Final Project _ Inspiring Your Day

Objective

Create a Python program that:

- Takes user input (mood, task, goals)
- Picks or generates a meaningful phrase
- Displays that phrase with a related image to inspire the user

User Experience Flow

1. Input Section
   A. Text fields (or dropdowns) for:
      i. Mood: Happy, Anxious, Tired, Motivated…
      ii. Major Task: Study, Meeting, Workout…
      iii. Wannabe: Leader, Creator, Peaceful, Energetic…
2. Generate Button
   A. When clicked, the program processes input and generate output
3. Output Section
   A. Display:
      i. An inspirational quote (filtered or generated)
      ii. A related image (keyword-based match)

Core Features

- Load a dataset of quotes
- Filter or score based on user input conditions
  → **What logic should be used to generate the phrase uniquely?**
- Display a matching image (simplified: use a fixed image library or keyword-image matching)
  → **How to create the related image in user's mind more impressable?**
- Basic UI using tkinter or graphics.py

Decomposition

To start, I split the program into the following components:

Phase 1
get_user_input()

- Purpose: Prompt the user for three inputs that represent their current state, immediate focus, and long-term aspiration
  - Mood: "How are you feeling today?"
  - Task: "What is your main task or responsibility for today?"
  - Wannabe: "What kind of person do you aspire to become?"
- Issue: User input might be vague, inconsistent, or irrelevant.
- Solution: Use predefined category keywords and a scoring function to map input text to the closest category
  - Keywords
    - Mood → ["calm", "anxious", "motivated", "tired"]
    - Task → ["focus", "meeting", "exercise", "creative work"]
    - Goal → ["leader", "creator", "peaceful", "successful"]
  - Scoring _ Fallback Order
    - Step 1. Try exact/synonym match → simple_overlap_score
      - ✓ It is a function to compare the user's free-form input (which can be noisy or vague) with a set of predefined keywords, and return a score indicating how well they align
      - ✓ Process:
        - ✧ Break down user input into words (basic tokenization)
        - ✧ Compare each word to your list of known keyword categories
        - ✧ Score based on how many words overlap
    - Step 2. If no match, try fuzzy matching → Fuzzy Matching (String Similarity)
      - ✓ Measures how similar two strings are in characters, not meaning
      - ✓ Based on Levenshtein distance: how many changes (insertions, deletions, substitutions) are needed to turn one string into another

- ✓ Strength
  - ✧ Simple and fast
  - ✧ Great for Typos, Slight phrasing differences, Synonym spelling issues
- ✓ Limitations
  - ✧ No understanding of meaning
  - ✧ Only useful when user input contains similar-looking words
  - ✧ Cannot handle abstract or expressive sentences

◆ Step 3. Always return a default fallback

Phase 2

Pool of Quotes & Images

- Quote Pool _ quotes.json
- URL-Based Images (Better for portability) _ images.json with URLs

Phase 3

Objective

- Match user input → best keyword tags (done in Phase 1)
- Use those tags to:
  - ➢ Select a relevant quote from quotes.json
  - ➢ Select a matching image URL from images.json
- Display them together using a simple GUI

Flow

- Step 1. Load the Quote Pool
- Step 2. Select the Best Quote based on Tags
- Step 3. Load the Image Pool (URL-based)
- Step 4. Select Image by Mood Tag
- Step 5. Simple Display with tkinter

load_quote_dataset() & select_relevant_quote(user_input, quotes)

- Purpose: Load a dataset of meaningful phrases and select one that best fits the user's input
- Rule-based Matching _ Static matching from a preloaded quote dataset (faster, predictable)
  - ➢ Load quotes with metadata tags like ["calm", "focus", "ambition"]
  - ➢ Score quotes based on keyword overlap with mood, task, goal
  - ➢ Select the highest-matching one

Load_image() & match_image(user_input)

- Purpose: Pair the selected phrase with a visually relevant image to 'mood'
- Pre-matched Local Images _ Local pre-tagged image folder (simple & safe)
  - ➢ Use keyword folders like /images/focus/, /images/peace/
  - ➢ Match mood or phrase tag to folder

[4] display_output(quote, image)

- Use tkinter or graphics.py to:
  - ➢ Full-screen canvas or window
  - ➢ Display the quote in large, readable font
  - ➢ Show the image (PNG/JPG) neatly (background or beside)
  - ➢ Regenerate button to try again
  - ➢ "Save this" to screenshot/record quote
- Design Focus: Minimal, warm tone, emotionally engaging layout