includes repositories (like Google and Maven) and plugin definitions.
- Module-level build.gradle:
 - Contains app-specific settings such as compileSdkVersion, minSdkVersion, and dependencies.
 - Defines build types (debug/release) and versioning of the app.

Gradle automates dependency management, testing, and APK generation.

- Demonstrate the use of AVD (Android Virtual Device) and ADB (Android Debug Bridge).

Feature	AVD (Android Virtual Device)	ADB (Android Debug Bridge)
Definition	AVD is an emulator that simulates an Android device on your computer.	ADB is a command-line tool used to communicate with Android devices or emulators.
Purpose	Used to run and test Android applications virtually without a physical device.	Used to debug, install, or control apps and devices via commands.
Dependency	Running your app on a virtual phone or tablet.	Works through Android SDK's platform tools.
Device Type	Virtual device (simulator).	Can connect to both physical and virtual devices.
Example	Running your app on a virtual phone or tablet.	Typing adb install app.apk to install the app on a device.

3. Create an App Using Different Views.

- Use TextView, Button, and ImageView components.

App Name to use: "ivbutton"

Code:MainActivity.java

```
package com.example.ivbutton;

import android.os.Bundle;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private boolean showingDefaultState = true;

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

        TextView greetingTextView = findViewById(R.id.textViewGreeting);
        ImageView iconImageView = findViewById(R.id.imageViewIcon);
        Button changeButton = findViewById(R.id.buttonChange);

        changeButton.setOnClickListener(view -> {
            if (showingDefaultState) {
                greetingTextView.setText(R.string.message_clicked);
                iconImageView.setImageResource(R.mipmap.ic_launcher_round);
                changeButton.setText(R.string.button_reset);
            } else {
                greetingTextView.setText(R.string.message_default);
                iconImageView.setImageResource(R.mipmap.ic_launcher);
                changeButton.setText(R.string.button_change);
            }
            showingDefaultState = !showingDefaultState;
        });
    }
}
```

2.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/textViewGreeting"
        android:layout_width="wrap_content"
```

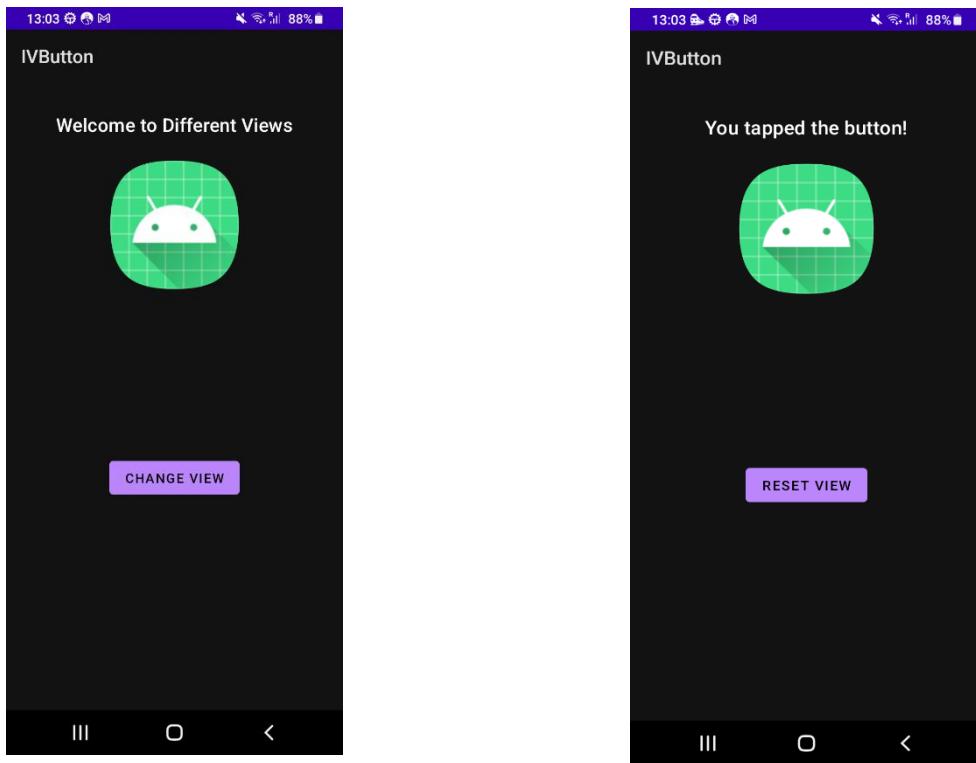
```
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/message_default"
        android:textAppearance="?attr/textAppearanceHeadline6"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageViewIcon"
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:layout_marginTop="24dp"
        android:contentDescription="@string/content_description_launcher"
        android:src="@mipmap/ic_launcher"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textViewGreeting" />

    <Button
        android:id="@+id/buttonChange"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="48dp"
        android:text="@string/button_change"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageViewIcon" />

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

Output Screen:



- o Demonstrate Android Activity Lifecycle using Log messages.

The **Android Activity Lifecycle** represents the different states an activity (screen) goes through from when it is created until it is destroyed. Understanding the lifecycle helps developers manage how the app behaves when the user opens, closes, or switches between screens.

Method	Description
onCreate()	Called when the activity is first created. Used to initialize UI components.
onStart()	Called when the activity becomes visible to the user.
onResume()	Called when the activity starts interacting with the user (app is in the foreground)
onPause()	Called when another activity comes into focus. Used to pause ongoing tasks.
onStop()	Called when the activity is no longer visible to the user.
onRestart()	Called when the activity is restarted after being stopped.
onDestroy()	Called before the activity is destroyed. Used to release resources.

4. Implement Intent and Bundle Passing Between Activities
App Name to use: "bpassing"

Code:MainActivity.java

```
package com.example.bpassing;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    private EditText etName, etAge, etEmail, etMessage;
    private Button btnSendData;

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

        // Initialize views
        etName = findViewById(R.id.etName);
        etAge = findViewById(R.id.etAge);
        etEmail = findViewById(R.id.etEmail);
        etMessage = findViewById(R.id.etMessage);
        btnSendData = findViewById(R.id.btnSendData);

        // Set click listener for button
        btnSendData.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendDataToSecondActivity();
            }
        });
    }

    private void sendDataToSecondActivity() {
        // Get input values with null safety
        String name = etName != null ? etName.getText().toString().trim() : "";
        String ageStr = etAge != null ? etAge.getText().toString().trim() : "";
        String email = etEmail != null ? etEmail.getText().toString().trim() : "";
        String message = etMessage != null ? etMessage.getText().toString().trim() : "";

        // Validate inputs
        if (name.isEmpty() || ageStr.isEmpty() || email.isEmpty()) {
            Toast.makeText(this, "Please fill in all required fields", Toast.LENGTH_SHORT).show();
        }
    }
}
```

```

        return;
    }

int age = Integer.parseInt(ageStr);

// Method 1: Passing data via Intent extras (direct method)
Intent intent = new Intent(MainActivity.this, SecondActivity.class);

// Put data directly into Intent extras
intent.putExtra("name", name);
intent.putExtra("age", age);
intent.putExtra("email", email);

// Method 2: Passing data via Bundle
Bundle bundle = new Bundle();
// Only put message if it's not empty
if (message != null && !message.isEmpty()) {
    bundle.putString("message", message);
} else {
    bundle.putString("message", "");
}
bundle.putBoolean("isStudent", true);
bundle.putDouble("score", 95.5);

// Attach bundle to Intent
intent.putExtras(bundle);

// Start SecondActivity
startActivity(intent);
}
}

```

2.SecondActivity.java

```

package com.example.bpassing;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    private TextView tvName, tvAge, tvEmail, tvMessage, tvBundleData;

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

        // Initialize views
        tvName = findViewById(R.id.tvName);
        tvAge = findViewById(R.id.tvAge);
        tvEmail = findViewById(R.id.tvEmail);

```

```

tvMessage = findViewById(R.id.tvMessage);
tvBundleData = findViewById(R.id.tvBundleData);

// Method 1: Receiving data via Intent extras (direct method)
Intent intent = getIntent();
if (intent != null) {
    // Get data passed directly via Intent extras
    String name = intent.getStringExtra("name");
    int age = intent.getIntExtra("age", 0);
    String email = intent.getStringExtra("email");

    // Display the data with null checks
    if (name != null && !name.isEmpty()) {
        tvName.setText("Name: " + name);
    } else {
        tvName.setText("Name: Not provided");
    }

    tvAge.setText("Age: " + String.valueOf(age));

    if (email != null && !email.isEmpty()) {
        tvEmail.setText("Email: " + email);
    } else {
        tvEmail.setText("Email: Not provided");
    }
}

// Method 2: Receiving data via Bundle
Bundle bundle = intent != null ? intent.getExtras() : null;
if (bundle != null) {
    String message = bundle.getString("message");
    boolean isStudent = bundle.getBoolean("isStudent", false);
    double score = bundle.getDouble("score", 0.0);

    if (message != null && !message.isEmpty()) {
        tvMessage.setText("Message: " + message);
    } else {
        tvMessage.setText("Message: Not provided");
    }

    // Display bundle data
    String studentText = isStudent ? "Yes" : "No";
    String scoreText = String.valueOf(score);
    String bundleInfo = "Student: " + studentText + "\n" + "Score: " + scoreText;
    tvBundleData.setText(bundleInfo);
} else {
    tvMessage.setText("Message: Not provided");
    tvBundleData.setText("Bundle data: Not provided");
}
}

3.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"
android:padding="16dp"
tools:context=".MainActivity"

<TextView
    android:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Intent & Bundle Passing Demo"
    android:textSize="24sp"
    android:textStyle="bold"
    android:layout_marginTop="32dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/tvSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enter your information:"
    android:textSize="16sp"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTitle" />

<EditText
    android:id="@+id/etName"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter Name"
    android:inputType="textPersonName"
    android:layout_marginTop="24dp"
    android:padding="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvSubtitle" />

<EditText
    android:id="@+id/etAge"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter Age"
    android:inputType="number"
    android:layout_marginTop="16dp"
    android:padding="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etName" />

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

```

        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:layout_marginTop="16dp"
        android:padding="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etAge" />

<EditText
    android:id="@+id/etMessage"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter Message (Optional)"
    android:inputType="textMultiLine"
    android:minLines="3"
    android:layout_marginTop="16dp"
    android:padding="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etEmail" />

<Button
    android:id="@+id	btnSendData"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Send Data to Second Activity"
    android:textSize="16sp"
    android:layout_marginTop="32dp"
    android:padding="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etMessage" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

4.activity_second.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"
    android:padding="16dp"
    tools:context=".SecondActivity">

<TextView
    android:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Received Data"
    android:textSize="24sp"
    android:textStyle="bold" />

```

```
        android:layout_marginTop="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/tvSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Data passed via Intent Extras:"
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_marginTop="32dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTitle" />

<TextView
    android:id="@+id/tvName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Name: "
    android:textSize="16sp"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvSubtitle" />

<TextView
    android:id="@+id/tvAge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Age: "
    android:textSize="16sp"
    android:layout_marginTop="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvName" />

<TextView
    android:id="@+id/tvEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Email: "
    android:textSize="16sp"
    android:layout_marginTop="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvAge" />

<TextView
    android:id="@+id/tvMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Message: "
    android:textSize="16sp"
```

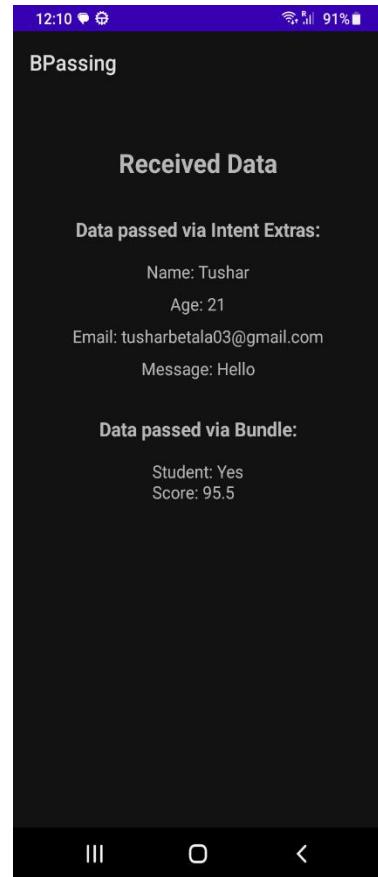
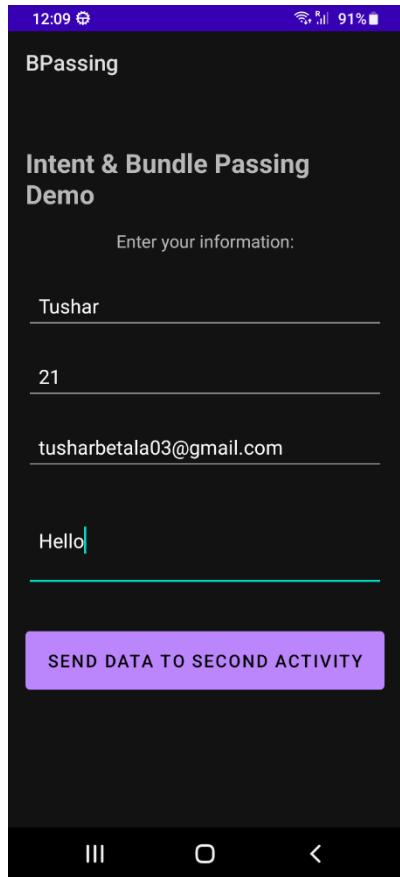
```
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvEmail" />

<TextView
    android:id="@+id/tvBundleTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Data passed via Bundle:"
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_marginTop="32dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvMessage" />

<TextView
    android:id="@+id/tvBundleData"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="16sp"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvBundleTitle" />

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

Output Screen:



5. Create a BMI Calculator Application.

App Name to use: "calcuatebmi"

Code:MainActivity.java

```
package com.example.calcuuatebmi;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText editHeight;
    private EditText editWeight;
    private Button btnCalculate;
    private TextView txtResult;
    private TextView txtCategory;

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

        editHeight = findViewById(R.id.editHeight);
        editWeight = findViewById(R.id.editWeight);
        btnCalculate = findViewById(R.id.btnCalculate);
        txtResult = findViewById(R.id.txtResult);
        txtCategory = findViewById(R.id.txtCategory);

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

    private void calculateAndDisplayBmi() {
        String heightInput = editHeight.getText() != null ? editHeight.getText().toString().trim() : "";
        String weightInput = editWeight.getText() != null ? editWeight.getText().toString().trim() : "";

        boolean hasError = false;

        if (TextUtils.isEmpty(heightInput)) {
            editHeight.setError(getString(R.string.error_required));
            hasError = true;
        }
        if (TextUtils.isEmpty(weightInput)) {
            editWeight.setError(getString(R.string.error_required));
            hasError = true;
        }
    }
}
```

```

        }
        if (hasError) {
            return;
        }

        double heightCm;
        double weightKg;
        try {
            heightCm = Double.parseDouble(heightInput);
        } catch (NumberFormatException ex) {
            editHeight.setError(getString(R.string.error_invalid_number));
            return;
        }
        try {
            weightKg = Double.parseDouble(weightInput);
        } catch (NumberFormatException ex) {
            editWeight.setError(getString(R.string.error_invalid_number));
            return;
        }

        if (heightCm <= 0) {
            editHeight.setError(getString(R.string.error_value_out_of_range));
            return;
        }
        if (weightKg <= 0) {
            editWeight.setError(getString(R.string.error_value_out_of_range));
            return;
        }

        double heightM = heightCm / 100.0;
        double bmi = weightKg / (heightM * heightM);

        String bmiText = getString(R.string.bmi_result_prefix) + " " + String.format("%.2f", bmi);
        txtResult.setText(bmiText);
        txtCategory.setText(getBmiCategoryText(bmi));
    }

    private String getBmiCategoryText(double bmi) {
        if (bmi < 18.5) {
            return getString(R.string.bmi_category_underweight);
        } else if (bmi < 25.0) {
            return getString(R.string.bmi_category_normal);
        } else if (bmi < 30.0) {
            return getString(R.string.bmi_category_overweight);
        } else {
            return getString(R.string.bmi_category_obese);
        }
    }
}

2. 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/txtTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/title_bmi_calculator"
        android:textStyle="bold"
        android:textSize="20sp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="24dp" />

    <EditText
        android:id="@+id/editHeight"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:hint="@string/hint_enter_height"
        android:ems="10"
        app:layout_constraintTop_toBottomOf="@+id/txtTitle"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="24dp" />

    <EditText
        android:id="@+id/editWeight"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:hint="@string/hint_enter_weight"
        android:ems="10"
        app:layout_constraintTop_toBottomOf="@+id/editHeight"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id	btnCalculate"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="@string/action_calculate"
        app:layout_constraintTop_toBottomOf="@+id/editWeight"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="24dp" />

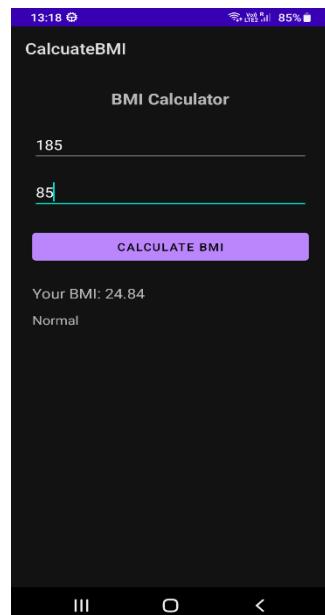
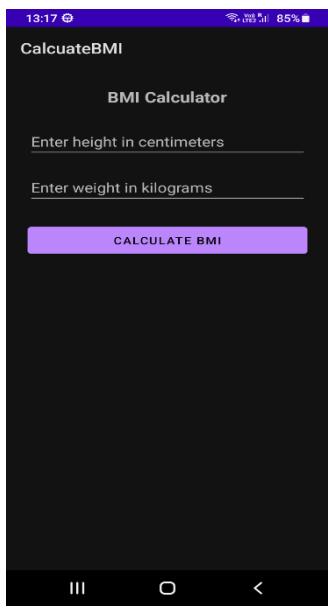
    <TextView
```

```
        android:id="@+id/txtResult"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@+id/btnCalculate"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="24dp" />

    <TextView
        android:id="@+id/txtCategory"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="16sp"
        app:layout_constraintTop_toBottomOf="@+id/txtResult"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginTop="12dp" />

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

Output Screen:



6. Create a Splash Screen for Your Application.

App Name to use: "splashes"

In add **AndroidManifest.xml**, inside <application>.....</application>:

```
<activity
    android:name=".SplashActivity"
    android:exported="true"
    android:theme="@style/Theme.Splash">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".MainActivity" android:exported="true"/>
```

Code:MainActivity.java

```
package com.example.splashes;

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

public class MainActivity extends AppCompatActivity {

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

2. SplashActivity.java

```
package com.example.splashes;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;

public class SplashActivity extends Activity {

    private static final long SPLASH_DELAY_MS = 800; // short, responsive delay

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // No setContentView: we rely on the splash theme window background

        new Handler(Looper.getMainLooper()).postDelayed(() -> {
            Intent intent = new Intent(SplashActivity.this, MainActivity.class);

```

```

        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        // finish so user can't return to splash
        finish();
    }, SPLASH_DELAY_MS);
}
}

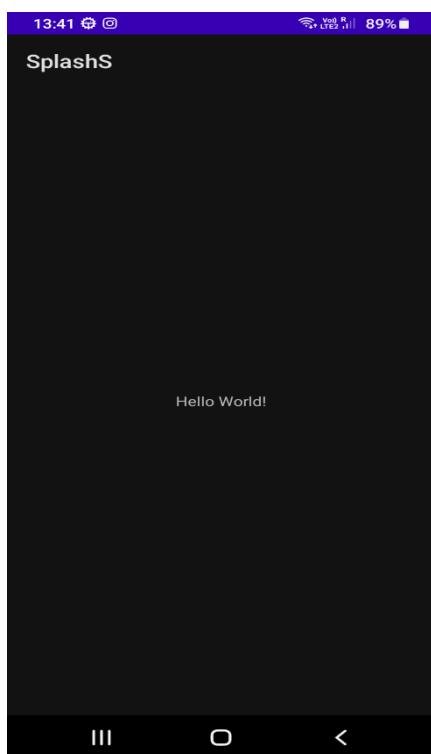
3.activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"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

```

Output Screen:



7. Display a Toast Message in an Android App.

App Name to use: "toastm"

Code:Main Activity.java

```
package com.example.toastm;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Toast.makeText(this, "MainActivity: Home screen is ready", Toast.LENGTH_SHORT).show();
    }
}
```

2.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>
```

Output Screen:



8. Create a Message Sending App in Android.

App Name to use: "messages"

add this in **AndroidManifest.xml**:

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

Code:MainActivity.java

```
package com.example.messages;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private static final int REQ_SEND_SMS = 1001;

    private EditText inputPhone;
    private EditText inputMessage;
    private Button buttonSend;

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

        inputPhone = findViewById(R.id.input_phone);
        inputMessage = findViewById(R.id.input_message);
        buttonSend = findViewById(R.id.button_send);

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

    private void attemptSendSms() {
        final String phone = inputPhone.getText() != null ? inputPhone.getText().toString().trim() : "";
        final String message = inputMessage.getText() != null ? inputMessage.getText().toString().trim() : "";

        if (TextUtils.isEmpty(phone)) {
            Toast.makeText(this, getString(R.string.error_empty_phone), Toast.LENGTH_SHORT).show();
        }
    }
}
```

```

        return;
    }

    if (TextUtils.isEmpty(message)) {
        Toast.makeText(this, getString(R.string.error_empty_message), Toast.LENGTH_SHORT).show();
        return;
    }

    if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS}, REQ_SEND_SMS);
        return;
    }

    sendSms(phone, message);
}

private void sendSms(String phone, String message) {
    try {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(phone, null, message, null, null);
        Toast.makeText(this, getString(R.string.sms_sent), Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(this, getString(R.string.sms_failed), Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQ_SEND_SMS) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            attemptSendSms();
        } else {
            Toast.makeText(this, getString(R.string.permission_required), Toast.LENGTH_SHORT).show();
        }
    }
}

2.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/label_phone"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="@string/label_phone"
        app:layout_constraintTop_toTopOf="parent"

```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="24dp" />

<EditText
    android:id="@+id/input_phone"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="phone"
    android:hint="@string/hint_phone"
    app:layout_constraintTop_toBottomOf="@+id/label_phone"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="8dp" />

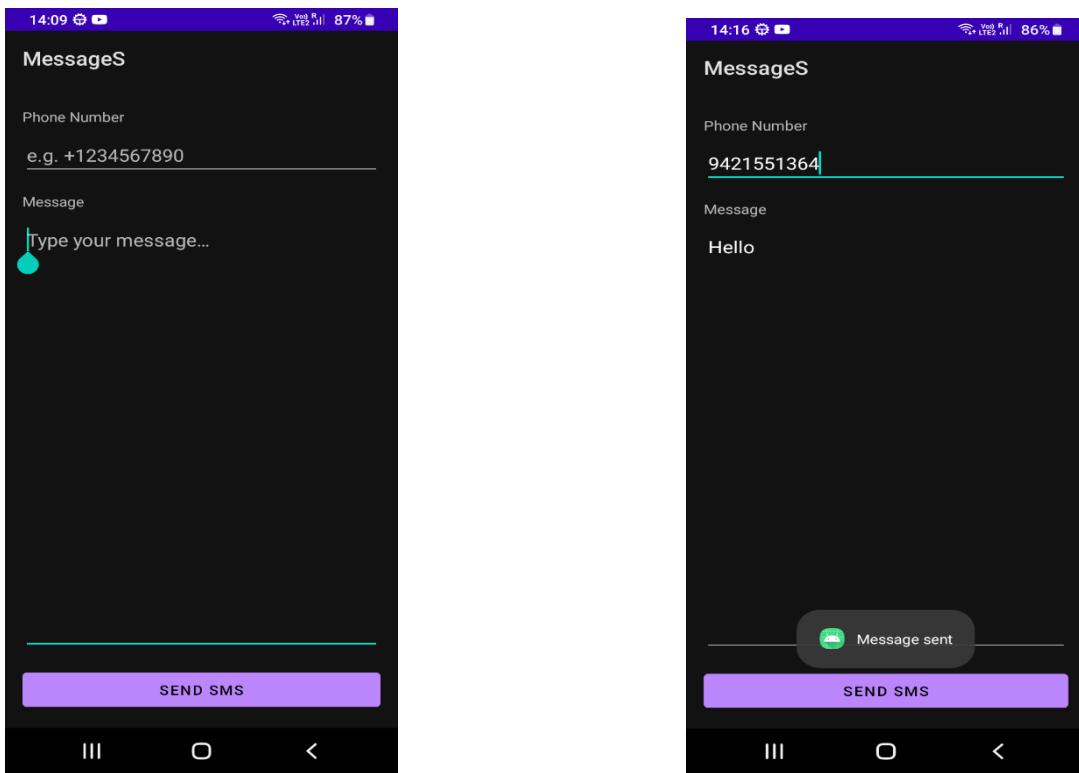
<TextView
    android:id="@+id/label_message"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/label_message"
    app:layout_constraintTop_toBottomOf="@+id/input_phone"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="16dp" />

<EditText
    android:id="@+id/input_message"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:hint="@string/hint_message"
    android:inputType="textMultiLine"
    android:gravity="top|start"
    android:minLines="5"
    app:layout_constraintTop_toBottomOf="@+id/label_message"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/button_send"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="16dp" />

<Button
    android:id="@+id/button_send"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/action_send"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
    android:layout_margin="16dp" />  
  </androidx.constraintlayout.widget.ConstraintLayout>
```

Output Screen:



9. Implement ListView and FrameLayout in an Android App.

App Name to use: "listframe"

Code:MainActivity.java

```
package com.example.listframe;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {

    private ListView listView;

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

        listView = findViewById(R.id.listViewItems);

        String[] items = new String[]{
            "Apple", "Banana", "Cherry", "Durian", "Elderberry",
            "Fig", "Grape", "Honeydew", "Indian Fig", "Jackfruit"
        };

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
            MainActivity.this,
            android.R.layout.simple_list_item_1,
            java.util.Arrays.asList(items)
        );
        listView.setAdapter(adapter);

        // Show default detail on first launch
        if (savedInstanceState == null) {
            replaceDetail(DetailFragment.newInstance("Select an item from the list"));
        }

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, android.view.View view, int position, long id) {
                String selected = (String) parent.getItemAtPosition(position);
                replaceDetail(DetailFragment.newInstance("You selected: " + selected));
            }
        });
    }

    private void replaceDetail(Fragment fragment) {
        FragmentTransaction tx = getSupportFragmentManager().beginTransaction();
        tx.replace(R.id.frameContainer, fragment);
        tx.commit();
    }
}
```

```

        }
    }

2.DetailFragment.java
package com.example.listframe;

import android.os.Bundle;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class DetailFragment extends Fragment {

    private static final String ARG_TEXT = "arg_text";

    public static DetailFragment newInstance(String text) {
        DetailFragment fragment = new DetailFragment();
        Bundle args = new Bundle();
        args.putString(ARG_TEXT, text);
        fragment.setArguments(args);
        return fragment;
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable
    Bundle savedInstanceState) {
        TextView textView = new TextView(requireContext());
        textView.setLayoutParams(new ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT));
        int padding = (int) (16 * getResources().getDisplayMetrics().density);
        textView.setPadding(padding, padding, padding, padding);
        textView.setTextSize(18);
        textView.setTextColor(Color.BLACK);

        Bundle args = getArguments();
        if (args != null) {
            String text = args.getString(ARG_TEXT, "");
            textView.setText(text);
        }
        return textView;
    }
}

```

3.activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

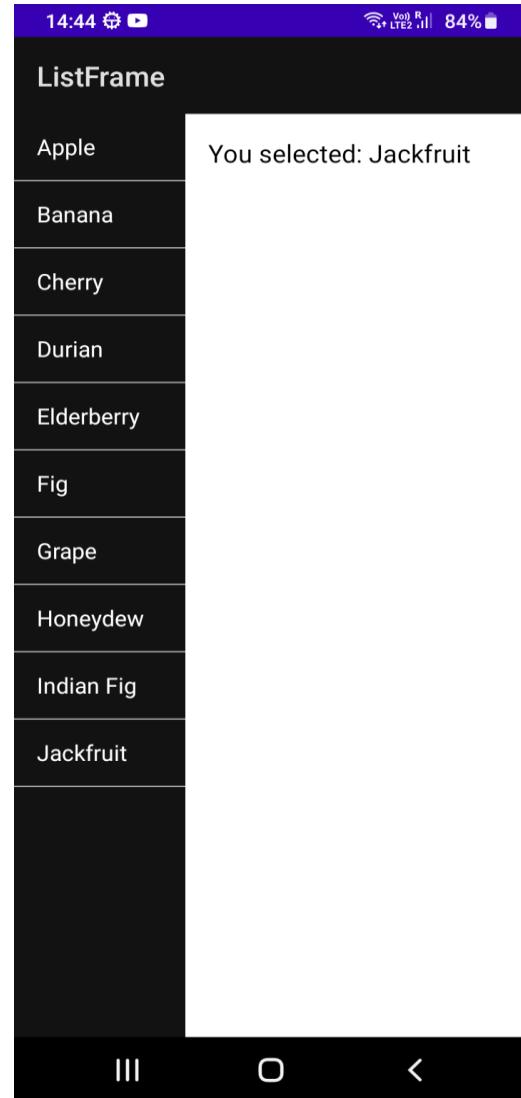
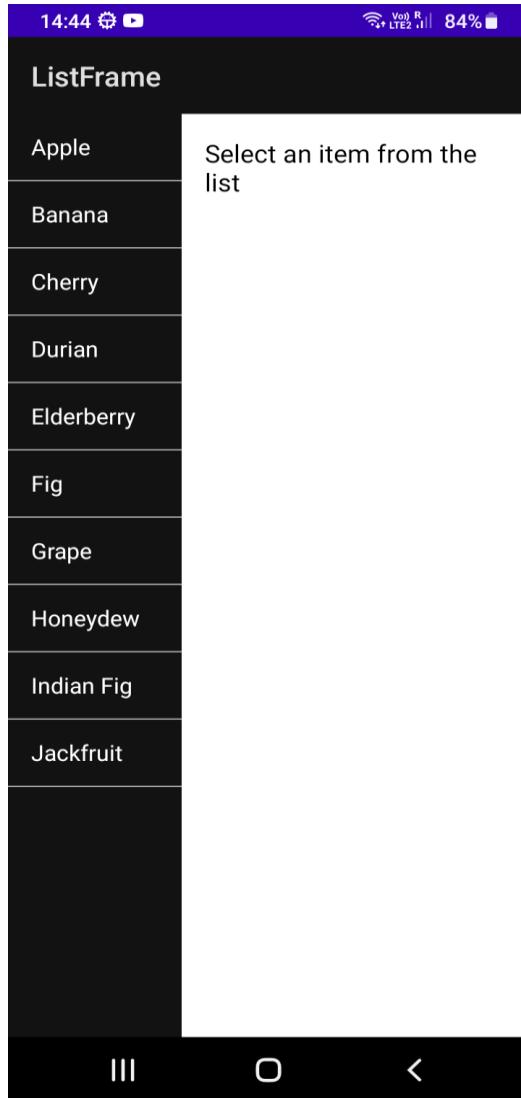
```
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/listViewItems"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:divider="@android:color/darker_gray"
        android:dividerHeight="1dp" />

    <FrameLayout
        android:id="@+id/frameContainer"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:background="@android:color/white" />

</LinearLayout>
```

Output Screen:



10. Implement Bottom Navigation and Navigation Drawer in an App.

App Name to use: "navigationb"

Code:MainActivity.java

```
package com.example.navigationb;

import android.os.Bundle;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.fragment.app.Fragment;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.navigation.NavigationView;

public class MainActivity extends AppCompatActivity {

    private DrawerLayout drawerLayout;
    private NavigationView navigationView;
    private BottomNavigationView bottomNavigationView;

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

            Toolbar toolbar = findViewById(R.id.toolbar);
            if (toolbar != null) {
                setSupportActionBar(toolbar);
            }

            drawerLayout = findViewById(R.id.drawer_layout);
            navigationView = findViewById(R.id.navigation_view);
            bottomNavigationView = findViewById(R.id.bottom_navigation);

            if (drawerLayout != null && toolbar != null) {
                ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                    this, drawerLayout, toolbar,
                    R.string.navigation_drawer_open, R.string.navigation_drawer_close);
                drawerLayout.addDrawerListener(toggle);
                toggle.syncState();
            }

            if (navigationView != null) {
                navigationView.setNavigationItemSelected(item -> {
                    handleDrawerSelection(item);
                    return true;
                });
            }
        }
    }
}
```

```

        if (bottomNavigationView != null) {
            bottomNavigationView.setOnItemSelectedListener(item -> {
                handleBottomSelection(item);
                return true;
            });
        }

        if (savedInstanceState == null) {
            switchToFragment(new HomeFragment());
            if (bottomNavigationView != null) {
                bottomNavigationView.setSelectedItemId(R.id.navigation_home);
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(this, "Error: " + e.getMessage(), Toast.LENGTH_LONG).show();
    }
}

private void handleBottomSelection(@NonNull MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.navigation_home) {
        switchToFragment(new HomeFragment());
    } else if (id == R.id.navigation_dashboard) {
        switchToFragment(new DashboardFragment());
    } else if (id == R.id.navigation_notifications) {
        switchToFragment(new NotificationsFragment());
    }
}

private void handleDrawerSelection(@NonNull MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.drawer_profile) {
        Toast.makeText(this, "Profile clicked", Toast.LENGTH_SHORT).show();
    } else if (id == R.id.drawer_settings) {
        Toast.makeText(this, "Settings clicked", Toast.LENGTH_SHORT).show();
    } else if (id == R.id.drawer_help) {
        Toast.makeText(this, "Help clicked", Toast.LENGTH_SHORT).show();
    }
    drawerLayout.closeDrawer(GravityCompat.START);
}

private void switchToFragment(Fragment fragment) {
    if (fragment != null && findViewById(R.id.nav_host_container) != null) {
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.nav_host_container, fragment)
            .commit();
    }
}
}

2. DashboardFragment.java
package com.example.navigationb;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class DashboardFragment extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable
    Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_dashboard, container, false);
    }
}
```

3.HomeFragment.java

```
package com.example.navigationb;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
```

```
public class HomeFragment extends Fragment {
```

```
    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable
    Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_home, container, false);
    }
}
```

4.NotificationFragment.java

```
package com.example.navigationb;
```

```
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
```

```
public class NotificationsFragment extends Fragment {
```

```
@Nullable  
@Override  
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable  
Bundle savedInstanceState) {  
    return inflater.inflate(R.layout.fragment_notifications, container, false);  
}  
}
```

5.activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.drawerlayout.widget.DrawerLayout 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/drawer_layout"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:openDrawer="start"  
    tools:context=".MainActivity">  
  
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <com.google.android.material.appbar.AppBarLayout  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
  
        <androidx.appcompat.widget.Toolbar  
            android:id="@+id/toolbar"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            app:title="@string/app_name" />  
  
    </com.google.android.material.appbar.AppBarLayout>  
  
<FrameLayout  
    android:id="@+id/nav_host_container"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginBottom="56dp" />  
  
<com.google.android.material.bottomnavigation.BottomNavigationView  
    android:id="@+id/bottom_navigation"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="bottom"  
    android:background="?attr/colorSurface"  
    app:menu="@menu/menu_bottom" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>  
  
<com.google.android.material.navigation.NavigationView  
    android:id="@+id/navigation_view"  
    android:layout_width="280dp"  
    android:layout_height="match_parent"
```

```

        android:layout_gravity="start"
        app:menu="@menu/menu_drawer" />

    </androidx.drawerlayout.widget.DrawerLayout>

6.fragment_dashboard.xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dashboard"
        android:textSize="24sp"
        android:layout_gravity="center" />

</FrameLayout>
7. fragment_home.xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Home"
        android:textSize="24sp"
        android:layout_gravity="center" />

</FrameLayout>
8. fragment_notification.xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notifications"
        android:textSize="24sp"
        android:layout_gravity="center" />

</FrameLayout>

```

Output Screen:

