Technical Requirements – Sticky Ad Banner

Challenge is accessible on Memberspot: https://pl-coding.mymemberspot.io/library/jx3b7Qik9ip5qpNI8IF2/jmJLYnGZ2RljR6im5zVR/JFWndhOnbRP7Ph3Xtamx/details

🎭 Scenario

This mini-app simulates an electronics store screen with a product grid and a Black Friday advertisement banner that initially appears in the middle of the list and becomes a sticky header when scrolling.

🎨 Figma Mockups

https://www.figma.com/design/OUKoX7XUafdcBm3mqWdqiu/Black-Friday-Madness?node-id=2006-2123

🔤 Fonts - Host Grotesk

🎯 Feature Goal

Implement an interactive product list where the Black Friday banner initially appears within the content, becomes fixed to the top when scrolling, and disappears permanently when dismissed with the “×” button.

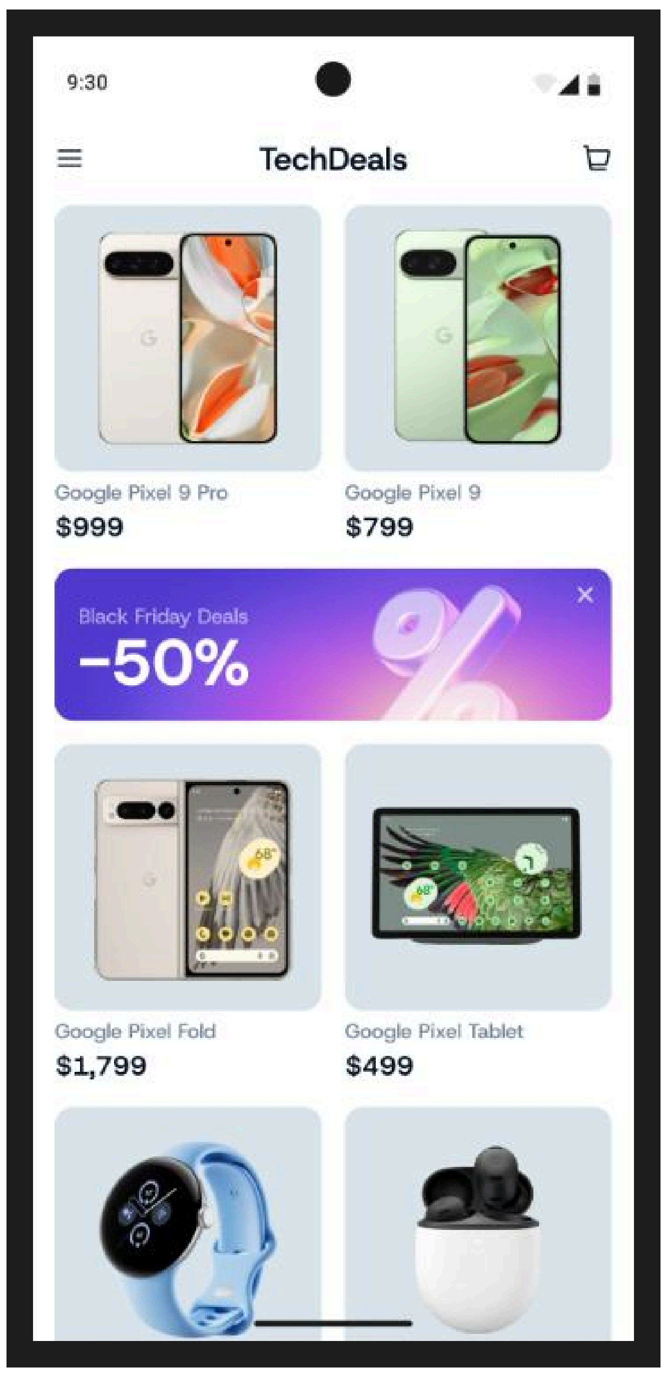
📌 Requirements

Top Bar

- Positioned at the top of the screen.
- Used purely for structural layout — **no functionality required**.
- Contains:
  - Left — burger menu icon.
  - Center — store name "TechDeals".
  - Right — cart icon.

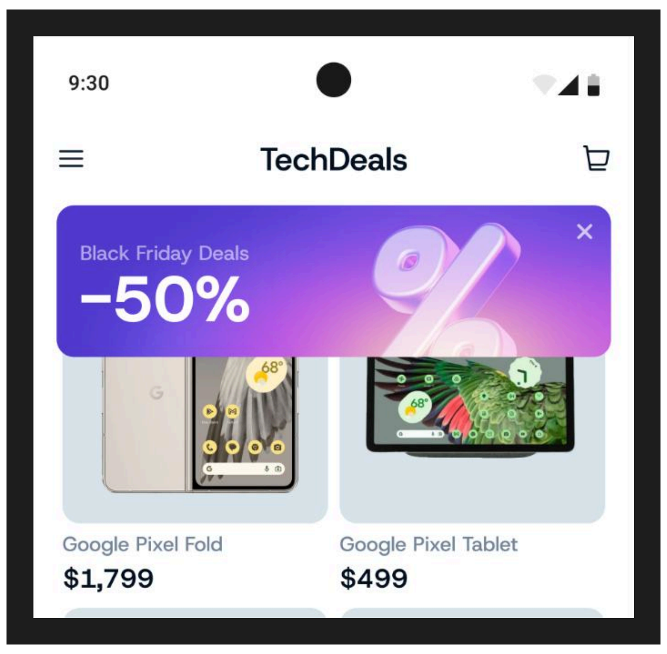
Product List

- Implemented using **LazyVerticalGrid** with two columns.



- Each product item includes:
  - product image;
  - name;
  - price.

Sticky Advertisement Banner



- Initially positioned **within the product list** (after the first two items).
- When scrolled upward, the banner transitions into a **sticky header**, fixed at the top of the screen.
- Contains:
  - Text: *"Black Friday Deals -50%"*;
  - Close button "×";
  - Background image or gradient related to the Black Friday theme.
- When the "×" button is pressed:
  - The banner disappears completely;
  - It does **not reappear** during further scrolling or recomposition.

⚙️ Logic

1. The screen loads with a product list, and the banner is positioned after the first two products.
2. When scrolling down, the banner becomes sticky and remains fixed at the top.
3. When the "×" button is pressed, the banner disappears, and the list occupies all available space.
4. The banner does not appear again even after UI updates or recomposition.

🧪 Initial Data for Testing

- **Product grid structure:** 2×N layout.
- **Banner placement:** after the first two product items.
- **Example product data:**
  - Google Pixel 9 Pro — $999
  - Google Pixel 9 — $799
  - Google Pixel 8 Pro — $899
  - Google Pixel 8 — $699
  - Google Pixel Fold — $1799
  - Google Pixel Tablet — $499
  - Google Pixel Watch 2 — $349
  - Google Pixel Buds Pro — $199
  - Google Nest Hub (2nd Gen) — $99
  - Google Nest Audio — $99

🤔 What's Allowed?

- Standard Android/Jetpack libraries only.
- Use Material 3 components or simple custom UI elements.
- Static hardcoded product list (no external data source).

⚠️ What's not important

- Functional behavior of the cart or menu icons.
- Fetching or loading data from a remote source.
- Complex or physics-based animations.
- Tablet optimization (mobile view only).

🔗 Useful Links for This Challenge

- Full Guide to Lazy Grid in Jetpack Compose
- How to Create a Lazy Column With Categories
- Official Docs: LazyVerticalGrid
- Sticky Header in LazyColumn
- Stateful vs. Stateless Composables
- State Hoisting in Compose
- Managing State in Jetpack Compose (Codelab)

🏆 Submission & Rewards

- Successfully submitting this challenge via the **/submit-challenge** command on Discord grants you **100 XP**.
- Your submission must include:
    a. A **Gist link** with your implementation.
    b. A **screen recording** (max 20 seconds) showing:
        - The initial state with the banner in the middle of the list;
        - Scrolling down — the banner becomes sticky at the top;
        - Pressing “×” — the banner disappears;
        - Further scrolling — the banner does not reappear.