Student Name: R.S.Logasubramaniyan

Registration Number: 20BME0566

ASSIGNMENT 2 FILE LINK:

https://drive.google.com/file/d/1sMdcx_EkRYgRSzrAeZf6nPi_BsQhVzgB/view?usp=share_link

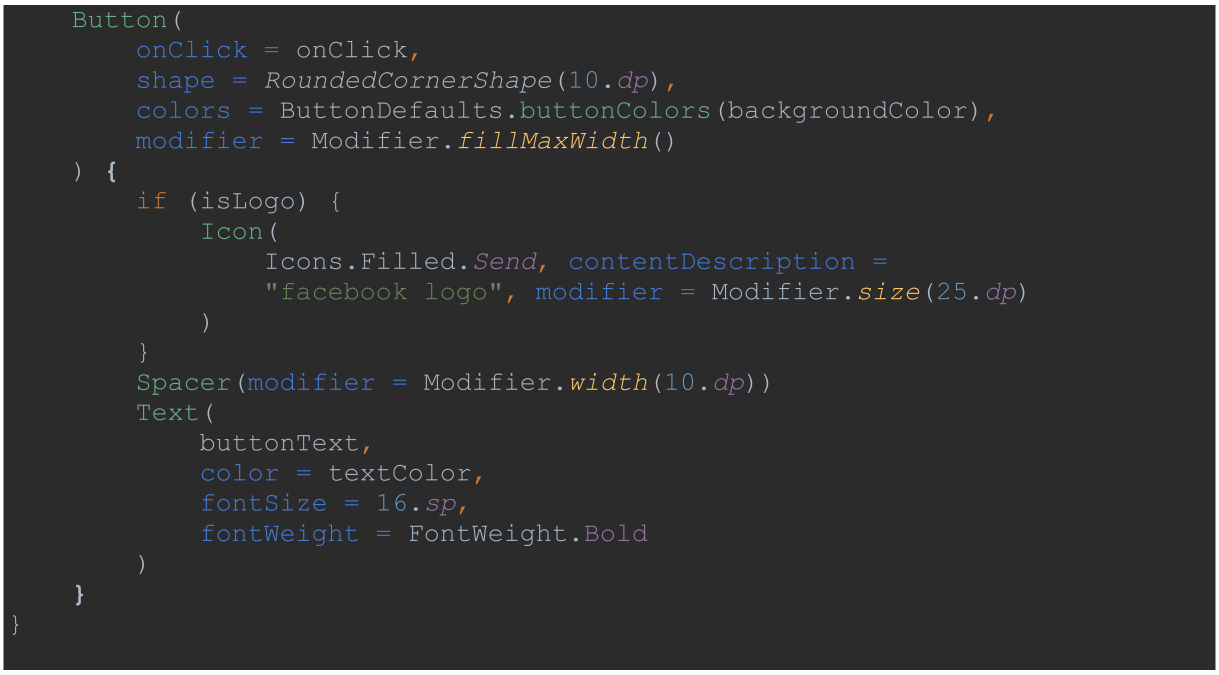
CODE:

```
package com.example.blood_bank

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.Image
import androidx.compose.foundation.border
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.ExperimentalLayoutApi
import androidx.compose.foundation.layout.PaddingValues
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.consumedWindowInsets
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.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.AccountBox
import androidx.compose.material.icons.filled.Favorite
import androidx.compose.material.icons.filled.Home
import androidx.compose.material.icons.filled.Person
import androidx.compose.material.icons.filled.Send
import androidx.compose.material3.BottomAppBar
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextField
import androidx.compose.material3.TextFieldDefaults
import androidx.compose.material3.TopAppBar
import androidx.compose.runtime.Composable
import androidx.compose.runtime.MutableState
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.drawBehind
import androidx.compose.ui.geometry.Offset
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.TextFieldValue
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.blood_bank.ui.theme.Blood_bankTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Blood_bankTheme {
                // A surface container using the 'background' color from
the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    App()
                }
            }
        }
    }
}

@Composable
fun App() {
    CustomScaffold()
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomScaffold() {
    Scaffold(topBar = { CustomTopBar() },
        bottomBar = { CustomBottomBar() },
        content = { pad -> MainContent(pad) }
    )
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomTopBar() {
    TopAppBar(
        title = {
            Row(
                verticalAlignment = Alignment.CenterVertically,
horizontalArrangement =
                Arrangement.SpaceBetween, modifier = Modifier
                    .fillMaxWidth()
                    .padding(end = 50
                        .dp
                    )
            ) {
                Icon(
                    Icons.Filled.AccountBox, contentDescription = "",
modifier = Modifier
                        .size(30.dp)
                )
                Image(
                    painterResource(id = R.drawable.img_4),
                    contentDescription = "",
                    modifier = Modifier.size(30.dp)
                )
                Spacer(modifier = Modifier)

            }
        },
        modifier = Modifier.drawBehind {
            drawLine(
                Color.LightGray,
                Offset(0f, size.height),
                Offset(size.width, size.height),
                5f
            )
        }
    )
}

@Composable
fun CustomBottomBar() {
    remember { mutableStateOf(0) }
    BottomAppBar(
        modifier = Modifier.drawBehind {
            drawLine(
                Color.LightGray,
                Offset(0f, 0f),
                Offset(size.width, 0f),
                4f
            )
        },
        containerColor = Color.White,
    ) {
        Row(
            modifier = Modifier.fillMaxWidth(),
            horizontalArrangement = Arrangement.SpaceEvenly,
            verticalAlignment =
            Alignment.CenterVertically
        ) {
            Icon(imageVector = Icons.Default.Home, "", modifier =
Modifier.size(30.dp))
            Icon(imageVector = Icons.Default.Favorite, "", modifier =
Modifier.size(30.dp))
            Icon(imageVector = Icons.Default.Person, "", modifier =
Modifier.size(30.dp))
        }
    }

}

@OptIn(ExperimentalLayoutApi::class)
@Composable
```

```
fun MainContent(padding: PaddingValues) {
    val primaryTextColor = remember {
        mutableStateOf(Color(115, 115, 115))
    }
    remember {
        mutableStateOf(Color(0, 55, 107))
    }
    val tertiaryTextColor = remember {
        mutableStateOf(Color.Black)
    }
    remember {
        mutableStateOf(Color(0, 149, 246))
    }
    remember {
        mutableStateOf(Color(0, 149, 246))
    }
    remember {
        mutableStateOf(Color(62, 174, 247, 255))
    }
    remember {
        mutableStateOf(Color(255, 255, 255))
    }
    remember {
        mutableStateOf(Color.White)
    }
    remember {
        mutableStateOf(Color.White)
    }
    val textFieldColor = remember {
        mutableStateOf(Color(250, 250, 250))
    }
    remember {
        mutableStateOf(Color(219, 219, 219))
    }
    val mobile = remember {
        mutableStateOf(TextFieldValue())
    }

    val fullName = remember {
        mutableStateOf(TextFieldValue())
    }

    val email = remember {
        mutableStateOf(TextFieldValue())
    }

    val address = remember {
        mutableStateOf(TextFieldValue())
    }

    remember {
        mutableStateOf(R.drawable.img)
    }
    val myDataList = remember {
        mutableListOf<String>()
    }

    val showList = remember {
        mutableStateOf(false)
    }
```

```kotlin
Column(
    modifier = Modifier.verticalScroll(rememberScrollState())
) {
    Column(
        modifier = Modifier
            .padding(20.dp)
            .padding(padding)
            .consumedWindowInsets(padding),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.img_3),
            contentDescription =
            "instagram logo",
            modifier = Modifier.size(width = 220.dp, height = 100.dp)
        )
        Spacer(modifier = Modifier.height(30.dp))
        Text(
            "Want to watch a miracle? Go and donate blood.",
            color = primaryTextColor.value,
            fontFamily =
            FontFamily.SansSerif,
            fontSize = 20.sp,
            fontWeight = FontWeight.SemiBold,
            textAlign = TextAlign.Center
        )

        Spacer(modifier = Modifier.height(40.dp))
        CustomTextField(
            modifier = Modifier.fillMaxWidth(),
            mutableValue =
            fullName,
            label = "Full Name",
            focusedColor = primaryTextColor.value,
            textColor = tertiaryTextColor.value,
            conColor = textFieldColor.value

        )
        Spacer(modifier = Modifier.height(10.dp))

        CustomTextField(
            modifier = Modifier.fillMaxWidth(),
            mutableValue =
            email,
            label = "Email",
            focusedColor = primaryTextColor.value,
            textColor = tertiaryTextColor.value,
            conColor = textFieldColor.value
        )
        Spacer(modifier = Modifier.height(10.dp))

        CustomTextField(
            modifier = Modifier
                .fillMaxWidth()
                .height(100.dp),
            mutableValue =
            address,
            label = "Address",
            focusedColor = primaryTextColor.value,
            textColor = tertiaryTextColor.value,
            conColor = textFieldColor.value
            )
            Spacer(modifier = Modifier.height(10.dp))

            CustomTextField(
                modifier = Modifier.fillMaxWidth(),
                mutableValue =
                mobile,
                label = "Mobile Number",
                focusedColor = primaryTextColor.value,
                textColor = tertiaryTextColor.value,
                conColor = textFieldColor.value
            )

            Spacer(modifier = Modifier.height(40.dp))

            CustomButton(
                buttonText = "Send my data to Blood Bank", onClick = {
                    myDataList.clear()
                    myDataList.addAll(
                        listOf(
                            fullName.value.text,
                            email.value.text,
                            address.value.text,
                            mobile.value.text
                        )
                    )
                    showList.value = false
                    showList.value = true

                }, isLogo =
                true
            )

            Spacer(modifier = Modifier.height(40.dp))

            if (showList.value) {
                CustomList(myList = myDataList)
            }
        }
    }
}

@Composable
fun CustomList(myList: MutableList<String>) {

    LazyColumn(
        modifier = Modifier
            .fillMaxWidth()
            .height(300.dp)
    ) {
        items(myList) { data ->
            CustomLazyItem(data = data)
        }

    }
}

@Composable
fun CustomLazyItem(data: String) {
    Column(
        horizontalAlignment = Alignment.CenterHorizontally, modifier =
```

```kotlin
Modifier
            .fillMaxWidth()
            .height(50.dp)
            .border(
                BorderStroke(0.2.dp, Color.Black), shape =
                RoundedCornerShape(4.dp)
            ), verticalArrangement = Arrangement.Center
    ) {
        Text(text = data, textAlign = TextAlign.Center)
    }
    Spacer(modifier = Modifier.height(20.dp))
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomTextField(
    modifier: Modifier = Modifier,
    mutableValue: MutableState<TextFieldValue>, label: String,
    placeholder: String
    = label,
    focusedColor: Color, conColor: Color = Color(250, 250, 250),
    isHideVal: Boolean = false, textColor: Color
) {
    TextField(
        modifier = modifier.border(
            BorderStroke(0.2.dp, focusedColor), RoundedCornerShape
                (4.dp)
        ),
        value = mutableValue.value,
        onValueChange = { mutableValue.value = it },
        label = { Text(text = label) },
        placeholder = { Text(text = placeholder) },
        colors = TextFieldDefaults.outlinedTextFieldColors(
            focusedBorderColor = Color.Transparent,
            focusedLabelColor = focusedColor,
            placeholderColor = focusedColor,
            textColor = textColor,
            unfocusedBorderColor = Color.Transparent,unfocusedLabelColor =
focusedColor,
            unfocusedLeadingIconColor = focusedColor,
            focusedLeadingIconColor = focusedColor,
            containerColor = conColor,

            ),

        visualTransformation = if (isHideVal) PasswordVisualTransformation(
            mask =
            '\u2022'
        ) else VisualTransformation.None,

        )
}

@Composable
fun CustomButton(
    buttonText: String,
    textColor: Color = Color.White,
    backgroundColor: Color = Color(0, 149, 246),
    onClick: () -> Unit = {},
    isLogo: Boolean = false
) {
```



```kotlin
    Button(
        onClick = onClick,
        shape = RoundedCornerShape(10.dp),
        colors = ButtonDefaults.buttonColors(backgroundColor),
        modifier = Modifier.fillMaxWidth()
    ) {
        if (isLogo) {
            Icon(
                Icons.Filled.Send, contentDescription =
                "facebook logo", modifier = Modifier.size(25.dp)
            )
        }
        Spacer(modifier = Modifier.width(10.dp))
        Text(
            buttonText,
            color = textColor,
            fontSize = 16.sp,
            fontWeight = FontWeight.Bold
        )
    }
}
```