

Code for Blood Bank App

Done by Chinmay Joshi (20BAI10133)

```
package com.example.VIT_20BAI10133_blood_bank_app

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
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.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Text
import androidx.compose.material3.TextFieldDefaults
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.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.window.Dialog
import androidx.compose.ui.window.DialogProperties
import com.example.blood_bank_app.R

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            sign_in()
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@Composable
fun sign_in()
{
    var bloodtypes = listOf<String>("A+", "A-", "B+", "B-", "AB+", "AB-",
    "O+", "O-")
}
```

```

var name by remember{
    mutableStateOf("")
}
var age by remember {
    mutableStateOf("")
}
var showDialog by remember{
    mutableStateOf(false) }
var bloodType by remember { mutableStateOf("") }
Column(horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center,
    modifier = Modifier
        .background(color = Color.White)
        .padding(all = 20.dp)
        .fillMaxSize()) {
    Image(painter = painterResource(id = R.drawable.drop),
contentDescription = "", modifier = Modifier.padding(top = 40.dp))
    Spacer(modifier = Modifier.height(20.dp))
    OutlinedTextField(value = name, onValueChange = {name = it},
        label = { Text("Name") },
        colors = TextFieldDefaults.outlinedTextFieldColors(
            focusedBorderColor = Color(0xFFBB0A1E),
            focusedLabelColor = Color(0xFFBB0A1E),
            unfocusedBorderColor = Color.LightGray
        ),)
    Spacer(modifier = Modifier.height(20.dp))
    OutlinedTextField(value = age, onValueChange = {age = it},
        label = {Text("Age") },
        colors = TextFieldDefaults.outlinedTextFieldColors(
            focusedBorderColor = Color(0xFFBB0A1E),
            focusedLabelColor = Color(0xFFBB0A1E),
            unfocusedBorderColor = Color.LightGray
        ),)
    Spacer(modifier = Modifier.height(20.dp))
    OutlinedTextField(value = bloodType, onValueChange = {bloodType =
it},
        label = {Text("BloodType") },
        colors = TextFieldDefaults.outlinedTextFieldColors(
            focusedBorderColor = Color(0xFFBB0A1E),
            focusedLabelColor = Color(0xFFBB0A1E),
            unfocusedBorderColor = Color.LightGray
        ),)

    Spacer(modifier = Modifier.height(20.dp))
    Button(onClick = {
        showDialog = true
    },
        colors = ButtonDefaults.buttonColors(
            containerColor = Color(0xFFBB0A1E),
            contentColor = Color.White
        ),
        shape = RoundedCornerShape(10.dp),
        modifier = Modifier
            .fillMaxWidth()
            .padding(all = 30.dp)
    ) {
        Text("Next")
    }
    if (showDialog) {
        Dialog(
            onDismissRequest = { showDialog = false },

```

```
properties = DialogProperties(usePlatformDefaultWidth =  
false),  
    ) {  
        Column(  
            modifier = Modifier  
                .padding(all = 16.dp)  
                .background(Color.White)  
        ) {  
            Text("Name: ${name}", style = TextStyle(  
                fontSize = 20.sp,  
            ), modifier = Modifier.padding(all = 20.dp))  
            Text("Age: ${age}", style = TextStyle(  
                fontSize = 20.sp,  
            ), modifier = Modifier.padding(all = 20.dp))  
            Text("Blood Type: ${bloodType}", style = TextStyle(  
                fontSize = 20.sp,  
            ), modifier = Modifier.padding(all = 20.dp))  
            Button(  
                onClick = { showDialog = false }, colors =  
ButtonDefaults.buttonColors(  
                    containerColor = Color(0xFFBB0A1E),  
                    contentColor = Color.White  
                ),  
                shape = RoundedCornerShape(10.dp),  
                modifier = Modifier  
                    .fillMaxWidth()  
                    .padding(all = 20.dp)  
            ) {  
                Text("Close")  
            }  
        }  
    }  
}
```