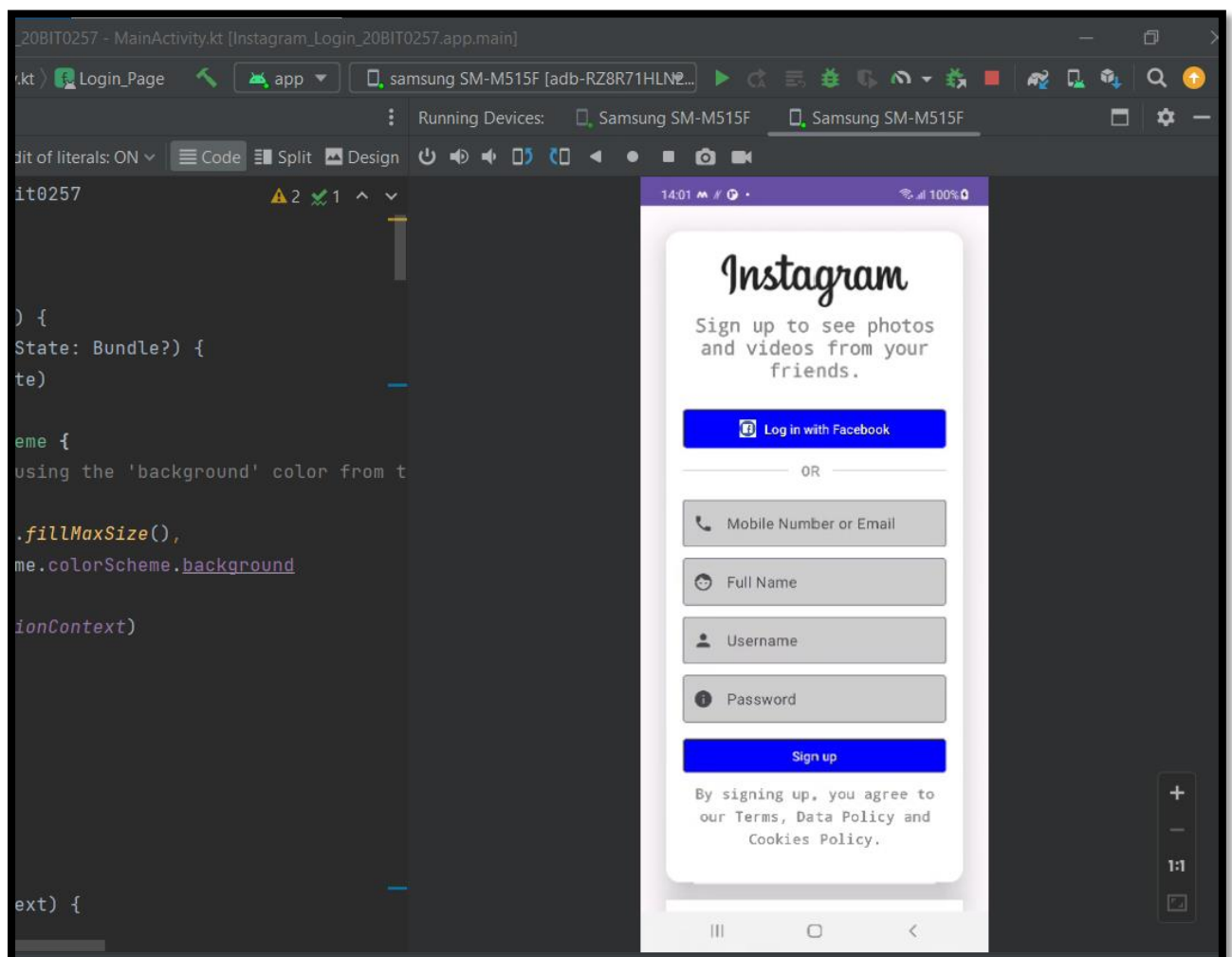


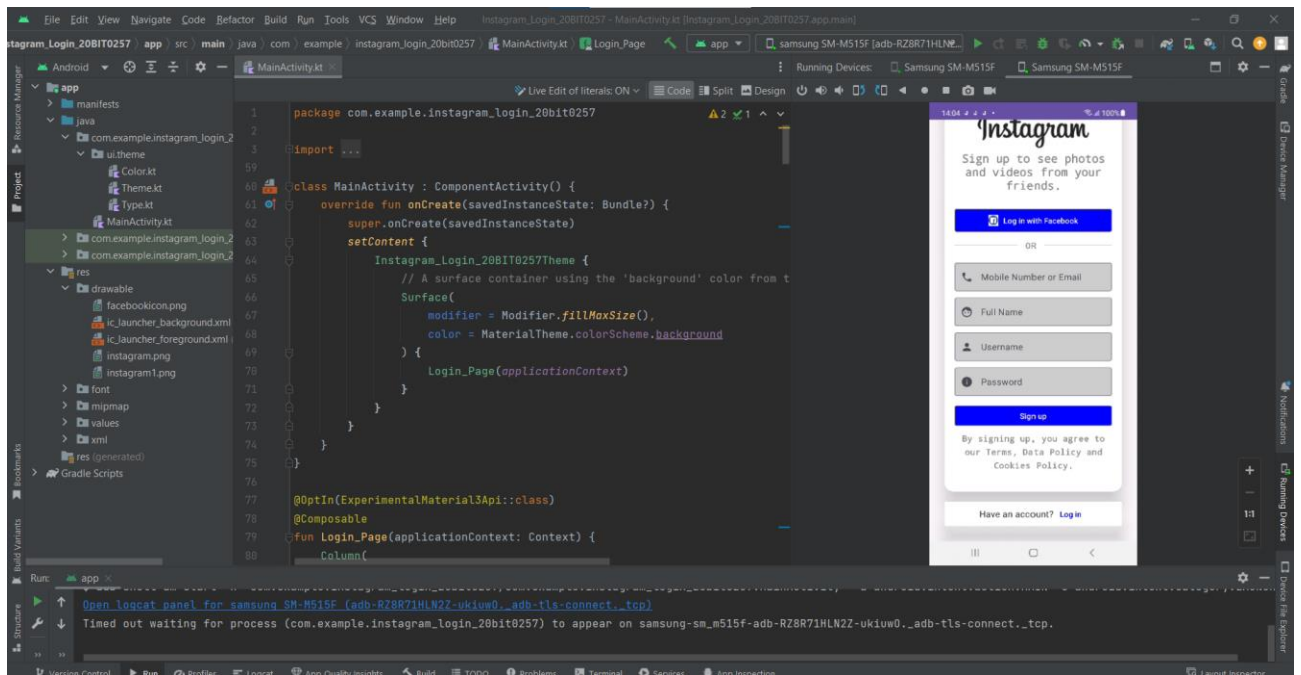
Week 1 – Android Application Development with Kotlin

Name: Suyash Patil

Registration Number: 20BIT0257

Instagram Login Page:





CODE:

```
package com.example.instagram_login_20bit0257

import android.content.Context
import android.os.Bundle
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Arrangement
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.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Face
import androidx.compose.material.icons.filled.Info
import androidx.compose.material.icons.filled.Person
import androidx.compose.material.icons.filled.Phone
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.Divider
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
```

```

import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedButton
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.material3.TextButton
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.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.ColorFilter
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.modifier.modifierLocalConsumer
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.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import
com.example.instagram_login_20bit0257.ui.theme.Instagram_Login_20BIT0257The
me

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Instagram_Login_20BIT0257Theme {
                // A surface container using the 'background' color from
the theme

                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Login_Page(applicationContext)
                }
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Login_Page(applicationContext: Context) {
    Column(
        modifier = Modifier
            .verticalScroll(rememberScrollState())
            .fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Card(
            elevation = CardDefaults.cardElevation(defaultElevation =
30.dp),

```

```

        modifier = Modifier.padding(top = 30.dp, start = 30.dp, end =
30.dp, bottom = 20.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        shape = RoundedCornerShape(5)
    ) {
        Column(
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally,
            modifier = Modifier
                .fillMaxSize()
                .padding(all = 20.dp)
        ) {
            Image(
                painterResource(id = R.drawable.instagram1),
                contentDescription = "",
                modifier = Modifier.size(width = 300.dp, height =
75.dp),
                contentScale = ContentScale.Crop
            )
            Text(
                text = "Sign up to see photos and videos from your
friends.",
                fontSize = 20.sp,
                fontFamily = FontFamily.Monospace,
                textAlign = TextAlign.Center,
                fontWeight = FontWeight.Bold,
                color = Color.Gray,
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(bottom = 20.dp)
            )
            OutlinedButton(
                onClick = { /*TODO*/ }, modifier = Modifier
                    .fillMaxWidth()
                    .padding(top = 10.dp, bottom = 10.dp), colors =
ButtonDefaults.buttonColors(
                        Color.Blue
                    ), shape = RoundedCornerShape(10)
            ) {
                Image(
                    painterResource(id = R.drawable.facebookicon),
                    contentDescription = "facebook",
                    modifier = Modifier
                        .size(30.dp)
                        .padding(end = 10.dp)
                )
                Text(text = "Log in with Facebook", color =
Color.White)
            }

            Row {
                Divider(
                    color = Color.LightGray,
                    thickness = 1.dp,
                    modifier = Modifier
                        .padding(top = 15.dp, bottom = 10.dp)
                        .width(125.dp),
                )
                Text(
                    text = "OR",
                    color = Color.Gray,

```

```

        textAlign = TextAlign.Center,
        fontSize = 15.sp,
        modifier = Modifier.padding(
            top = 5.dp,
            bottom = 5.dp,
            start = 15.dp,
            end = 15.dp
        ),
        fontFamily = FontFamily.Monospace,
        fontWeight = FontWeight.Bold
    )
    Divider(
        color = Color.LightGray,
        thickness = 1.dp,
        modifier = Modifier
            .padding(top = 15.dp, bottom = 10.dp)
            .width(135.dp)
    )
}

var contact by remember {
    mutableStateOf("")
}
OutlinedTextField(
    value = contact,
    onChange = { contact = it },
    label = {
        Text(
            text = "Mobile Number or Email"
        )
    },
    leadingIcon = {
        Icon(Icons.Default.Phone, contentDescription =
"contact")
    },
    modifier = Modifier
        .fillMaxWidth()
        .padding(bottom = 5.dp, top = 10.dp),
    colors = TextFieldDefaults.outlinedTextFieldColors(
        containerColor = Color.LightGray,
        focusedBorderColor = Color.Blue,
        focusedLabelColor = Color.Blue
    )
)

var name by remember {
    mutableStateOf("")
}
OutlinedTextField(
    value = name,
    onChange = { name = it },
    label = {
        Text(
            text = "Full Name"
        )
    },
    leadingIcon = {
        Icon(Icons.Default.Face, contentDescription = "full
name")
    },
    modifier = Modifier

```

```

        .fillMaxWidth()
        .padding(bottom = 5.dp),
        colors = TextFieldDefaults.outlinedTextFieldColors(
            containerColor = Color.LightGray,
            focusedBorderColor = Color.Blue,
            focusedLabelColor = Color.Blue
        )
    )

    var username by remember {
        mutableStateOf("")
    }
    OutlinedTextField(
        value = username,
        onValueChange = { username = it },
        label = {
            Text(
                text = "Username"
            )
        },
        leadingIcon = {
            Icon(Icons.Default.Person, contentDescription =
"username")
        },
        modifier = Modifier
            .fillMaxWidth()
            .padding(bottom = 5.dp),
        colors = TextFieldDefaults.outlinedTextFieldColors(
            containerColor = Color.LightGray,
            focusedBorderColor = Color.Blue, focusedLabelColor
= Color.Blue
        )
    )

    var password by remember {
        mutableStateOf("")
    }
    OutlinedTextField(
        value = password,
        onValueChange = { password = it },
        label = {
            Text(
                text = "Password"
            )
        },
        leadingIcon = {
            Icon(Icons.Default.Info, contentDescription =
"password")
        },
        modifier = Modifier
            .fillMaxWidth()
            .padding(bottom = 5.dp),
        colors = TextFieldDefaults.outlinedTextFieldColors(
            containerColor = Color.LightGray,
            focusedBorderColor = Color.Blue, focusedLabelColor
= Color.Blue
        ),
        visualTransformation = PasswordVisualTransformation(),
        keyboardOptions = KeyboardOptions(keyboardType =

```

```

KeyboardType.Password)
    )

    OutlinedButton(
        onClick = { login(username, password,
applicationContext) },
        modifier = Modifier
            .fillMaxWidth()
            .padding(bottom = 10.dp, top = 10.dp),
        colors = ButtonDefaults.buttonColors(Color.Blue), shape
= RoundedCornerShape(10)
    ) {
        Text(text = "Sign up", color = Color.White)
    }

    Text(
        text = "By signing up, you agree to our Terms, Data
Policy and Cookies Policy.",
        fontSize = 15.sp,
        fontFamily = FontFamily.Monospace,
        textAlign = TextAlign.Center,
        fontWeight = FontWeight.Bold,
        color = Color.Gray,
        modifier = Modifier
            .fillMaxWidth()
            .padding(bottom = 20.dp)
    )
}

Card(
    elevation = CardDefaults.cardElevation(defaultElevation =
30.dp),
    modifier = Modifier
        .padding(bottom = 30.dp, start = 30.dp, end = 30.dp)
        .fillMaxWidth(),
    colors = CardDefaults.cardColors(containerColor = Color.White),
    shape = RoundedCornerShape(5)
) {
    Row(
        modifier = Modifier.fillMaxWidth(),
        verticalAlignment = Alignment.CenterVertically,
        horizontalArrangement = Arrangement.Center
    ) {
        Text(text = "Have an account?", textAlign =
TextAlign.Center)
        TextButton(onClick = { /*TODO*/ }) {
            Text(text = "Log in", color = Color.Blue)
        }
    }
}
}

fun login(username: String, password: String, applicationContext: Context)
{
    if (username == "suyash" && password == "12345") {
        Toast.makeText(applicationContext, "Logged In Successfully",
Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(applicationContext, "Log In Failed",

```

```
Toast.LENGTH_SHORT).show()  
    }  
}
```

Video Link:

https://drive.google.com/drive/folders/1RBPEVU4opJ8j17_ErVOydlweAelZHypZ?usp=share_link