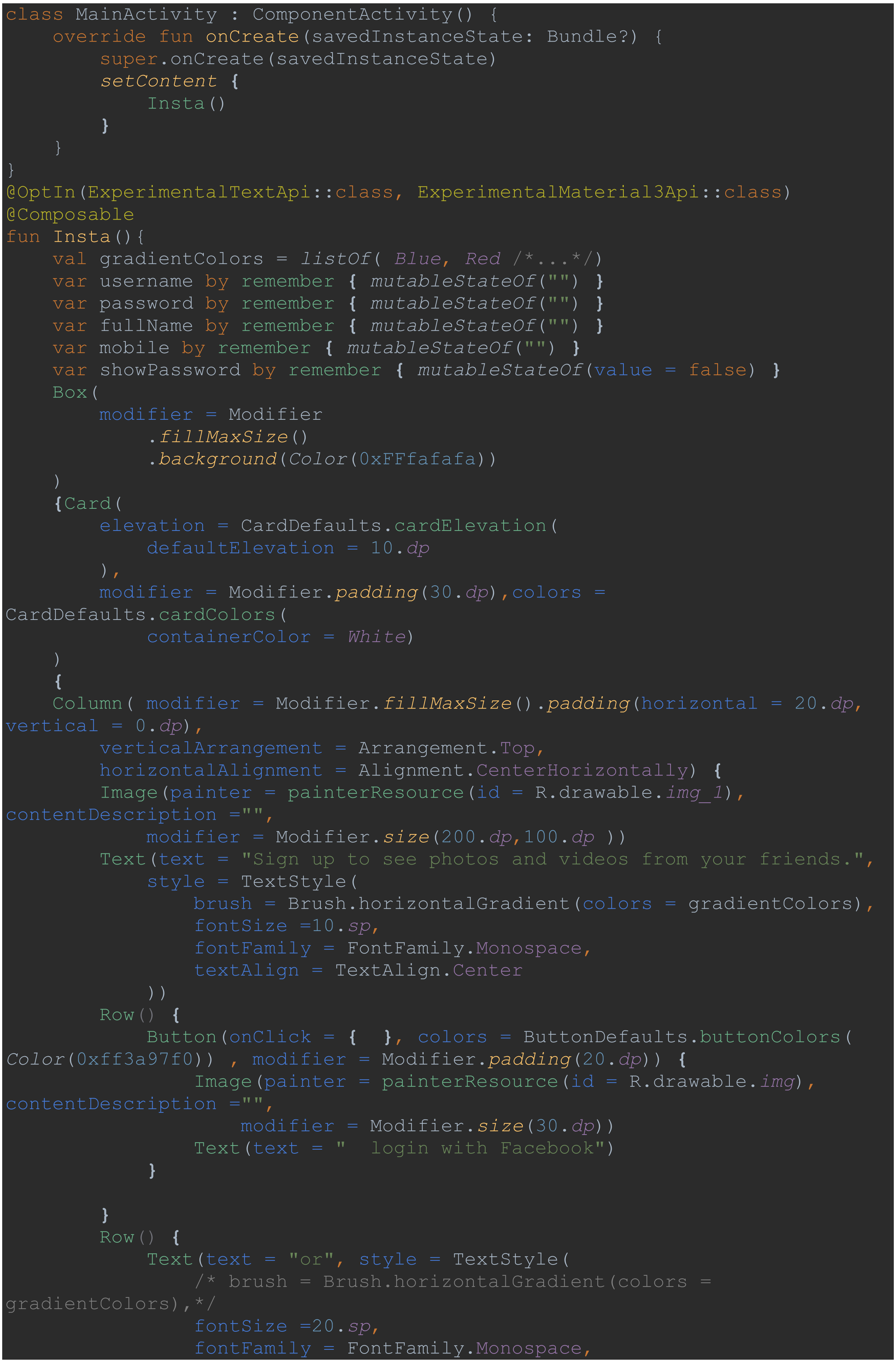
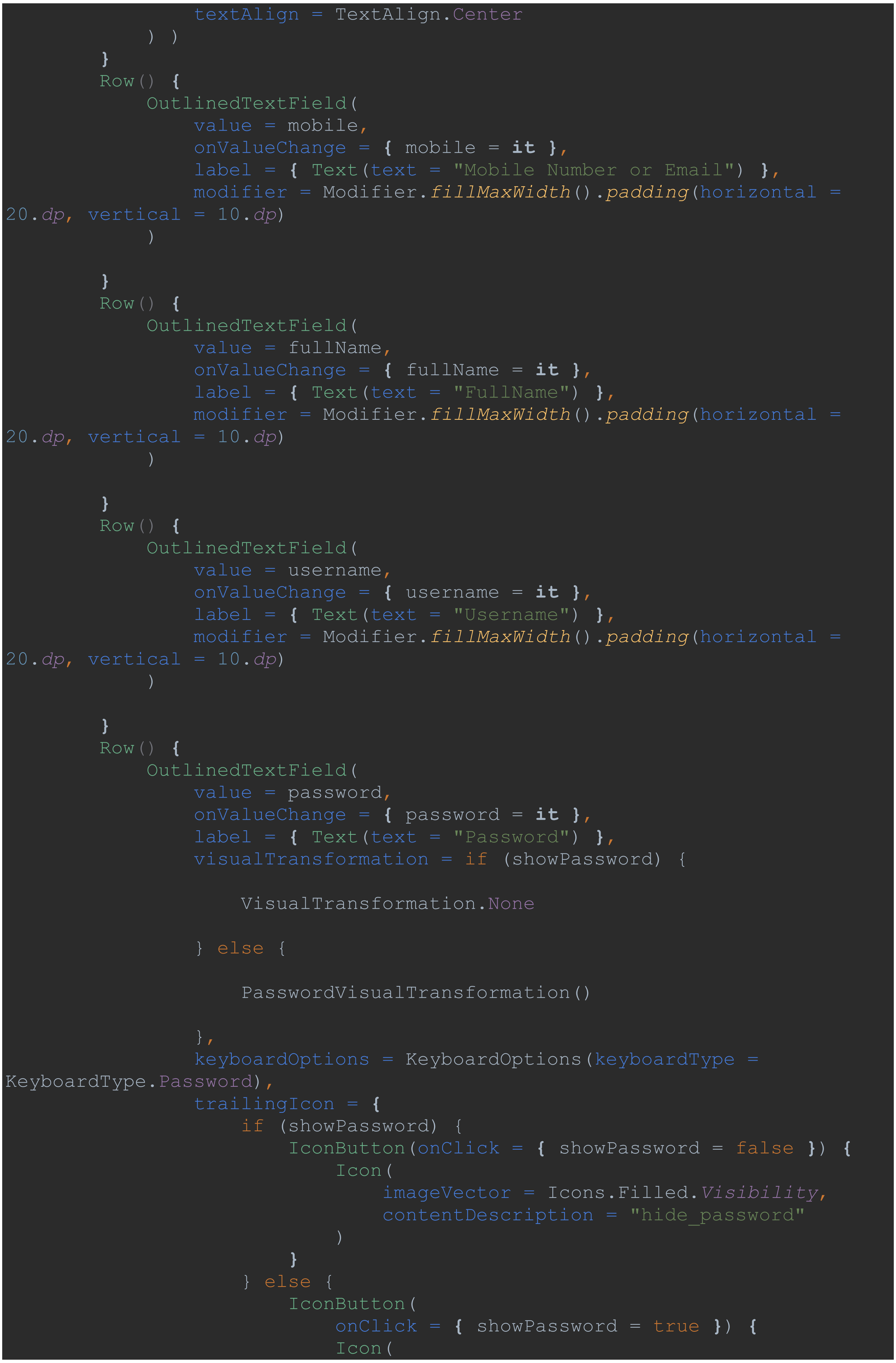
# Week 1  Assignment – Insta Login UI

## By Dyutin R 20BCG10060

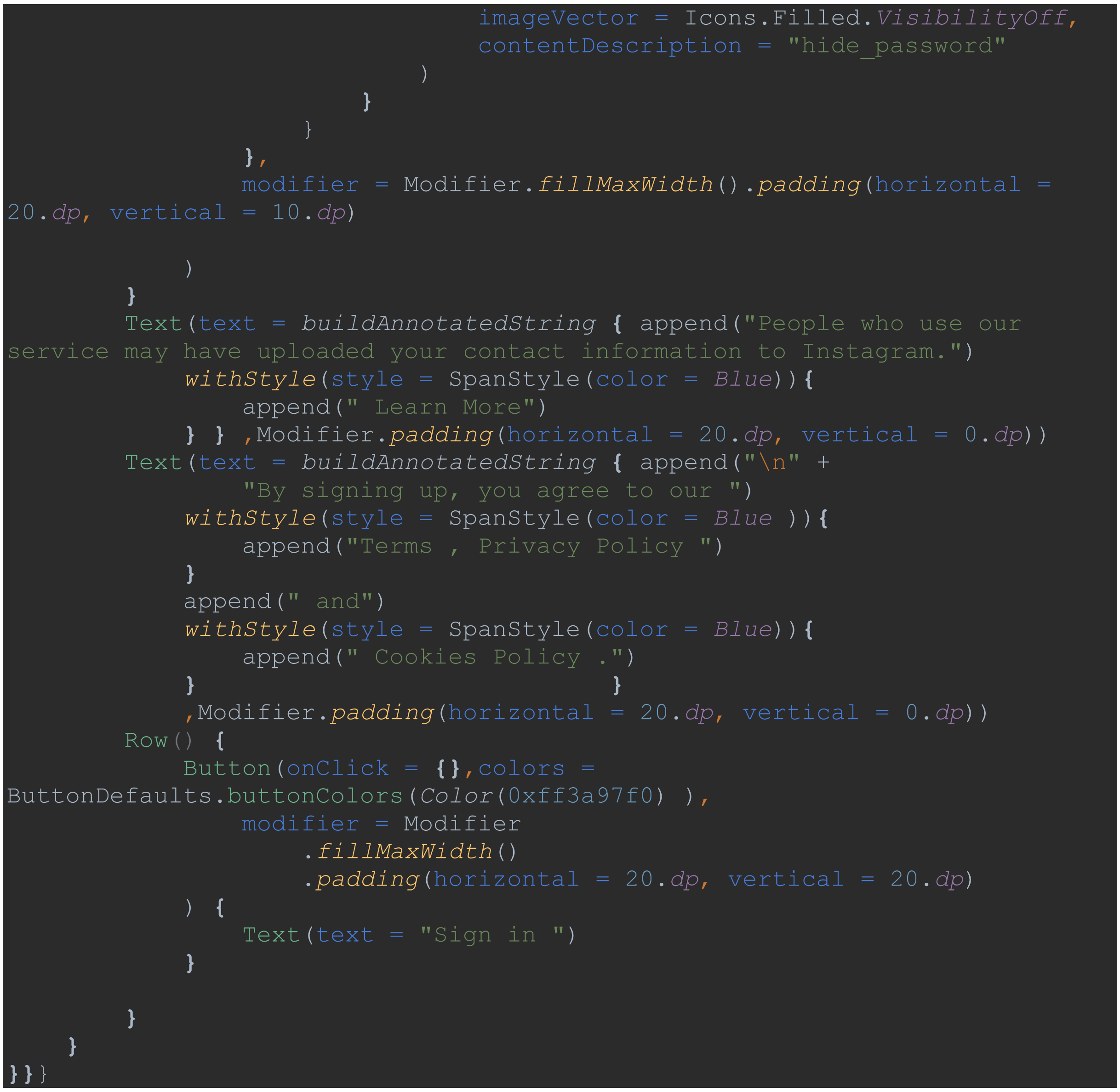
```kotlin
package com.example.instalogin

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.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.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Visibility
import androidx.compose.material.icons.filled.VisibilityOff
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.IconButton
import androidx.compose.material3.OutlinedTextField
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.Brush
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.Color.Companion.Blue
import androidx.compose.ui.graphics.Color.Companion.Red
import androidx.compose.ui.graphics.Color.Companion.White
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.ExperimentalTextApi
import androidx.compose.ui.text.SpanStyle
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.buildAnnotatedString
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.text.withStyle
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
```



```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Insta()
        }
    }
}
@OptIn(ExperimentalTextApi::class, ExperimentalMaterial3Api::class)
@Composable
fun Insta(){
    val gradientColors = listOf( Blue, Red /*...*/)
    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var fullName by remember { mutableStateOf("") }
    var mobile by remember { mutableStateOf("") }
    var showPassword by remember { mutableStateOf(value = false) }
    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFfafafa))
    )
    {Card(
        elevation = CardDefaults.cardElevation(
            defaultElevation = 10.dp
        ),
        modifier = Modifier.padding(30.dp),colors =
CardDefaults.cardColors(
            containerColor = White)
    )
    {
    Column( modifier = Modifier.fillMaxSize().padding(horizontal = 20.dp,
vertical = 0.dp),
        verticalArrangement = Arrangement.Top,
        horizontalAlignment = Alignment.CenterHorizontally) {
        Image(painter = painterResource(id = R.drawable.img_1),
contentDescription ="",
            modifier = Modifier.size(200.dp,100.dp ))
        Text(text = "Sign up to see photos and videos from your friends.",
            style = TextStyle(
                brush = Brush.horizontalGradient(colors = gradientColors),
                fontSize =10.sp,
                fontFamily = FontFamily.Monospace,
                textAlign = TextAlign.Center
            ))
        Row() {
            Button(onClick = {  }, colors = ButtonDefaults.buttonColors(
Color(0xff3a97f0)) , modifier = Modifier.padding(20.dp)) {
                Image(painter = painterResource(id = R.drawable.img),
contentDescription ="",
                    modifier = Modifier.size(30.dp))
                Text(text = "  login with Facebook")
            }

        }
        Row() {
            Text(text = "or", style = TextStyle(
                /* brush = Brush.horizontalGradient(colors =
gradientColors),*/
                fontSize =20.sp,
                fontFamily = FontFamily.Monospace,
```



```
                textAlign = TextAlign.Center
            ) )
        }
        Row() {
            OutlinedTextField(
                value = mobile,
                onValueChange = { mobile = it },
                label = { Text(text = "Mobile Number or Email") },
                modifier = Modifier.fillMaxWidth().padding(horizontal =
20.dp, vertical = 10.dp)
            )

        }
        Row() {
            OutlinedTextField(
                value = fullName,
                onValueChange = { fullName = it },
                label = { Text(text = "FullName") },
                modifier = Modifier.fillMaxWidth().padding(horizontal =
20.dp, vertical = 10.dp)
            )

        }
        Row() {
            OutlinedTextField(
                value = username,
                onValueChange = { username = it },
                label = { Text(text = "Username") },
                modifier = Modifier.fillMaxWidth().padding(horizontal =
20.dp, vertical = 10.dp)
            )

        }
        Row() {
            OutlinedTextField(
                value = password,
                onValueChange = { password = it },
                label = { Text(text = "Password") },
                visualTransformation = if (showPassword) {

                    VisualTransformation.None

                } else {

                    PasswordVisualTransformation()

                },
                keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Password),
                trailingIcon = {
                    if (showPassword) {
                        IconButton(onClick = { showPassword = false }) {
                            Icon(
                                imageVector = Icons.Filled.Visibility,
                                contentDescription = "hide_password"
                            )
                        }
                    } else {
                        IconButton(
                            onClick = { showPassword = true }) {
                            Icon(
```



```
                                imageVector = Icons.Filled.VisibilityOff,
                                contentDescription = "hide_password"
                            )
                        }
                    }
                },
                modifier = Modifier.fillMaxWidth().padding(horizontal =
20.dp, vertical = 10.dp)

            )
        }
        Text(text = buildAnnotatedString { append("People who use our
service may have uploaded your contact information to Instagram.")
            withStyle(style = SpanStyle(color = Blue)){
                append(" Learn More")
            } } ,Modifier.padding(horizontal = 20.dp, vertical = 0.dp))
        Text(text = buildAnnotatedString { append("\n" +
                "By signing up, you agree to our ")
            withStyle(style = SpanStyle(color = Blue )){
                append("Terms , Privacy Policy ")
            }
            append(" and")
            withStyle(style = SpanStyle(color = Blue)){
                append(" Cookies Policy .")
            }                                 }
            ,Modifier.padding(horizontal = 20.dp, vertical = 0.dp))
        Row() {
            Button(onClick = {},colors =
ButtonDefaults.buttonColors(Color(0xff3a97f0) ),
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(horizontal = 20.dp, vertical = 20.dp)
            ) {
                Text(text = "Sign in ")
            }

        }
    }
}}}
```