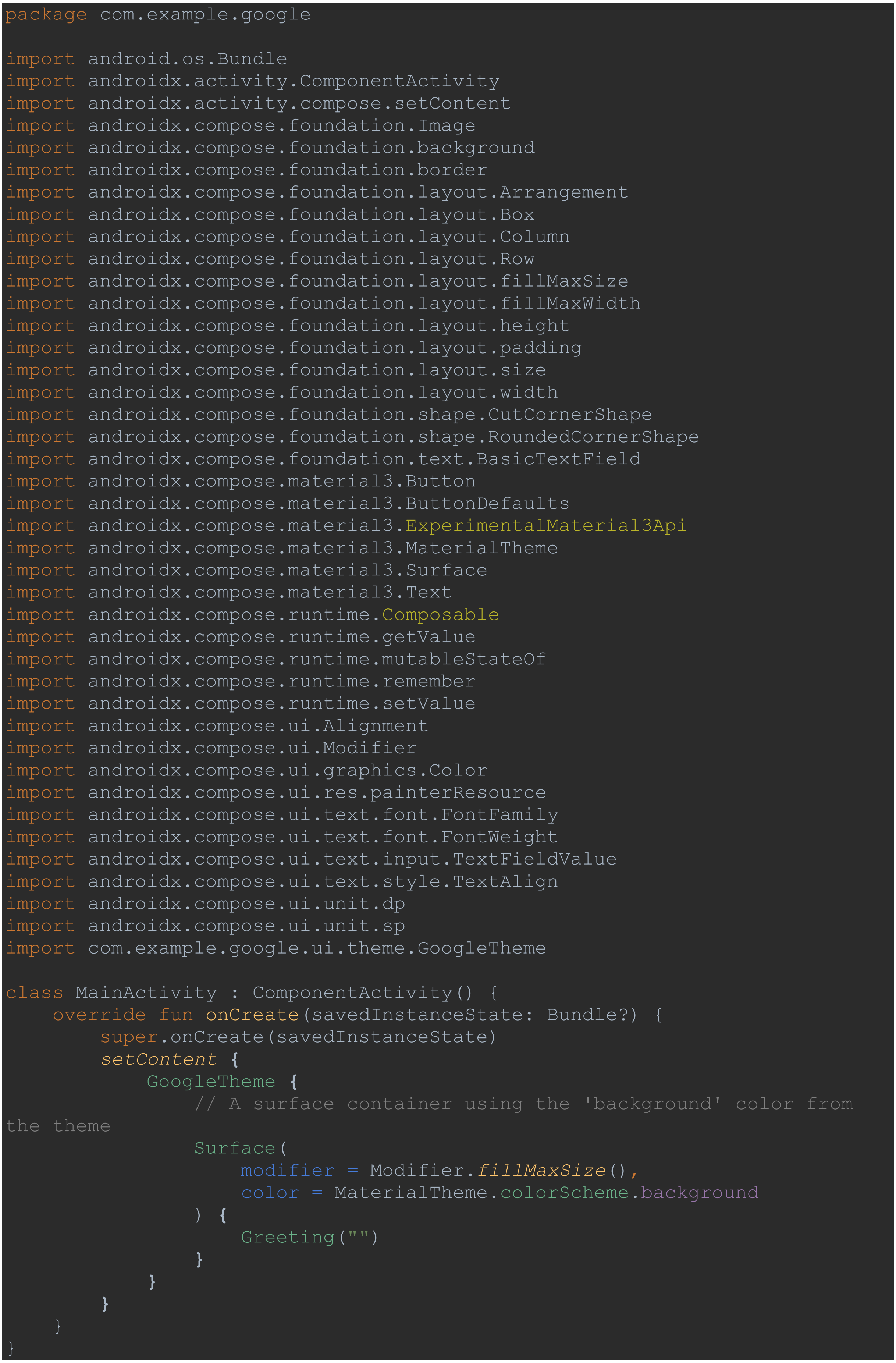
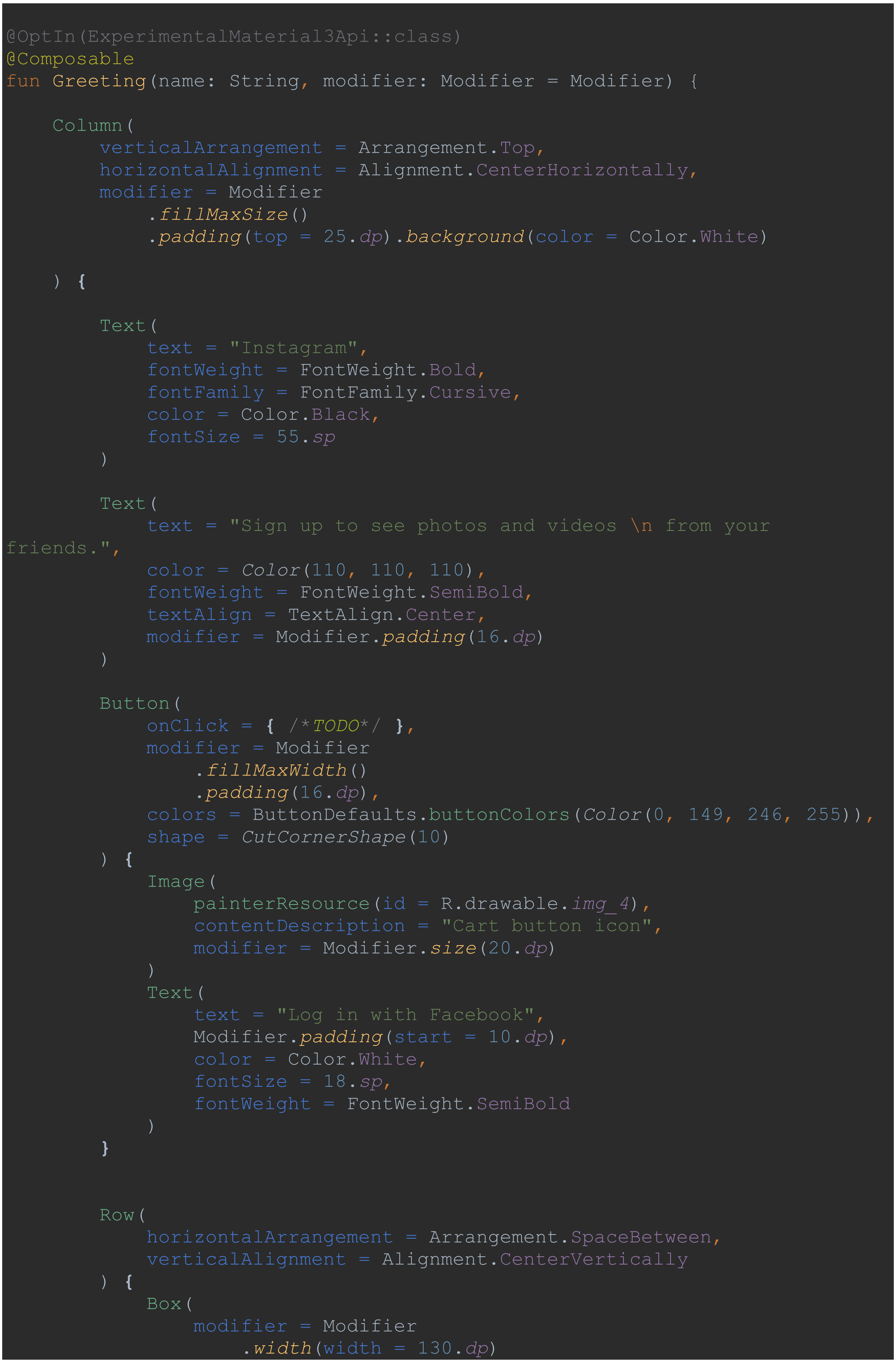
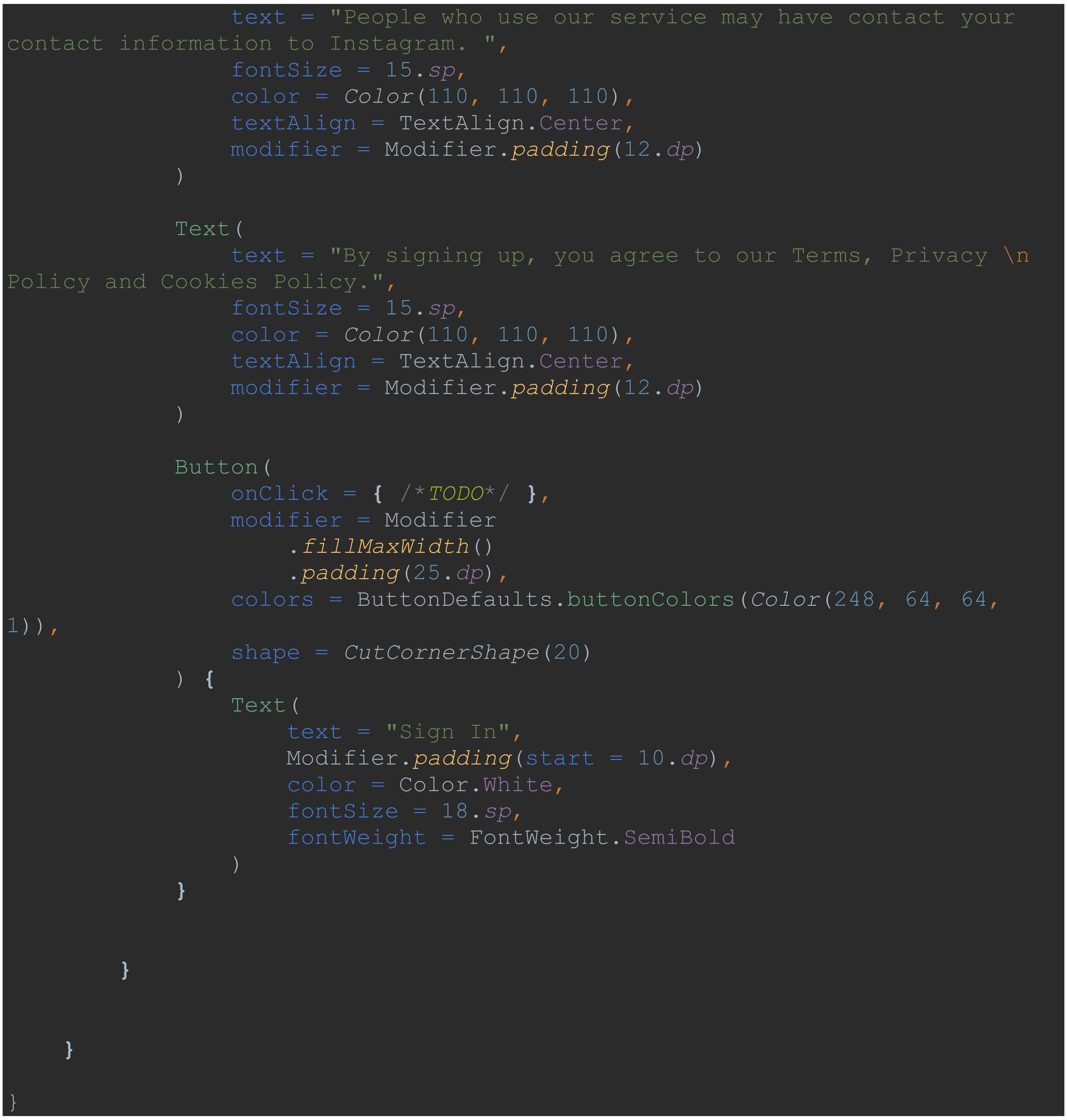


```kotlin
package com.example.google

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.border
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
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.shape.CutCornerShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.BasicTextField
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
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.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.TextFieldValue
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.google.ui.theme.GoogleTheme

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



```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {

    Column(
        verticalArrangement = Arrangement.Top,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier
            .fillMaxSize()
            .padding(top = 25.dp).background(color = Color.White)

    ) {

        Text(
            text = "Instagram",
            fontWeight = FontWeight.Bold,
            fontFamily = FontFamily.Cursive,
            color = Color.Black,
            fontSize = 55.sp
        )

        Text(
            text = "Sign up to see photos and videos \n from your
friends.",
            color = Color(110, 110, 110),
            fontWeight = FontWeight.SemiBold,
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(16.dp)
        )

        Button(
            onClick = { /*TODO*/ },
            modifier = Modifier
                .fillMaxWidth()
                .padding(16.dp),
            colors = ButtonDefaults.buttonColors(Color(0, 149, 246, 255)),
            shape = CutCornerShape(10)
        ) {
            Image(
                painterResource(id = R.drawable.img_4),
                contentDescription = "Cart button icon",
                modifier = Modifier.size(20.dp)
            )
            Text(
                text = "Log in with Facebook",
                Modifier.padding(start = 10.dp),
                color = Color.White,
                fontSize = 18.sp,
                fontWeight = FontWeight.SemiBold
            )
        }


        Row(
            horizontalArrangement = Arrangement.SpaceBetween,
            verticalAlignment = Alignment.CenterVertically
        ) {
            Box(
                modifier = Modifier
                    .width(width = 130.dp)
                    .height(height = 3.dp)
                    .background(color = Color.Blue)
                    .padding(end = 5.dp)

            )
            Text(
                text = "OR",
                modifier = Modifier.padding(10.dp),
                color = Color(184, 184, 184),
                fontWeight = FontWeight.SemiBold
            )
            Box(
                modifier = Modifier
                    .width(width = 130.dp)
                    .height(height = 3.dp)
                    .background(color = Color.Blue)
                    .padding(start = 5.dp)

            )

        }

        Column(
            verticalArrangement = Arrangement.SpaceEvenly,
            horizontalAlignment = Alignment.CenterHorizontally,
            modifier = Modifier.padding(10.dp)
        ) {

            var mn by remember { mutableStateOf(TextFieldValue("")) }
            var fn by remember { mutableStateOf(TextFieldValue("")) }
            var un by remember { mutableStateOf(TextFieldValue("")) }
            var pw by remember { mutableStateOf(TextFieldValue("")) }

            BasicTextField(
                value = mn,
                onValueChange = { mn = it },
                modifier = Modifier.padding(5.dp),
                decorationBox = { innerTextField ->
                    Row(
                        Modifier
                            .background(Color.White,
RoundedCornerShape(percent = 10))
                            .border(1.dp, Color.Black,
RoundedCornerShape(percent = 10))
                            .padding(12.dp)
                            .fillMaxWidth()
                    ) {
                        if (mn.text.isEmpty()) {
                            Text("Mobile Number")
                        }
                        innerTextField()
                    }
                },
            )

            BasicTextField(
                value = fn,
                onValueChange = { fn = it },
                modifier = Modifier.padding(5.dp),
                decorationBox = { innerTextField ->
                    Row(
                        Modifier
                            .background(Color.White,
RoundedCornerShape(percent = 10))
                            .border(1.dp, Color.Black,
RoundedCornerShape(percent = 10))
                            .padding(12.dp)
                            .fillMaxWidth()
                    ) {
                        if (fn.text.isEmpty()) {
                            Text("FullName")
                        }
                        innerTextField()
                    }
                },
            )

            BasicTextField(
                value = un,
                onValueChange = { un = it },
                modifier = Modifier.padding(5.dp),
                decorationBox = { innerTextField ->
                    Row(
                        Modifier
                            .background(Color.White,
RoundedCornerShape(percent = 10))
                            .border(1.dp, Color.Black,
RoundedCornerShape(percent = 10))
                            .padding(12.dp)
                            .fillMaxWidth()
                    ) {
                        if (un.text.isEmpty()) {
                            Text("User Name")
                        }
                        innerTextField()
                    }
                },
            )

            BasicTextField(
                value = pw,
                onValueChange = { pw = it },
                modifier = Modifier.padding(5.dp),
                decorationBox = { innerTextField ->
                    Row(
                        Modifier
                            .background(Color.White,
RoundedCornerShape(percent = 10))
                            .border(1.dp, Color.Black,
RoundedCornerShape(percent = 10))
                            .padding(12.dp)
                            .fillMaxWidth()
                    ) {
                        if (pw.text.isEmpty()) {
                            Text("Password")
                        }
                        innerTextField()
                    }
                },
            )

            Text(
```



```
                text = "People who use our service may have contact your
contact information to Instagram. ",
                fontSize = 15.sp,
                color = Color(110, 110, 110),
                textAlign = TextAlign.Center,
                modifier = Modifier.padding(12.dp)
            )

            Text(
                text = "By signing up, you agree to our Terms, Privacy \n
Policy and Cookies Policy.",
                fontSize = 15.sp,
                color = Color(110, 110, 110),
                textAlign = TextAlign.Center,
                modifier = Modifier.padding(12.dp)
            )

            Button(
                onClick = { /*TODO*/ },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(25.dp),
                colors = ButtonDefaults.buttonColors(Color(248, 64, 64,
1)),
                shape = CutCornerShape(20)
            ) {
                Text(
                    text = "Sign In",
                    Modifier.padding(start = 10.dp),
                    color = Color.White,
                    fontSize = 18.sp,
                    fontWeight = FontWeight.SemiBold
                )
            }


        }


    }

}
```