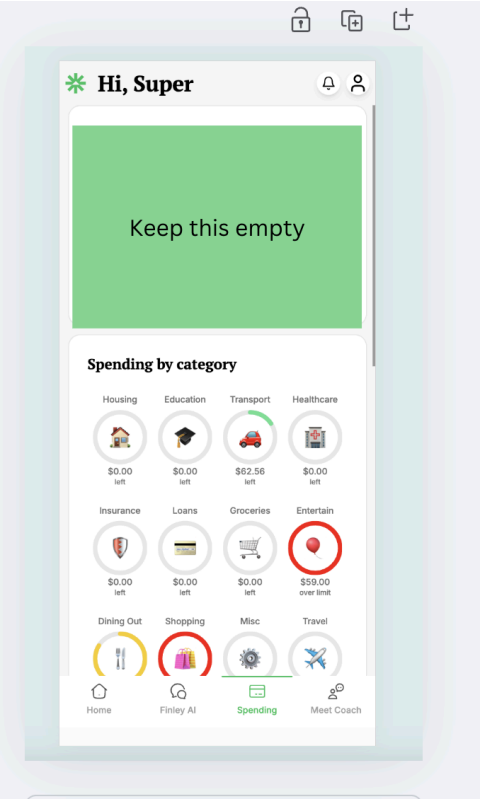
# Flutter - Mid Level Resource 2

## Financial Dashboard with Category Spending

**Time Allocation:** 1 day

### Objective

Build a financial dashboard interface showing spending breakdown by categories, similar to the Finley app home screen.




### Requirements

**UI Components:**

- Top app bar with lock, copy, and share icons
- Greeting header "Hi, User" with notification bell and profile icons

- Empty state placeholder area (green box with "Keep this empty" text)
- "Spending by category" section with a grid of category cards
- Bottom navigation bar (Home, Finley AI, Spending, Meet Coach)

**Category Grid:**

- 12 spending categories displayed in a 4×3 grid layout
- Each category card should display:
  - Category icon
  - Category name
  - Amount spent (formatted as currency)
  - Status text ("left", "over limit")
  - Visual indicator (circular progress ring showing budget usage)

**Functionality:**

1. **Mock API Integration**
   - Create a mock API service that returns the provided JSON response
   - Parse the JSON into proper Dart data models
   - Use Future/async-await patterns appropriately
   - Add realistic network delay (e.g., 500-1000ms) to simulate API call
2. **Visual Budget Indicators**
   - Implement color-coded circular progress rings based on `spendStatus`:
     - **UNDER_SPENT**: Green color (under budget)
     - **OVER_THRESHOLD_SPENT**: Yellow/Orange color (approaching limit, 80-100%)
     - **OVER_SPENT**: Red color (exceeded budget)
   - Use `spendPercentage` to calculate the progress ring fill
   - Cap the visual progress at 100% even if percentage exceeds it
3. **Display Logic**

- Show amount spent: `categorySpend` (e.g., "$78.50")
- Display remaining/over budget text:
  - If `spendRemaining > 0`: Show "$X.XX left"
  - If `spendRemaining < 0`: Show "$X.XX over limit"
  - If `spendRemaining == 0 && categorySpend == 0`: Show "$0.00 left"

4. **Category Icons Mapping**
   Map each `finleyCategory` to appropriate icons:
   - `HOUSING_AND_UTILITIES` → House icon
   - `EDUCATION_AND_CHILDCARE` → Graduation cap icon
   - `TRANSPORTATION` → Car icon
   - `HEALTHCARE_AND_MEDICAL` → Hospital/Medical icon
   - `INSURANCE` → Shield icon
   - `LOANS_AND_CREDIT_CARDS` → Credit card icon
   - `GROCERIES` → Shopping cart icon
   - `ENTERTAINMENT` → Balloon/Entertainment icon
   - `DINING_OUT` → Fork & knife icon
   - `SHOPPING` → Shopping bag icon
   - `MISCELLANEOUS` → Gear/Settings icon
   - `TRAVEL` → Airplane icon
5. **Interactive Categories**
   - Tapping a category card should show a bottom sheet or dialog
   - Display detailed breakdown:
     - Category name
     - Amount spent
     - Budget limit (calculated from `spendRemaining + categorySpend`)
     - Percentage spent

    - Status message
  - Include slide-up animation for bottom sheet
6. **Bottom Navigation**
  - Implement bottom navigation with 4 tabs
  - Highlight "Spending" tab as active (green color)
  - Other tabs clickable but show simple placeholder screens or snackbar
7. **Out of Scope**
  - Lock, copy, and share buttons (non-functional)
  - Notification bell (non-functional)
  - Profile icon (non-functional)
  - The green "Keep this empty" box functionality

## Evaluation Criteria

- Accurate JSON parsing and data modeling
- Grid layout implementation and responsiveness
- Custom widget creation and reusability
- Responsive to different device sizes
- Color logic based on spending status
- Error handling in API layer
- Animation quality

## Deliverables

- Complete Flutter project that runs on iOS/Android
- README with:
  - Setup instructions
  - Architecture and state management approach
  - How the JSON data is being used

    - Any assumptions made
- Mock API service with the provided JSON response

## Design Specifications

- **Spacing**: Use 8px, 16px, 24px grid system
- **Colors**:
    - Green (under budget): `#00C853`
    - Yellow (threshold): `#FFB300`
    - Red (over budget): `#E53935`
    - Background: `#F5F5F5`
    - Card background: `#FFFFFF`
- **Typography**:
    - Category name: 12-14px, medium weight
    - Amount: 14-16px, bold
    - Status text: 10-12px, regular
- **Grid**: 4 columns with equal spacing

## Provided Data

Use the attached `Sample API Response.json` file as your mock API response. The candidate should:

1. Parse this JSON into their data models
2. Handle all spending statuses correctly
3. Display the data accurately in the UI
4. Properly calculate and visualize the spending percentages

Sample API Response.json