AADK Task 7 Report

Title: Building Stateful and Calculative Apps using Jetpack Compose

1. Objective

To develop a stateful Android application using Jetpack Compose that combines input fields, slider, switch, validation logic, and real-time calculation updates.

2. Source Code

```
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
import kotlin.math.ceil

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            CalculatorApp()
        }
    }
}

@Composable
fun CalculatorApp() {

    var amount by remember { mutableStateOf("") }
    var percentage by remember { mutableStateOf(10f) }
    var roundUp by remember { mutableStateOf(false) }

    val numericAmount = amount.toDoubleOrNull()
    val result = if (numericAmount != null) {
        var tip = (numericAmount * percentage) / 100
        if (roundUp) tip = ceil(tip)
        tip
    } else 0.0

    Column(modifier = Modifier.padding(16.dp)) {

        OutlinedTextField(
            value = amount,
            onValueChange = { amount = it },
            label = { Text("Enter Amount") },
            modifier = Modifier.fillMaxWidth()
        )

        Spacer(modifier = Modifier.height(16.dp))

        Text("Tip Percentage: ${percentage.toInt()}%")
        Slider(
            value = percentage,
            onValueChange = { percentage = it },
            valueRange = 0f..30f
        )

        Spacer(modifier = Modifier.height(16.dp))

        Row {
            Text("Round Up")
            Switch(
                checked = roundUp,
                onCheckedChange = { roundUp = it }
            )
        }

        Spacer(modifier = Modifier.height(16.dp))

        Text("Calculated Tip: ■%.2f".format(result))
    }
}
```

3. Explanation

- State variables are created using remember and mutableStateOf to store user input and UI values.
- OutlinedTextField captures numeric input safely using toDoubleOrNull().
- Slider dynamically updates the percentage value.
- Switch controls whether the calculated result should be rounded using ceil().
- When any state changes, Jetpack Compose recomposes and updates the UI automatically.
- Calculation logic is structured and directly linked to UI state changes.

4. Conclusion

This task demonstrates structured data flow, reactive state handling, input validation, and real-time UI updates using Jetpack Compose. It builds a strong foundation for developing modern Android applications.