**Minor Project 1- Simple Registration Form Jetpack Compose**

**Objective:**

Create a simple registration form using Row and Column in Jetpack Compose. The form will collect a user’s name, email, and password. When the user submits the form, the input will be displayed in a Toast.

**Steps:**

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

1. Open **Android Studio**.
2. Create a new project:
   * Select **New Project** -> **Empty Compose Activity**.
   * Name the project SimpleRegistrationFormCompose.
3. Ensure your project includes the necessary Jetpack Compose dependencies in build.gradle:

dependencies {

implementation "androidx.activity:activity-compose:1.7.0"

implementation "androidx.compose.ui:ui:1.4.0"

implementation "androidx.compose.material3:material3:1.1.0"

implementation "androidx.compose.ui:ui-tooling-preview:1.4.0"

debugImplementation "androidx.compose.ui:ui-tooling:1.4.0"

}

**Step 2: Build the Registration Form Layout Using Row and Column**

In this step, we’ll design a simple registration form layout using Row and Column composables.

1. Open MainActivity.kt and define the form layout using Row and Column composables.

import android.os.Bundle

import android.widget.Toast

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.layout.\*

import androidx.compose.foundation.text.KeyboardOptions

import androidx.compose.material3.\*

import androidx.compose.runtime.\*

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.platform.LocalContext

import androidx.compose.ui.text.input.KeyboardType

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

import androidx.compose.ui.unit.dp

import com.example.simpleregistrationform.ui.theme.SimpleRegistrationFormTheme

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

SimpleRegistrationFormTheme {

RegistrationForm()

}

}

}

}

@Composable

fun RegistrationForm() {

// Form state variables

var name by remember { mutableStateOf("") }

var email by remember { mutableStateOf("") }

var password by remember { mutableStateOf("") }

val context = LocalContext.current

Column(

modifier = Modifier

.fillMaxSize()

.padding(16.dp),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

// Name Input Field

OutlinedTextField(

value = name,

onValueChange = { name = it },

label = { Text("Name") },

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(8.dp))

// Email Input Field

OutlinedTextField(

value = email,

onValueChange = { email = it },

label = { Text("Email") },

keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Email),

modifier = Modifier.fillMaxWidth()

)

Spacer(modifier = Modifier.height(8.dp))

// Password Input Field

OutlinedTextField(

value = password,

onValueChange = { password = it },

label = { Text("Password") },

keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Password),

modifier = Modifier.fillMaxWidth()

)

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

// Submit Button

Button(

onClick = {

Toast.makeText(

context,

"Name: $name\nEmail: $email\nPassword: $password",

Toast.LENGTH\_LONG

).show()

},

modifier = Modifier.fillMaxWidth()

) {

Text("Submit")

}

}

}

@Preview(showBackground = true)

@Composable

fun RegistrationFormPreview() {

SimpleRegistrationFormTheme {

RegistrationForm()

}

}

**Step 3: Understanding the Code**

* **State Management**:
  + Three variables (name, email, and password) are defined using remember and mutableStateOf. These store the user inputs.
* **Layout**:
  + The form layout is created using a Column, which contains an OutlinedTextField for each input (name, email, and password), and a Button for submission.
  + Spacer is used between form elements to add spacing.
* **Displaying Input Using Toast**:
  + The Button's onClick handler shows the user’s inputs using a Toast. The LocalContext.current is used to get the Context necessary to display the Toast.

**Step 4: Running the Registration Form**

1. Run the project on an emulator or physical device.
2. You will see a form with fields for entering the user’s name, email, and password, and a "Submit" button.
3. When the "Submit" button is pressed, a Toast will display the entered information (name, email, and password).

**Conclusion:**

In this lab exercise, you created a simple registration form using Row and Column in Jetpack Compose. You used state variables to manage the form inputs, and displayed the form values using a Toast. This exercise demonstrates how to build basic forms in Jetpack Compose with input validation and user feedback.