FindMyPhotos! Product Requirements Document

AI Career Boost Capstone Project
Contact: vicky.mccann@gmail.com

Context & Rationale
It is difficult and time-consuming to find specific and related photos in a sea of folders on a computer. As storage increases on smart phones, users take more and more photos without needing to clear up space. This results in hundreds, if not thousands, of photos stored in a multitude of folders when uploaded onto a computer, making it time-consuming to locate desired photos.

Goals & Success Metrics
FindMyPhotos! will tag photos using AI, allowing the user to quickly locate photos from folders using specified filters. This saves the user time by not having to go into each and every folder looking for desired characteristics, but also reduces accidentally missing photos or skipping directories.

Use Case(s)
Vicky wants to find photos of her dog, Nugget, that could be useful for Harsha's AI Art Therapy project. Unfortunately, she has years of photos from her phone uploaded into hundreds of folders on her computer. Going through each folder looking for Nugget pictures is very time consuming, and in her haste, she may miss some relevant photos.

Edwin wants to find all pictures on his computer that include his parents, brother, and nieces so that he can choose photos for a photo album to bring to his family this summer. He has hundreds of folders containing images downloaded from his phone, iPad, Facebook, and WhatsApp. He wants a quick way to locate all relevant pictures from the many directories on his computer, and tag them for quick retrieval.

Requirements
- Create an easy-to-use UI for users to interact with photo tagging and viewing tasks
    - One tab for filtering and viewing (FindMyPhotos)
    - One tab for tagging tasks (TagMyPhotos)
    - One tab for Settings
        - To point to folder(s) containing a photo library
        - To set default Viewing and Sorting options, include a max # of images to display per page
    - An "About" button will describe this app as well as provide the app version
    - A "Help" chatbot will always be available on the bottom right corner of the UI
- Filtering:
    - By default, all images in the Library setting are available for viewing (no filtering)
    - Allow for comma-separated user input in a text box to be applied as filters
    - The collapsible "Additional Filters" section allows for quick filters based on Date Range, Location, Favorites, and other common filters
    - Display thumbnails of matching photos in the viewer
- Tagging:
    - AI Tag (recommend as a first pass)
    - Edit Tags
        - Edit tag names
        - Remove unwanted tags
        - Add additional tags

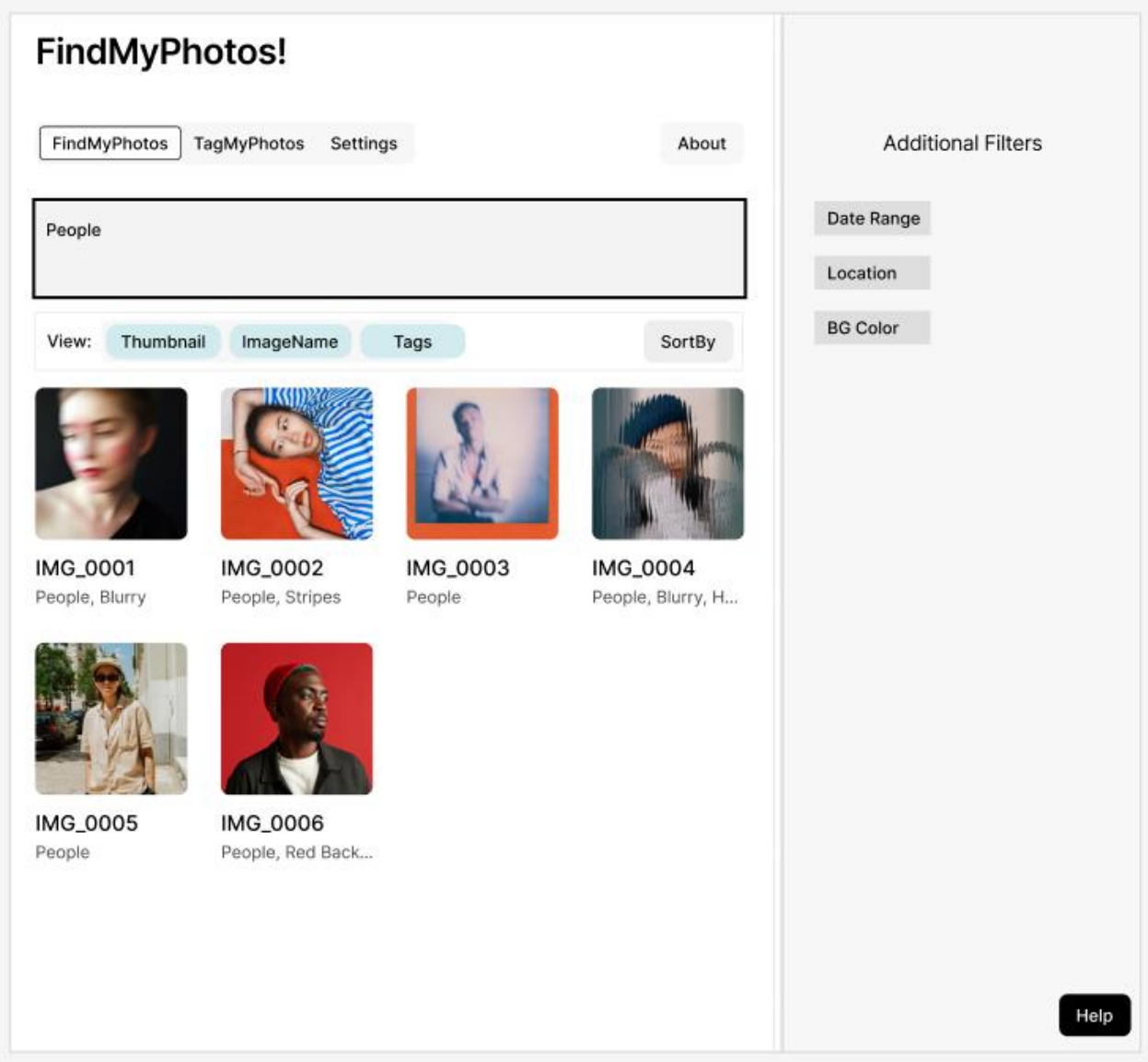
Front-end

- UI Design
  This tool would be a desktop application with a sleek, clean design. Tabs at the top will determine the function being performed – Finding/Viewing photos, Tagging photos, or editing the Settings. The Filter Box allows the user to type in custom filters on which to find matching tags.

  The FindMyPhotos tab will display a thumbnail view of all photos in the specified library path that contains the requested tags. This section also includes buttons for toggling the view of the image thumbnail, image name, and associated tags, as well as the ability to sort the view (a pulldown for sorting by image name or timestamp). The right side of the UI consists of a collapsible "Additional Filters" section where the user can quickly set a date range or location for the search filter.

- Include a Help Chatbot that can answer usage questions and provide a link to the Help documentation

- Sample UI:



API

- Connection to AI for tagging functionality
- Store tagging information
    - Edit EXIF data to include tagging information
    - Index search for faster results (how to do this?)
- Use user Filter input to view matching photos

Future Enhancements:

- Selection:
    - Allow user to select photos from the viewing window and perform specific actions:
        - Favorite selected photos (can then be used in additional filtering)
        - Copy/Move photos to a new folder (file folder integration)
        - Zip and save to folder (file folder integration)
        - Zip and send in an email
        - Delete (WARN, then move files to trash)
        - Edit tags
    - Select ALL
    - Select NONE

Data Sources

- Library of photos on local drive

Dependencies

- Python3.12
- Yolo
- PyTorch
- Cuda