**Lab Exercise 5- Working with LazyColumn, and LazyRowin Jetpack Compose**

**Objective:**

**Step 1: Displaying a Scrollable List with LazyColumn**

A LazyColumn is an efficient way to display a list of items that are only composed when they are visible on the screen.

1. Modify the code to display a simple scrollable list with LazyColumn:

import androidx.compose.foundation.lazy.LazyColumn

import androidx.compose.foundation.lazy.items

@Composable

fun LazyColumnExample() {

val itemsList = listOf("Item 1", "Item 2", "Item 3", "Item 4", "Item 5", "Item 6")

LazyColumn(

modifier = Modifier

.fillMaxSize()

.padding(16.dp)

) {

items(itemsList) { item ->

Text(text = item, modifier = Modifier.padding(8.dp))

}

}

}

@Preview(showBackground = true)

@Composable

fun LazyColumnPreview() {

ColumnRowLabTheme {

LazyColumnExample()

}

}

1. **Explanation**:
   * LazyColumn: A scrollable vertical list where only visible items are composed.
   * items: A convenient function to iterate over a list of data and display each item.
2. **Run the app** to see the list of items displayed in a vertical scrollable list.

**Step 2: Displaying a Horizontal Scrollable List with LazyRow**

A LazyRow is similar to LazyColumn, but it arranges the items horizontally.

1. Modify the code to display a horizontally scrollable list using LazyRow:

import androidx.compose.foundation.lazy.LazyRow

@Composable

fun LazyRowExample() {

val itemsList = listOf("Item A", "Item B", "Item C", "Item D", "Item E", "Item F")

LazyRow(

modifier = Modifier

.fillMaxWidth()

.padding(16.dp)

) {

items(itemsList) { item ->

Text(text = item, modifier = Modifier.padding(8.dp))

}

}

}

@Preview(showBackground = true)

@Composable

fun LazyRowPreview() {

ColumnRowLabTheme {

LazyRowExample()

}

}

1. **Explanation**:
   * LazyRow: A scrollable horizontal list that only renders visible items.
   * Similar to LazyColumn, but items are arranged horizontally.
2. **Run the app** to see the horizontally scrollable list of items.

**Step 3: Combining LazyColumn and LazyRow**

You can combine LazyColumn and LazyRow to display a more complex list with both vertical and horizontal scrolling.

1. Modify the code to create a LazyColumn where each row is a LazyRow:

@Composable

fun LazyColumnWithLazyRowExample() {

val itemsList = listOf("Row 1", "Row 2", "Row 3", "Row 4", "Row 5")

LazyColumn(

modifier = Modifier

.fillMaxSize()

.padding(16.dp)

) {

items(itemsList) { rowItem ->

Text(text = rowItem, modifier = Modifier.padding(8.dp))

LazyRow(

modifier = Modifier

.fillMaxWidth()

.padding(8.dp)

) {

items(6) { index ->

Text(text = "Item $index", modifier = Modifier.padding(8.dp))

}

}

}

}

}

@Preview(showBackground = true)

@Composable

fun LazyColumnWithLazyRowPreview() {

ColumnRowLabTheme {

LazyColumnWithLazyRowExample()

}

}

1. **Explanation**:
   * Each LazyColumn row contains a LazyRow that displays horizontally scrollable items.
2. **Run the app** to see both vertical and horizontal scrolling combined.