**Lab Exercise 4- Working with Column, Row in Jetpack Compose**

**Objective:**

* Learn how to use Column and Row to arrange elements vertically and horizontally.

**Step 1: Set Up a New Project**

1. **Open Android Studio**.
2. **Create a new project** with **Empty Compose Activity**.
3. Set the **Minimum SDK** to 21 or higher and **Click Finish**.

**Step 2: Create a Simple Layout with Column**

A Column arranges its children vertically, one after the other.

1. Open MainActivity.kt and modify the setContent block to create a simple vertical layout:

package com.example.columnrowlab

import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import com.example.columnrowlab.ui.theme.ColumnRowLabTheme

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

ColumnRowLabTheme {

Surface(modifier = Modifier.fillMaxSize()) {

ColumnExample()

}

}

}

}

}

@Composable

fun ColumnExample() {

Column(

modifier = Modifier

.fillMaxSize()

.padding(16.dp),

verticalArrangement = Arrangement.Center, // Align vertically at the center

horizontalAlignment = Alignment.CenterHorizontally // Align horizontally at the center

) {

Text(text = "First Item")

Spacer(modifier = Modifier.height(16.dp)) // Add space between items

Text(text = "Second Item")

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

Text(text = "Third Item")

}

}

@Preview(showBackground = true)

@Composable

fun ColumnPreview() {

ColumnRowLabTheme {

ColumnExample()

}

}

1. **Explanation**:
   * Column: Arranges the text items vertically.
   * Spacer: Adds space between items to avoid them being too close together.
   * verticalArrangement: Aligns items vertically.
   * horizontalAlignment: Aligns items horizontally.
2. **Run the app** to see the vertically aligned text items.

**Step 3: Create a Simple Layout with Row**

A Row arranges its children horizontally, side by side.

1. Modify the code to create a horizontal layout using Row:

@Composable

fun RowExample() {

Row(

modifier = Modifier

.fillMaxWidth()

.padding(16.dp),

horizontalArrangement = Arrangement.SpaceBetween, // Space items evenly

verticalAlignment = Alignment.CenterVertically // Align items vertically at the center

) {

Text(text = "First Item")

Text(text = "Second Item")

Text(text = "Third Item")

}

}

@Preview(showBackground = true)

@Composable

fun RowPreview() {

ColumnRowLabTheme {

RowExample()

}

}

1. **Explanation**:
   * Row: Arranges the text items horizontally.
   * horizontalArrangement: Specifies how to space the items (e.g., SpaceBetween spreads them evenly across the row).
   * verticalAlignment: Aligns items vertically in the row.
2. **Run the app** to see the horizontally aligned text items.