Task 1 — Project Setup, Run Log & Guided Walkthrough

Student name : VARSHITHA M

Course : Android App Development with Kotlin

Date : 20-02-2026

Environment Checklist

Item	Details
Android Studio	Hedgehog / Iguana (Latest Installed)
Language	Kotlin
UI Toolkit	Jetpack Compose
Minimum SDK	API 24
Target SDK	Latest available
Emulator	Pixel 6
Android Version	Android 14
Internet Connection	Required for Gradle & SDK downloads

Step-by-Step Setup Log (With Timestamps)

Time	Action	Result
5:00 PM	Opened Android Studio	Welcome screen displayed
5:02 PM	Clicked New Project	Project templates opened
5:03 PM	Selected Empty Compose Activity	Configuration page opened
5:04 PM	Named project "GreetingCard"	Project created
5:05 PM	Gradle Sync started	Dependencies downloading
5:07 PM	SDK components missing	Installed automatically
5:10 PM	Opened AVD Manager	Pixel 6 emulator launched
5:12 PM	Clicked Run ▶	App compiled
5:23 PM	App opened in emulator	Greeting message displayed
5:45 PM	Modified text to my name	Preview updated successfully
6:20 PM	Connect to Android Device	Connected sucessfully

Errors Faced & Solutions

Problem 1: First build took long time

Reason: Gradle downloading dependencies

Solution: Waited and ensured stable internet connection

Problem 2: Emulator slow startup

Solution: Enabled hardware acceleration (Windows Hypervisor)

Code Understanding

What does setContent {} do?

The setContent {} block defines the UI of the application using Jetpack Compose.
Instead of XML layout files, the UI is written directly in Kotlin code inside this block.
Android reads this block and draws the screen based on the composable functions written inside it.

What is @Composable?

@Composable marks a function that creates a UI element.
These functions describe what should appear on the screen rather than how to draw it step-by-step.
When data changes, Compose automatically updates only the required parts of the UI.

Lessons Learned

- First project build takes time because dependencies download
- Jetpack Compose replaces XML layouts with Kotlin UI functions
- Preview feature helps test UI without running emulator

Codelab Tasks Completed

- As part of the setup, the official Android Developer onboarding codelabs were completed to ensure the development environment works correctly.

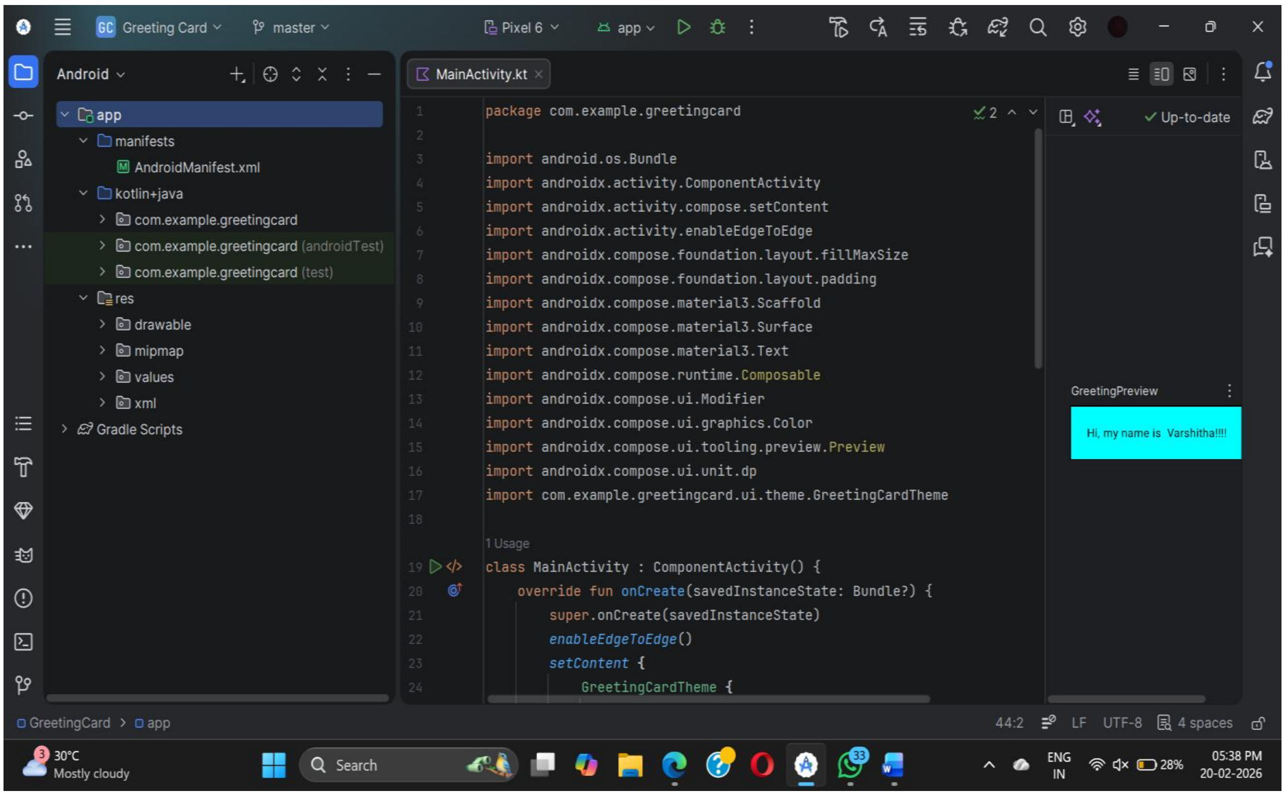
Codelab Task	**Status**
Download & Install Android Studio	Completed
Create First Android App	Completed
Run App on Android Emulator	Completed
Connect Android Device	Completed

These steps confirmed that the SDK, emulator, and development tools were properly configured before building the Jetpack Compose project.

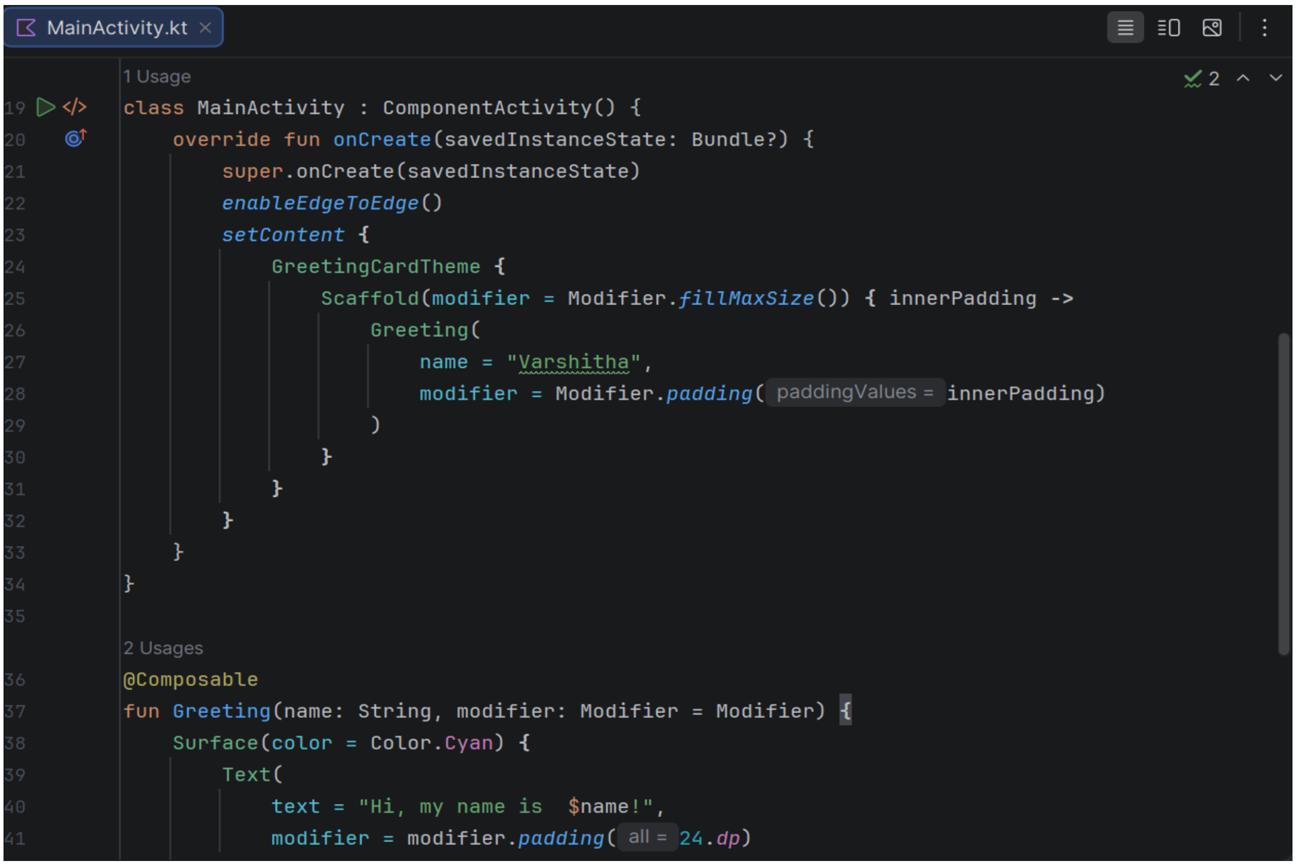
Conclusion

The Android Studio environment was successfully configured and verified by creating and running a Jetpack Compose project.
The application displayed the greeting message correctly on the emulator, confirming that the development environment is ready for further Compose development.
The environment was validated using both the Android Developer codelab verification steps and successful execution of the Compose project.

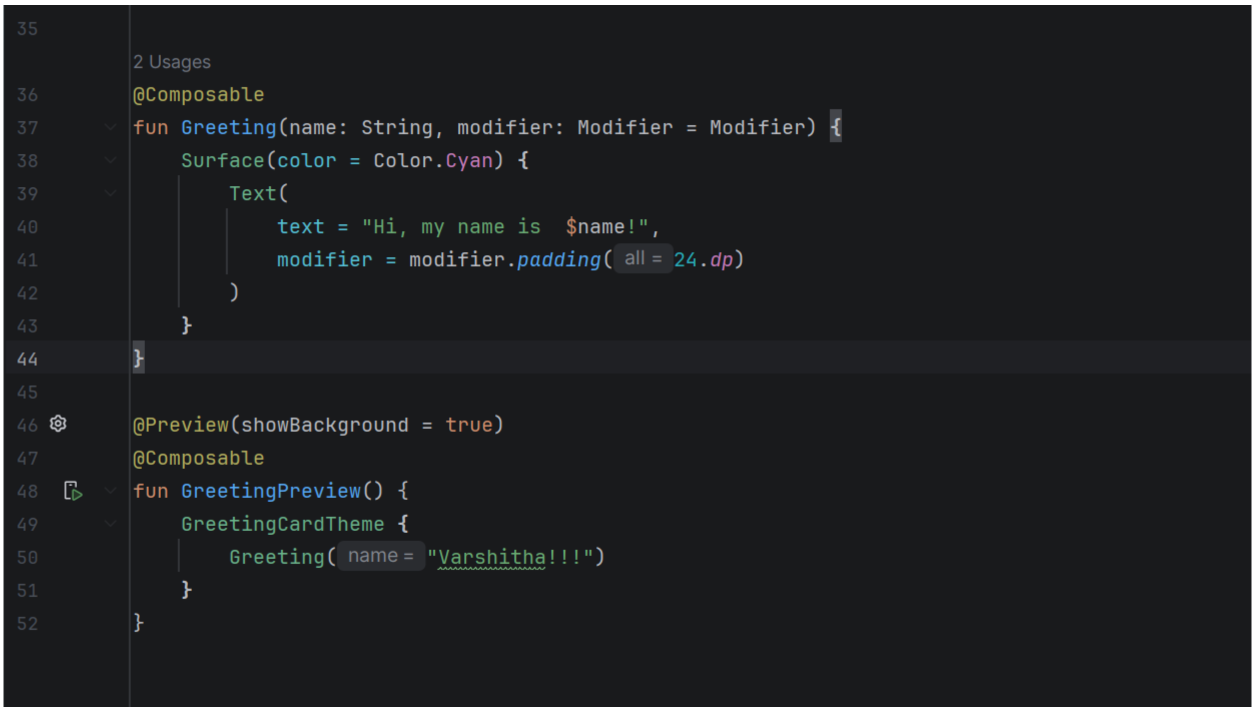
SCREEN SHOTS:



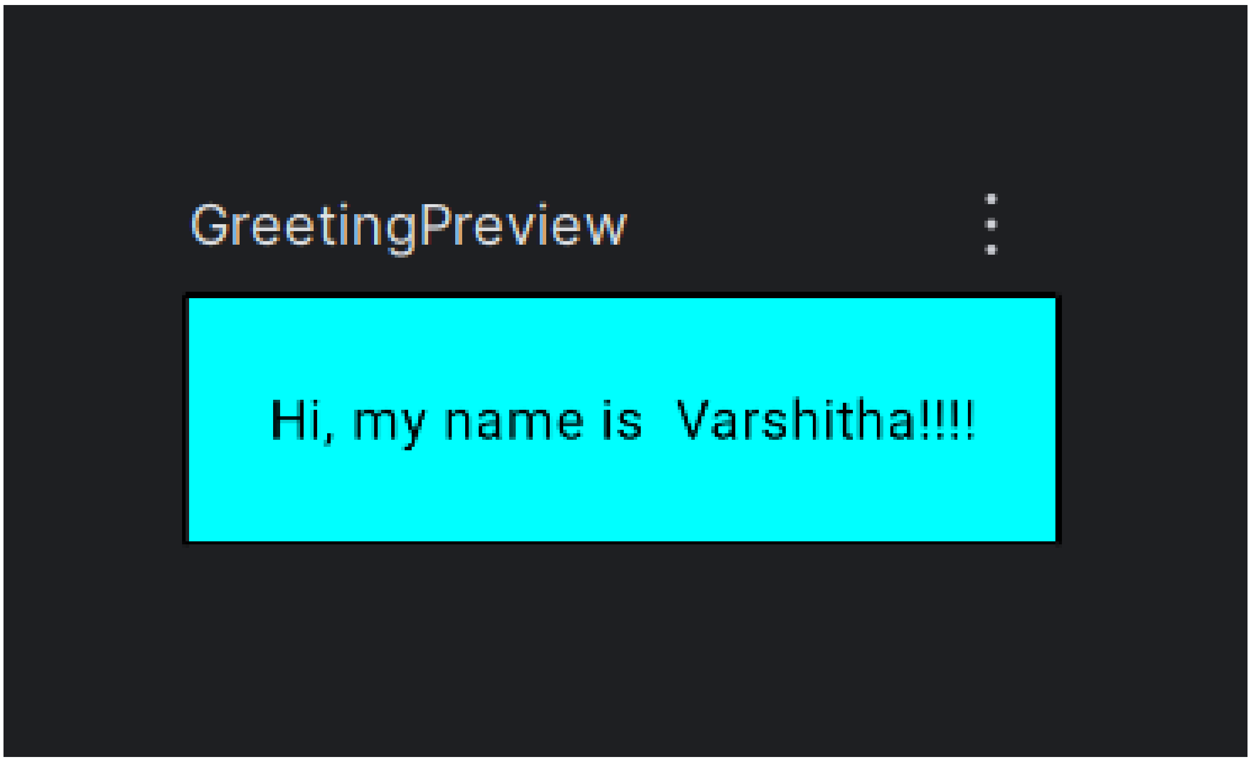
```kotlin
package com.example.greetingcard

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.example.greetingcard.ui.theme.GreetingCardTheme

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



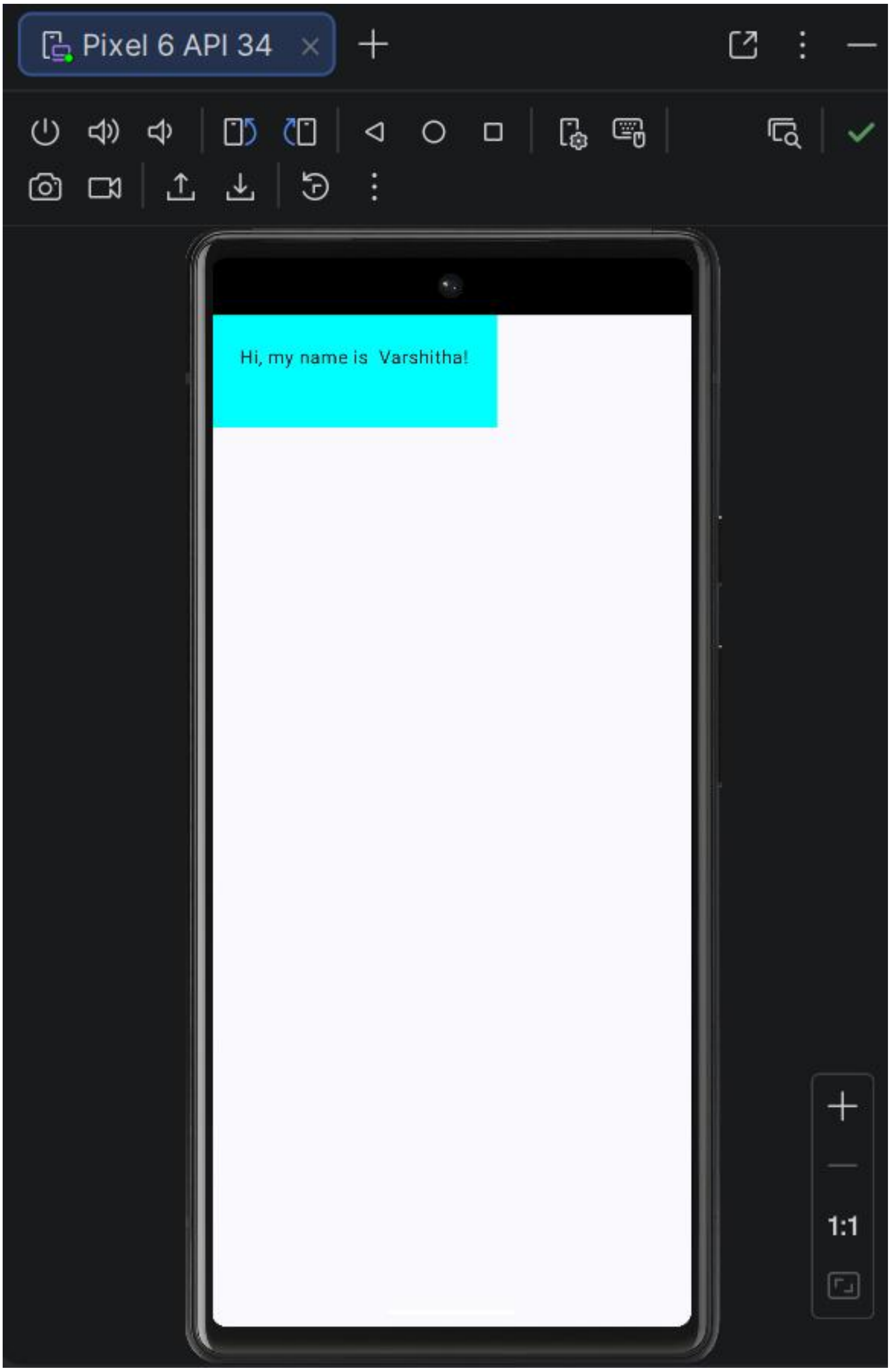
```kotlin
1 Usage
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            GreetingCardTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Greeting(
                        name = "Varshitha",
                        modifier = Modifier.padding( paddingValues = innerPadding)
                    )
                }
            }
        }
    }
}

2 Usages
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = Color.Cyan) {
        Text(
            text = "Hi, my name is  $name!",
            modifier = modifier.padding( all = 24.dp)
```



```
2 Usages
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = Color.Cyan) {
        Text(
            text = "Hi, my name is  $name!",
            modifier = modifier.padding( all = 24.dp)
        )
    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    GreetingCardTheme {
        Greeting( name = "Varshitha!!!")
    }
}
```



```
Hi, my name is  Varshitha!!!!
```



Hi, my name is Varshitha!