Workshop: Building the "Hero Card" App

Class 3: Thinking in Compose Time: 60 Minutes

Part 0: What is Jetpack Compose?

Before we code, you need to understand the shift in mindset.
The Old Way (Imperative/XML):
You build a UI like a construction worker. You find a specific wall (View) by its ID, then you paint it red. Later, if you want it blue, you find it again and paint over it.

- Example: findViewById(R.id.textView).setText("Hello")

The New Way (Declarative/Compose):
You build a UI like an architect. You hand a blueprint to the system that says, "If the user is logged in, the screen looks like this. If not, it looks like that." You don't manually update views; you just redraw the blueprint.

- Example: if (loggedIn) Text("Hello")

The @Composable Annotation:
This is a magic sticker you put on a function. It tells the compiler: "This function doesn't return data (like an Int or String). This function draws UI on the screen."

Part 1: Images & Resources (Theory)

Before we start coding the header, we need a custom image. In Android, you don't just dump files anywhere; there is a system.

1. Raster vs. Vector:

- **Raster (PNG, JPG, WebP):** Made of pixels. Great for photos (like our hero's face). If you stretch them too much, they get blurry.
- **Vector (XML / SVG):** Made of math instructions. Great for icons and logos. You can stretch them infinitely, and they stay sharp.

2. The `res` (Resources) Folder: Look at the project structure on the left `app > res`.

- **`drawable`:** This is your workhorse. Put your hero photos, background images, and icon vectors here.
- **`mipmap`: STOP!** Do not put UI images here. This folder is *strictly* for the App Launcher Icon (the icon the user clicks on their home screen).

3. The Golden Rule of Naming: Android resource filenames must be:

- **Lowercase only** (`a-z`)
- **No spaces** (use underscores `_`)
- **No starting numbers**
- *Bad:* `My Hero.jpg`, `123image.png`
- *Good:* `hero_profile.jpg`, `background_red.png`

Step 1: Import Your Custom Image

1. **Find an Image:** Go to Google Images or Unsplash and download a cool "Robot" or "Superhero" image (JPG or PNG).
2. **Rename It:** Rename the file on your computer to **`hero_avatar.jpg`** (or `.png`). *Remember: All lowercase!*
3. **Import:**
    - In Android Studio, locate the **`res > drawable`** folder on the left.
    - Drag and drop your file from your computer into the `drawable` folder.
    - A dialog will appear. Click **Refactor/OK**.

Step 2: Project Setup & The Container

(Same setup as before)

Add this function to `MainActivity.kt`:

```
@Composable
fun HeroProfile() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFF0F0F0)) // Light Gray
            .padding(16.dp)

    ) {
        // Content goes here...
    }

}
```

Step 3: The Header (Box & Custom Image)

Concept: contentScale When an image size doesn't match the `Image` container size, we have to decide how to resize it.

- `ContentScale.Fit`: Shows the whole image (might leave empty space).
- `ContentScale.Crop`: Fills the whole container (might cut off edges of the image). Usually best for headers.

Task: Add this inside your `Column` block. Notice we are now using `R.drawable.hero_avatar`.

```
// 1. The Wrapper Box
Box(
    modifier = Modifier
        .fillMaxWidth()
        .height(200.dp) // Fixed height
        .clip(RoundedCornerShape(16.dp)) // Round the corners of the Box
        .background(Color.Gray)
) {
    // LAYER 1: The Custom Image (Bottom)
    Image(
        // R.drawable points to the folder where we dropped the file
        painter = painterResource(id = R.drawable.hero_avatar),
        contentDescription = "Hero Image",
        contentScale = ContentScale.Crop, // Zoom image to fill the bounds
        modifier = Modifier.fillMaxSize()
    )

    // LAYER 2: The Badge (Top)
    Text(
        text = "Active",
        color = Color.White,
        modifier = Modifier
            .align(Alignment.TopEnd) // Gravity: Top-Right
            .padding(12.dp)
            .background(Color.Green, RoundedCornerShape(8.dp))
            .padding(8.dp)
    )
}
```

Step 4: Content & Spacing (Text)

(Standard Text Components)

Add this inside `Column`, after the `Box`:

```
    Spacer(modifier = Modifier.height(16.dp))
    Text(
        text = "Pixel Droid",
        fontSize = 30.sp,
        fontWeight = FontWeight.Bold,
        color = Color.Black
    )

    Text(
        text = "A high-tech android capable of hacking any mainframe in seconds.",
        fontSize = 16.sp,
        color = Color.DarkGray
    )
    Spacer(modifier = Modifier.height(24.dp))
```

Step 5: Horizontal Layouts (Row)

(Introduction to creating helper functions)

Task A: Create the helper function at the **very bottom** of your file (outside `HeroProfile`):

```
@Composable
fun StatBadge(label: String, value: String) {
    Column(horizontalAlignment = Alignment.CenterHorizontally) {
        Text(text = value, fontSize = 22.sp, fontWeight = FontWeight.Bold)
        Text(text = label, fontSize = 12.sp, color = Color.Gray)
    }
}
```

Task B: Use it inside your main `HeroProfile` Column:

```
    Row(
        modifier = Modifier.fillMaxWidth(),
        horizontalArrangement = Arrangement.SpaceBetween
    ) {
        StatBadge("Strength", "95")
        StatBadge("Intellect", "120")
```

```
    StatBadge("Speed", "300")
  }
  Spacer(modifier = Modifier.height(24.dp))
```

Step 6: Interaction (Button)

(Clickables and lambda functions)

Add to `Column`:

```
  Button(
    onClick = { /* Logic later */ },
    modifier = Modifier.fillMaxWidth(),
    colors = ButtonDefaults.buttonColors(containerColor = Color.Blue)
  ) {
    Text(text = "Hire Hero")
  }
```

🛑 Homework: The Villain Card
Download a new image of a "Villain" or "Monster".
Import it into res/drawable (Name it villain_avatar.jpg).
Create VillainProfile() and use this new image.
Extra Challenge: Add a 2dp Border around the Villain's image using the border modifier on the Box.
Hint: Modifier.border(2.dp, Color.Red, RoundedCornerShape(16.dp))
Delivery at: https://www.dropbox.com/request/6aUkImHMHi6e5O6QqXxI