

```

package main.healthcare.sampletraining

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
//import androidx.compose.foundation.layout.BoxScopeInstance.align
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.modifier.modifierLocalConsumer
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import main.healthcare.sampletraining.ui.theme.SampleTrainingTheme
import org.intellij.lang.annotations.JdkConstants.HorizontalAlignment

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Column(modifier = Modifier.fillMaxSize(), horizontalAlignment =
Alignment.CenterHorizontally) {
                simpleText()
            }
            Column(modifier = Modifier.fillMaxSize(), verticalArrangement =
Arrangement.Center, horizontalAlignment = Alignment.CenterHorizontally) {
                simpleTextField()
                simpleTextField2()
            }
            Column(modifier = Modifier.fillMaxSize(), horizontalAlignment =
Alignment.CenterHorizontally, verticalArrangement = Arrangement.Bottom) {
                simpleButton()
            }
        }
    }
}

@Composable
fun simpleText() {
    Text(text = "Login Screen", modifier = Modifier.padding(16.dp),
        style = TextStyle.Default
    )
}

@Composable
fun simpleButton() {
    Button(modifier = Modifier.padding(16.dp), onClick = { /* TODO */ }) {

```

```

        Text(text="Submit")
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun simpleTextField() {
    var text by remember { mutableStateOf("") }
    TextField(value = text, onValueChange = {it -> text = it}, label = {
Text(text="User Name")})
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun simpleTextField2() {
    var text by remember { mutableStateOf("") }
    TextField(value = text, onValueChange = {it -> text = it}, label = {
Text(text="Password")})
}

@Preview
@Composable
fun simpleTextPreview() {
    Column(modifier = Modifier.fillMaxSize(),horizontalAlignment =
Alignment.CenterHorizontally){
        simpleText()
    }
    Column(modifier = Modifier.fillMaxSize(), verticalArrangement =
Arrangement.Center, horizontalAlignment = Alignment.CenterHorizontally){
        simpleTextField()
        simpleTextField2()
    }
    Column(modifier = Modifier.fillMaxSize(), horizontalAlignment =
Alignment.CenterHorizontally, verticalArrangement = Arrangement.Bottom) {
        simpleButton()
    }
}

```

