

# Session 3: AADK Task 3 – Compose Layouts: Text and Image

**Student Name:** Srajan

**Session:** Session 3 – Android App Development Kickstart

---

## Task Overview

In this task, I explored how Text and Image composables work together to build structured and visually balanced UI screens in Jetpack Compose.

I planned layouts before implementation, ensuring proper hierarchy, spacing, and alignment of components.

## Key Concepts Applied

1. Use of Text() and Image() composables to display content clearly.
2. Layout primitives such as Column, Row, and Box for UI structure.
3. Modifiers like padding(), fillMaxWidth(), size(), and align() to control layout behavior.
4. Proper drawable resource management using painterResource().
5. Accessibility improvements through meaningful contentDescription for images.
6. Basic performance awareness by keeping layouts optimized and simple.

## Birthday Greeting Card Compose Example

```
@Composable
fun GreetingText(message: String, from: String, modifier: Modifier = Modifier) {
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = modifier
            .fillMaxWidth()
            .padding(16.dp)
    ) {
        Text(
            text = message,
            fontSize = 28.sp
        )
        Text(
            text = from,
            fontSize = 20.sp
        )
    }
}
```

```

    }
}

@Composable
fun GreetingImage(message: String, from: String, modifier: Modifier = Modifier) {
    val image = painterResource(R.drawable.androidparty)

    Box(modifier) {
        Image(
            painter = image,
            contentDescription = "Birthday background image",
            contentScale = ContentScale.Crop,
            modifier = Modifier.fillMaxSize()
        )
        GreetingText(
            message = message,
            from = from,
            modifier = Modifier
                .fillMaxSize()
                .padding(8.dp)
        )
    }
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    BirthdayTheme {
        GreetingImage(
            message = "Happy Birthday Sam!",
            from = "From Emma"
        )
    }
}

```