Week1 App Proposal

Submission 1 **: API Selection**

[Selected API] The Metropolitan Museum of Art Collection API
[Endpoint] https://metmuseum.github.io/
[Summary] A Met Explorer that lets users browse museum collections by department/culture/era, and create personal digital art galleries for research/themed collections of educational use, visit planning, social sharing, and etc purposes.

Submission 2 : **Design**

Persistence

- Store an array of artwork objectID values locally
- Re-fetch complete artwork data using stored IDs (via /objects/{id} endpoint)

UI Basic

Tab1 - Browse

Stack1A: Department List

- Data source: `GET /departments`
- Displayed fields: `displayName` (department title)
- User action: Select department → Stack1B

Stack1B: Department Artworks

- Data source: `GET /objects?departmentIds={selectedId}`
- Displayed fields:
  - `title` (artwork name)
  - `primaryImageSmall` (thumbnail)
  - `artistDisplayName` (artist)
- User action:
  - Search: Client-side filtering by search some keywords of `title`
  - Filter: Group target artworks by `culture` field
  - Tap artwork → Stack1C

Stack1C: Artwork Detail

- Data source: `GET /objects/{selectedObjectID}`
- Displayed fields:
  - `primaryImage` (high-res image)
  - `objectDate` (creation period)

    - `title + artistDisplayName`(Title & artist)
    - `medium` (materials)
- User action:

  Smart Favorite Button:

    - Default State (Not collected):
        - Button label: "Add to Collection";
        - Single Tap: Immediately launches tag selector
    - Collected State:
        - Button shows active tag emoji (e.g., ❤️)
        - Single Tap: Opens tag management panel

  Tag Selection Dialog :

    - Last-used tags appear first ;
    - "Custom..." opens text input for new tags

Tab2 - My Collection

Stack2A: Favorites List

- Data source:
    - locally stored `objectID` array
    - Call `GET /objects/{id}` for each ID to get updated data
- Displayed fields:
    - `primaryImageSmall` (thumbnail)
    - `title` (artwork name)
    - `artistDisplayName + objectDate` (secondary text)
    - `Active tag emoji` (e.g., ❤️`) right-aligned`
- User actions:
    - Pull-to-refresh: Re-fetch all artwork data
    - Tap artwork → Stack2B (Detail View with edit mode)
    - Swipe left:
        - Delete option (removes `objectID` from local storage)
        - Tag edit option (opens Tag Selection Dialog)

Stack2B: Collection Detail (Edit Mode)

- Enhanced features vs Stack1C:
    - Tag management bar:
        - Shows all assigned tags
        - Tap any tag to:
            1. Remove from current artwork
            2. Filter all artworks with same tag (→ Stack2C)
    - Metadata comparison:

        - "Last updated: [timestamp]" from API response header
    - Context menu:
        - "Suggest similar" (uses `culture/medium` fields)

Stack2C: Tag Filter View

- *Trigger*: Selecting tag in Stack2A/2B
- *Data flow*:
    - Filter local `objectID` array by tag
    - Call `GET /objects/{id}` for each filtered ID
- *Display*: List view with `primaryImageSmall` + `title`

Planned Feature: Share collection via link/image export