```kotlin
package com.example.ca2_android
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
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.layout.width
import androidx.compose.material3.Button
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.RadioButton
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Switch
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.platform.LocalContext
import androidx.compose.ui.text.Placeholder
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.ca2_android.ui.theme.CA2_AndroidTheme
import com.google.android.ads.mediationtestsuite.activities.HomeActivity

class MainActivity : ComponentActivity() {
```

```kotlin
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            SignupScreen()
        }
    }
}
@Preview(showBackground = true)
@Composable
fun SignupScreen() {

    val context = LocalContext.current

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var userType by remember { mutableStateOf("Student") }
    var darkMode by remember { mutableStateOf(false) }

    val bgColor = if (darkMode) Color.Black else Color.White
    val textColor = if (darkMode) Color.White else Color.Black

    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(bgColor)
            .padding(16.dp)
    ) {

        Text("Signup")

        Spacer(modifier = Modifier.height(16.dp))

        OutlinedTextField(
            value = username,
            onValueChange = { username = it },
            placeholder = { Text("Username") },
            modifier = Modifier.fillMaxWidth()
        )

        Spacer(modifier = Modifier.height(8.dp))

        OutlinedTextField(
            value = password,
```

```kotlin
        onValueChange = { password = it },
        placeholder = { Text("Password") },
        visualTransformation = PasswordVisualTransformation(),
        modifier = Modifier.fillMaxWidth()
)

Spacer(modifier = Modifier.height(16.dp))

Text("Select User Type")

Row(
    verticalAlignment = Alignment.CenterVertically
) {
    RadioButton(
        selected = userType == "Student",
        onClick = { userType = "Student" }
    )
    Text("Student")

    Spacer(modifier = Modifier.width(16.dp))

    RadioButton(
        selected = userType == "Admin",
        onClick = { userType = "Admin" }
    )
    Text("Admin")
}

Spacer(modifier = Modifier.height(16.dp))

Row(
    verticalAlignment = Alignment.CenterVertically
) {
    Text("Dark Mode")
    Spacer(modifier = Modifier.width(8.dp))
    Switch(
        checked = darkMode,
        onCheckedChange = { darkMode = it }
    )
}

Spacer(modifier = Modifier.height(24.dp))

Button(
```

```kotlin
        modifier = Modifier.fillMaxWidth(),
        onClick = {

            // ---------- VALIDATION USING IF-ELSE ----------
            if (username.length < 4) {
                Toast.makeText(context, "Username must be ≥ 4 characters",
Toast.LENGTH_SHORT).show()
            } else if (password.length < 6) {
                Toast.makeText(context, "Password must be ≥ 6 characters",
Toast.LENGTH_SHORT).show()
            } else if (userType.isEmpty()) {
                Toast.makeText(context, "Please select user type", Toast.LENGTH_SHORT).show()
            } else {

                // ---------- SUCCESS ----------
                Toast.makeText(context, "Signed Up Successfully", Toast.LENGTH_SHORT).show()

                Log.d("SIGNUP_DATA", "Username: $username")
                Log.d("SIGNUP_DATA", "Password: $password")
                Log.d("SIGNUP_DATA", "UserType: $userType")
                Log.d("SIGNUP_DATA", "DarkMode: $darkMode")

                val intent = Intent(context, HomeActivity::class.java)
                intent.putExtra("username", username)
                intent.putExtra("userType", userType)
                context.startActivity(intent)
            }
        }
    ) {
        Text("Signup")
    }
  }
}
```

```kotlin
package com.example.ca2_android

import ...

1 Usage
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            SignupScreen()
        }
    }
}

1 Usage
@Preview(showBackground = true)
@Composable
fun SignupScreen() {

    val context = LocalContext.current

    var username by remember { mutableStateOf( value = "") }
    var password by remember { mutableStateOf( value = "") }
    var userType by remember { mutableStateOf( value = "Student") }
    var darkMode by remember { mutableStateOf( value = false) }

    val bgColor = if (darkMode) Color.Black else Color.White
    val textColor = if (darkMode) Color.White else Color.Black

    Column(
        modifier = Modifier
            .fillMaxSize()
```