

Mobile Computing: To do List

Aniket Mishra

October 2024

1 Code: (Kotlin)

```
package com.example.todolist

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material3.Button
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBar
import androidx.compose.material3.TopAppBarState
import androidx.compose.runtime.Composable
import androidx.compose.runtime.MutableState
import androidx.compose.runtime.mutableStateListOf
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.todolist.ui.theme.TODOListTheme

class MainActivity : ComponentActivity() {
    @OptIn(ExperimentalMaterial3Api::class)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            TODOListTheme {
                TODOApp()
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun TODOApp() {

    val todoTask: MutableState<String> = remember {
```

```
        mutableStateOf("")
    }

    val todoTaskList: MutableList<String> = remember {
        mutableStateListOf()
    }

    val todoTaskValue: String = todoTask.value

    fun addTodo() {
        todoTaskList.add(todoTaskValue)
        todoTask.value = ""
    }

    Scaffold(
        topBar = { TopAppBar(title = { Text("Todo list") }, modifier = Modifier) },
        modifier = Modifier.fillMaxSize()
    ) { innerPadding ->

        Column(
            modifier = Modifier
                .padding(innerPadding)
                .fillMaxSize()
                .background(color = Color(0xFFeceff1)),
            verticalArrangement = Arrangement.Top,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {

            Spacer(modifier = Modifier.height(10.dp))
            OutlinedTextField(
                value = todoTaskValue,
                onChange = { value: String ->
                    todoTask.value = value
                },
                placeholder = {
                    Text("Todo Task")
                }
            )

            Button(
                modifier = Modifier.padding(vertical = 10.dp),
                onClick = { addTodo() },

                ) { Text("Add item") }

            Spacer(modifier = Modifier.height(20.dp))
        }
    }
}
```

```
        if (todoTaskList.isEmpty()) {
            Text("No item in the todo")
        } else {
            LazyColumn {
                items(todoTaskList) { item: String ->
                    Text(item)
                }
            }
        }
    }
}
```

Output:

