Experiment 4: BMI Calculator

1. Objective

Develop a Body Mass Index (BMI) calculator that takes weight and height as input and displays the BMI value along with a basic health assessment. This exercise involves more complex UI inputs, arithmetic operations, and conditional logic.

2. Steps to Build the app

1. Design the UI:

   Use two EditText elements for user input: one for weight (in kilograms) and one for height (in meters).

   Add a Button for calculating the BMI.

   Include a TextView to display the result and the health assessment.

```
<?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">

    <EditText
        android:id="@+id/etWeight"
        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_marginStart="32dp"

        android:layout_marginTop="32dp"

        android:layout_marginEnd="32dp"

        android:hint="Weight (kg)"

        android:inputType="numberDecimal"

        app:layout_constraintEnd_toEnd="parent"

        app:layout_constraintStart_toStart="parent"

        app:layout_constraintTop_toTop="parent" />


    <EditText

        android:id="@+id/etHeight"

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_marginTop="16dp"

        android:hint="Height (m)"

        android:inputType="numberDecimal"

        app:layout_constraintEnd_toEnd="parent"

        app:layout_constraintStart_toStart="parent"

        app:layout_constraintTop_toBottom="@+id/etWeight" />


    <Button

        android:id="@+id/btnCalculateBMI"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"

        android:text="Calculate BMI"

        app:layout_constraintEnd_toEnd="parent"

        app:layout_constraintStart_toStart="parent"

        app:layout_constraintTop_toBottom="@+id/etHeight" />


    <TextView

        android:id="@+id/tvBMIResult"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_marginTop="24dp"

        android:text="BMI Result"

        android:textSize="18sp"

        app:layout_constraintEnd_toEnd="parent"

        app:layout_constraintStart_toStart="parent"


app:layout_constraintTop_toBottom="@+id/btnCalculateBMI" />


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

2. Handle User Input:

   Set the inputType of the weight and height EditText elements to numberDecimal to accept decimal values.

3. Set Up the Calculation Logic:

   In MainActivity.java, retrieve the values from the EditText elements when the button is clicked.

Calculate the BMI using the formula: BMI = weight (kg) / (height (m) * height (m)).

Determine the health assessment based on the BMI value (e.g., Underweight, Normal, Overweight).

4. Display the Results:

   Show the calculated BMI and the corresponding health assessment in the TextView.

```
public class MainActivity extends AppCompatActivity {

    EditText etWeight, etHeight;
    Button btnCalculateBMI;
    TextView tvBMIResult;

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

        etWeight = findViewById(R.id.etWeight);
        etHeight = findViewById(R.id.etHeight);
        btnCalculateBMI = findViewById(R.id.btnCalculateBMI);
        tvBMIResult = findViewById(R.id.tvBMIResult);

        btnCalculateBMI.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!etWeight.getText().toString().isEmpty() &&
!etHeight.getText().toString().isEmpty()) {
                    float                weight                 =
Float.parseFloat(etWeight.getText().toString());
                    float                height                 =
Float.parseFloat(etHeight.getText().toString());
                    float bmi = weight / (height * height);
                    String              assessment              =
getHealthAssessment(bmi);
                    tvBMIResult.setText(String.format("BMI:
%.2f (%s)", bmi, assessment));
                }
            }
        });
    }

    private String getHealthAssessment(float bmi) {
        if (bmi < 18.5) {
            return "Underweight";
        } else if (bmi < 25) {
            return "Normal";
        } else if (bmi < 30) {
            return "Overweight";
        } else {
            return "Obese";
        }
    }
}
```

3. Explanation

The EditText elements (etWeight and etHeight) capture the user's weight and height.

When the Button (btnCalculateBMI) is clicked, the app checks if the input fields are not empty, then parses the input values to float.

The BMI is calculated using the given formula and assessed using the getHealthAssessment method, which categorizes the BMI value into common health categories.

The result is formatted and displayed in the TextView (tvBMIResult), providing the user with their BMI value and a simple health assessment.