GTU CSE 495 GRADUATION PROJECT

FINAL PRESENTATION

IMAGE RECOMMENDER

ÇAĞRI ÇAYCI

ADVISOR: Dr. GÖKHAN KAYA

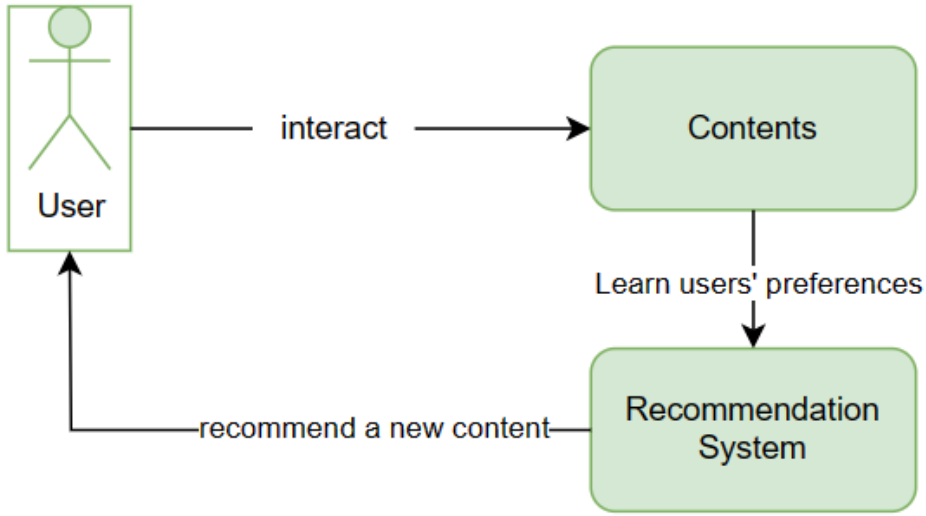
NOVEMBER 2023

Contents

- Introduction
- Design Plan
- Completed Tasks
- Project Timeline
- Success Criteria
- Sources

Introduction

Recommendation systems are a type of software application designed to offer personalized suggestions to users. These suggestions are tailored based on the users' past interactions with the system. Such systems find widespread use across various online platforms, aiming to enhance the user experience by minimizing the effort and time users spend searching for content they are interested in.



Introduction

Image recommendation systems are a specialized type of recommendation system that specifically focuses on providing suggestions for images. One of the techniques employed in image recommendation is tag-based recommendation, where tags associated with images play a crucial role in generating personalized suggestions for users. This approach leverages the descriptive tags attached to images to understand user preferences and offer relevant recommendations.

Introduction

Therefore, this project is based on two main assumptions:

- **User Preference for Similar Images:** Users are more likely to enjoy and engage with images that share visual similarities.
- **Semantic Similarity through Labeling:** Images with similar labels convey semantic similarities, aligning with user expectations.

These assumptions guide our system's design, emphasizing personalized recommendations and enhanced relevance. As we progress, we will explore how these assumptions contribute to the user experience and overall system performance.

Design Plan

This project was deployed as five stages.

- Collecting Unique, Proper, Uniformly Distributed Images
- Labelling and Preprocessing Images
- Setting Dataset Structure
- Developing Recommendation Algorithm
- Developing Web Application

Design Plan

1. Collecting Unique, Proper, Uniformly Distributed Images

To collect images, the project utilized the API Ninjas and ImageCDN APIs. After acquiring 35,000 images, the dataset was inspected for identical images, and duplicates were removed, resulting in a dataset with 9,000 unique images. In an effort to increase the dataset size, the Google Images Search API was employed, utilizing random labels. Subsequently, the collected images underwent manual inspection to ensure their appropriateness for a working environment.

Design Plan

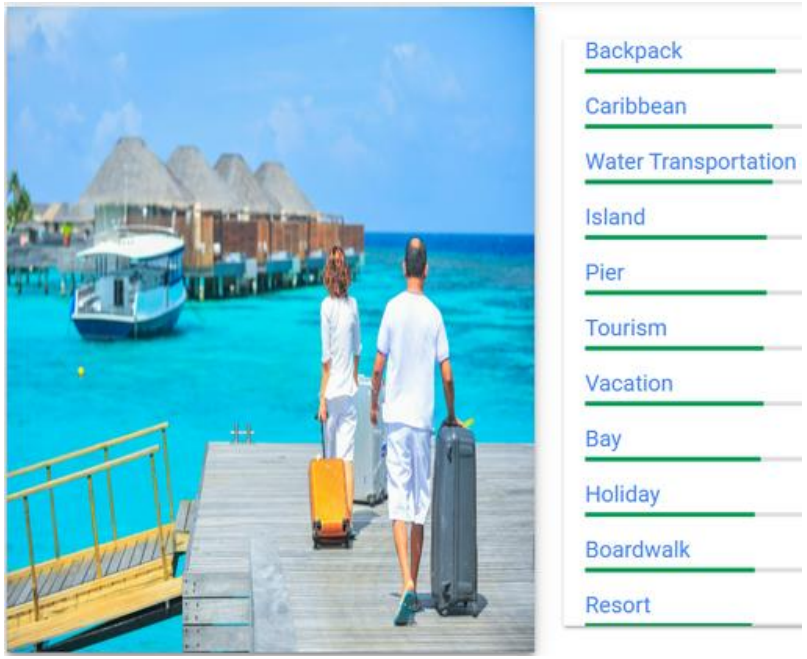
2. Labelling and Preprocessing Images

For labeling images, the project utilized the Google Vision API. The choice of Google Vision API was motivated by its capability to provide both abstract (content-related) and concrete (object-related) labels, adding a level of sophistication to the labeling process.

Design Plan

2. Labelling and Preprocessing Images

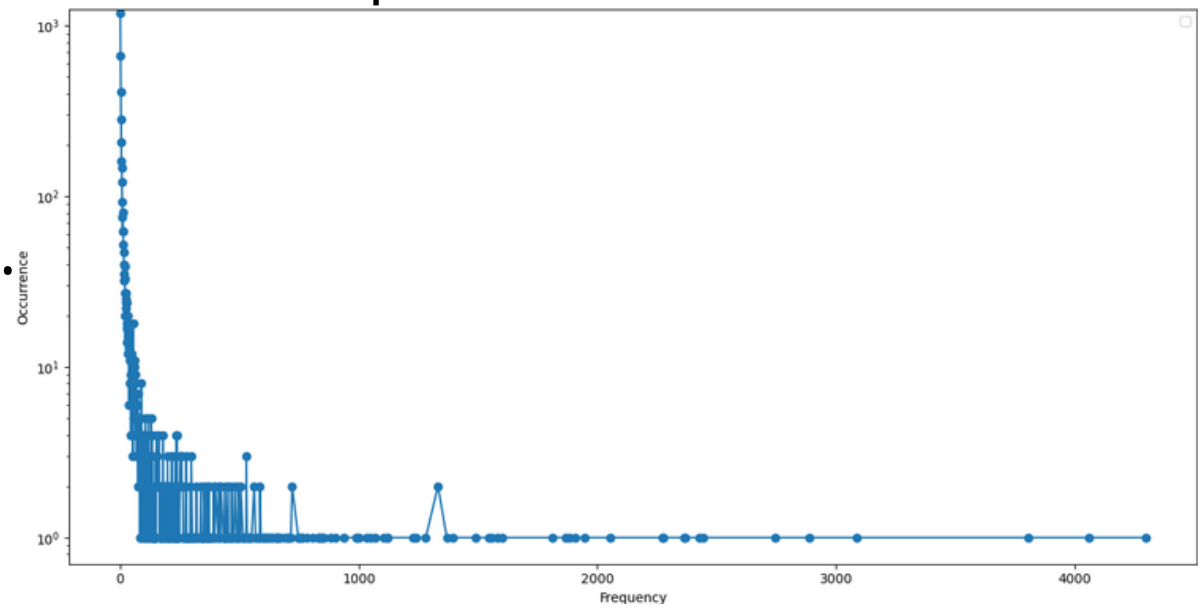
An example from the Google Vision API labeling demonstrates that some labels are object-related, like "Backpack" and "Resort," while others are content-related, such as "Holiday," "Tourism," and "Vacation." This dual perspective enriches the labeling with a broader understanding of image content.



Design Plan

2. Labelling and Preprocessing Images

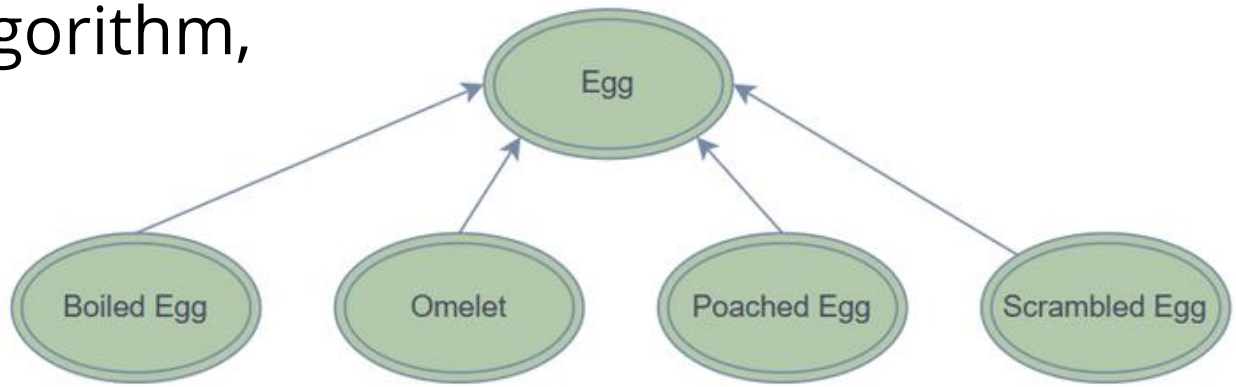
During the preprocessing of images and labels, the statistics of labels were calculated, exposing a challenge related to very rare and very frequent labels. These extremes had the potential to negatively impact the recommendation algorithm. To tackle this issue, two potential solutions were experimented with: hierarchical clustering and setting boundaries.



Design Plan

2.1. Hierarchical Clustering

Hierarchical clustering was implemented on labels by converting them into vectors using the spaCy library and measuring the distance between labels. The merging process involved pairing labels that were the closest vectors in the dataset. The labels within a cluster were summarized using the most frequent label in that cluster. However, it was observed that this approach reduced the resolution of the recommendation algorithm, potentially impacting its effectiveness.

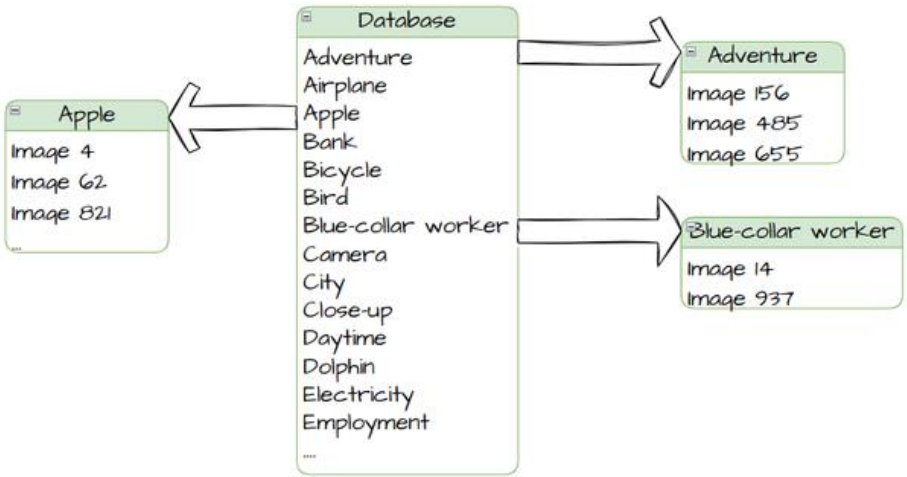


2.2. Setting Boundaries

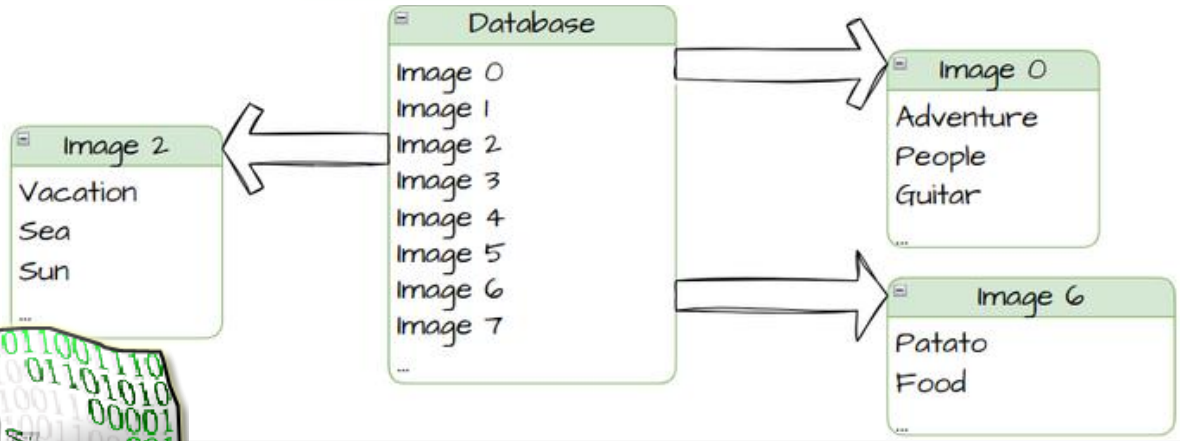
To address the challenge of excessively frequent or rare labels, the optimal solution was identified as removing labels that fell outside specified boundaries. After conducting various experiments with different boundary values, it was determined that a label should appear in at least 5 and at most 500 images in the dataset. Labels beyond these boundaries were removed. Additionally, after removing extreme labels, some images contained very few labels, which could make them less representative. To mitigate this, such images were also removed from the dataset.

3. Setting Dataset Structure

To make recommendation algorithm efficient, the dataset is structured as follows;



All labels contains a link for the images which contains that label. Additionally, all images contains the labels which it contains.



Design Plan

4. Developing Recommendation Algorithm

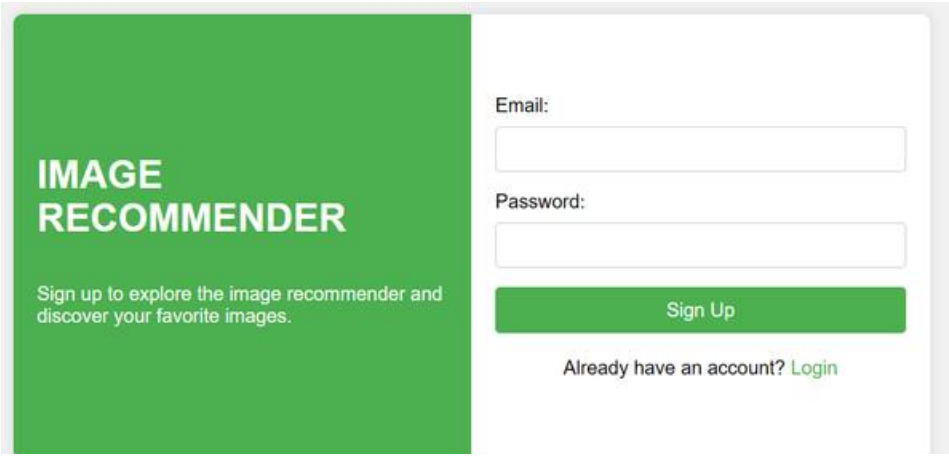
To recommend a new image to a user, a point scoring system is employed. Initially, each user is assigned a vector of labels, with each value in the vector set to 0. This vector is then updated based on the user's interactions with images. Subsequently, the images containing the user's top 10 favorite labels are scored according to the user's vector. Ultimately, the image with the highest score is recommended to the user. This approach ensures that the recommendation aligns with the user's preferences as reflected in their labeled interactions.

5. Developing Web Application

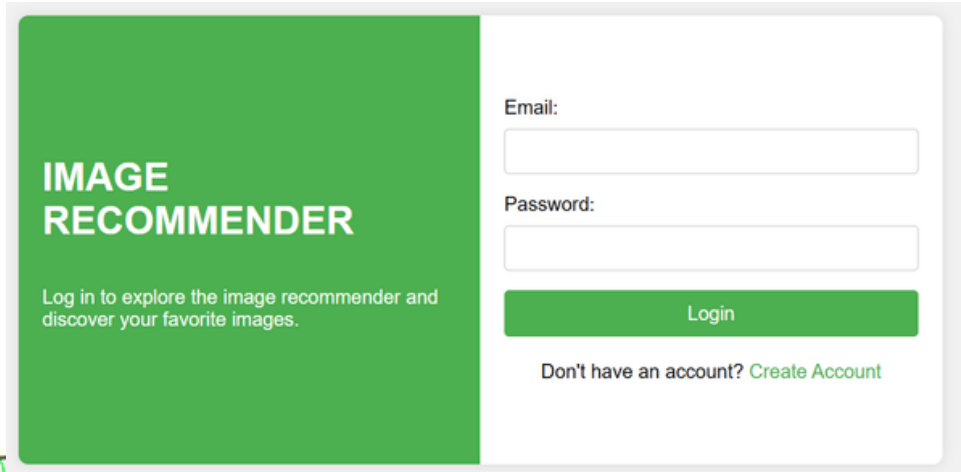
The choice of a web platform for testing the system with real, objective users stems from the general accessibility of web applications compared to other platforms. Given the need for user testing, the web application must also provide authentication for each user. Firebase was utilized for this purpose, serving as the authentication mechanism and storage solution for both images and the dataset. This ensures a secure and reliable environment for user interactions and testing.

Design Plan

5. Developing Web Application



Login Page



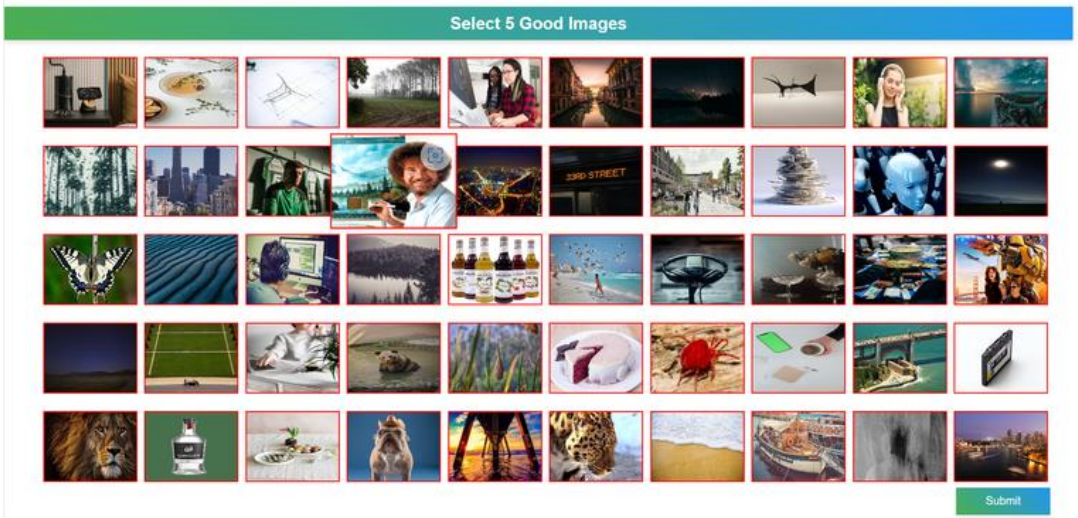
Sign Up Page

The design includes a Login Page and a Sign Up page to facilitate user authentication and access to the system. These pages serve as the entry points for users to log in or create new accounts, ensuring secure and personalized interactions with the recommendation system.

Design Plan

5. Developing Web Application

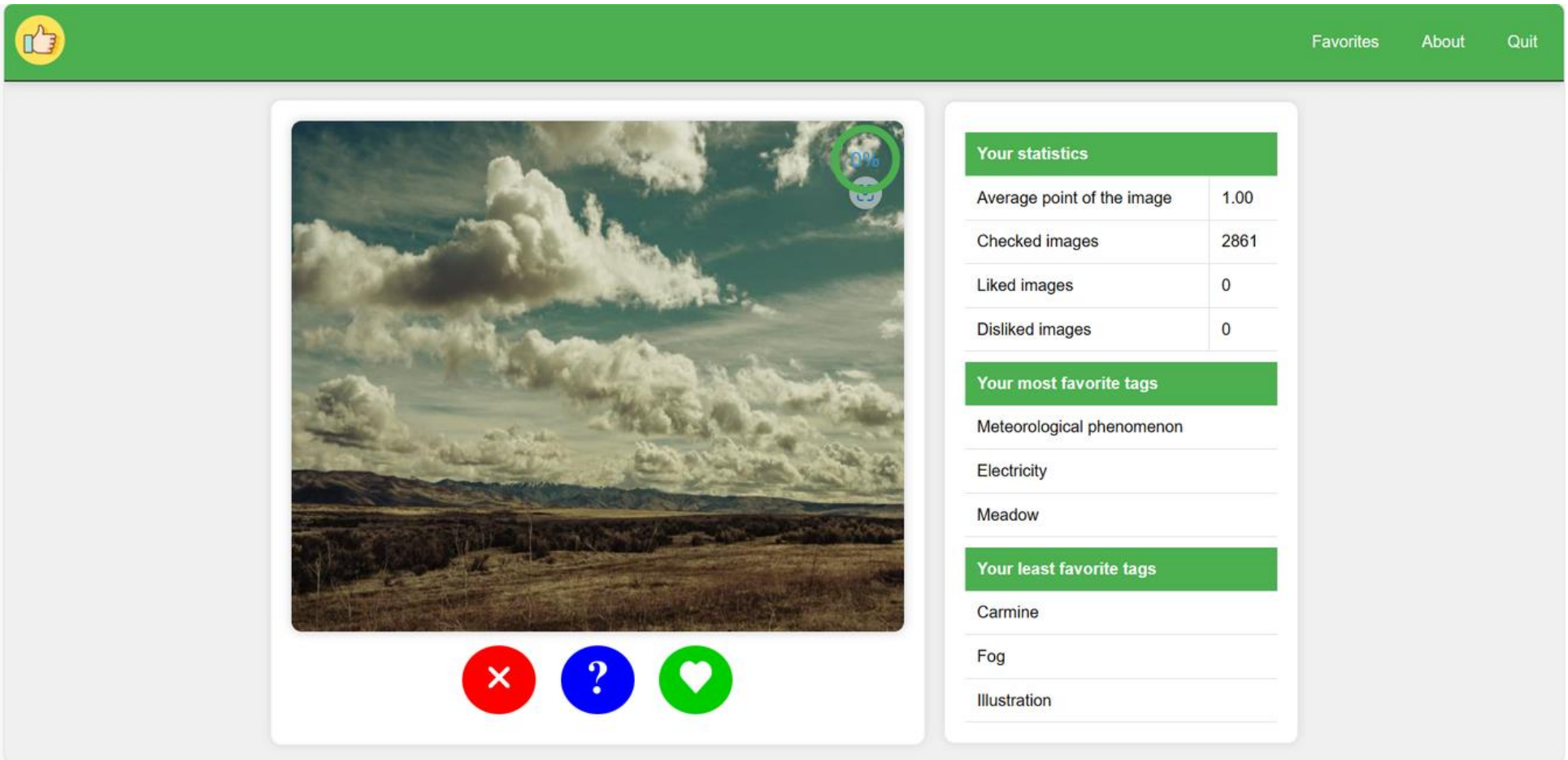
The web application prompts the user to select 5 positive and 5 negative images sequentially. The primary reason for this requirement is to establish a foundation for understanding users' preferences. Upon completion of this process, the web application is now ready to suggest new images to the user based on their interactions and established preferences.



5. Developing Web Application

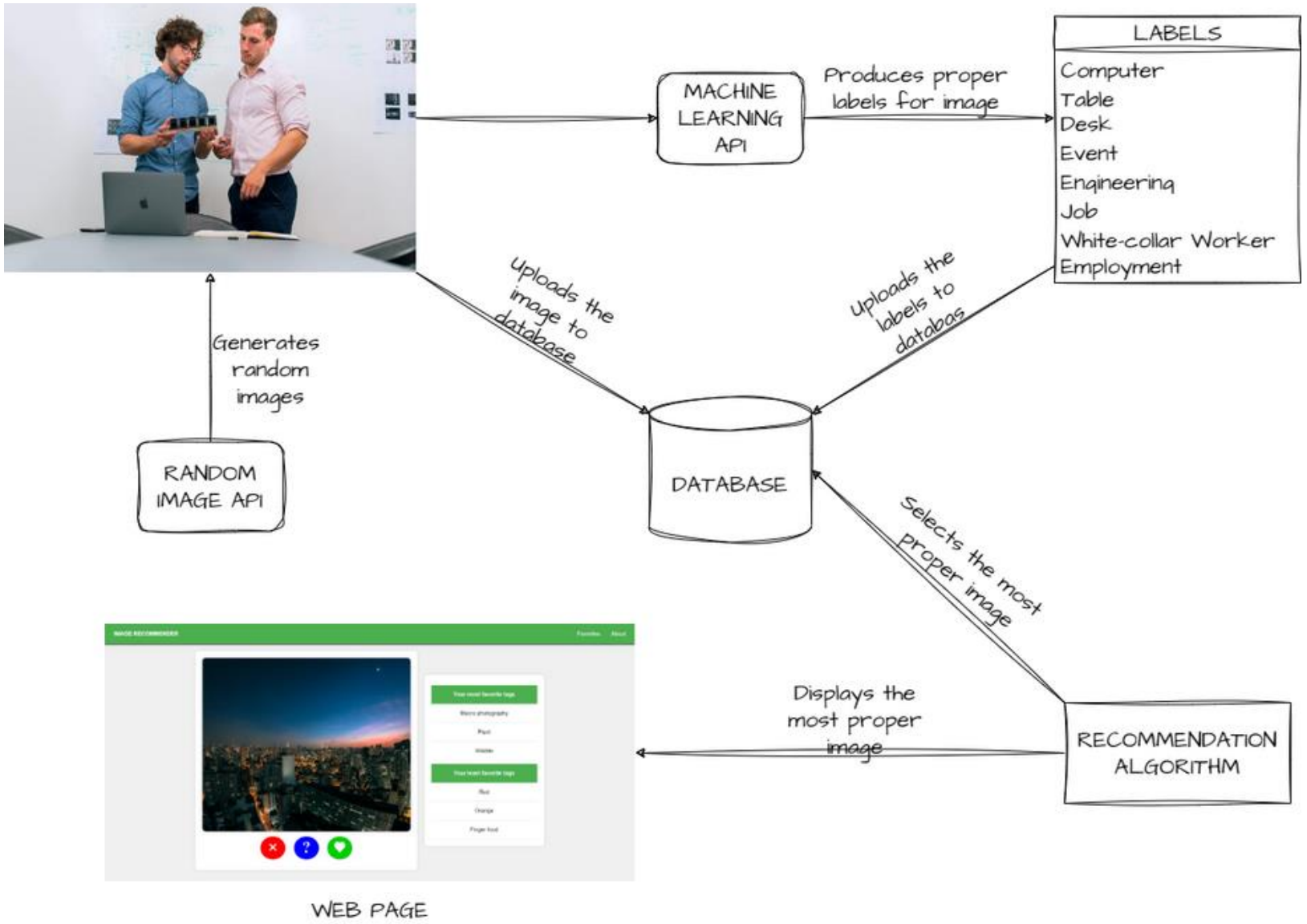
The home page of the web application provides various functionalities, allowing users to like and dislike an image, view the labels associated with the image, and access their statistics. When a user interacts with an image, the system considers both previous and current interactions to promptly recommend a new image. Furthermore, it updates the labels associated with the current image and relevant statistical data. This immediate feedback and dynamic updating contribute to a seamless and engaging user experience.

5. Developing Web Application



Home Page

Design Plan



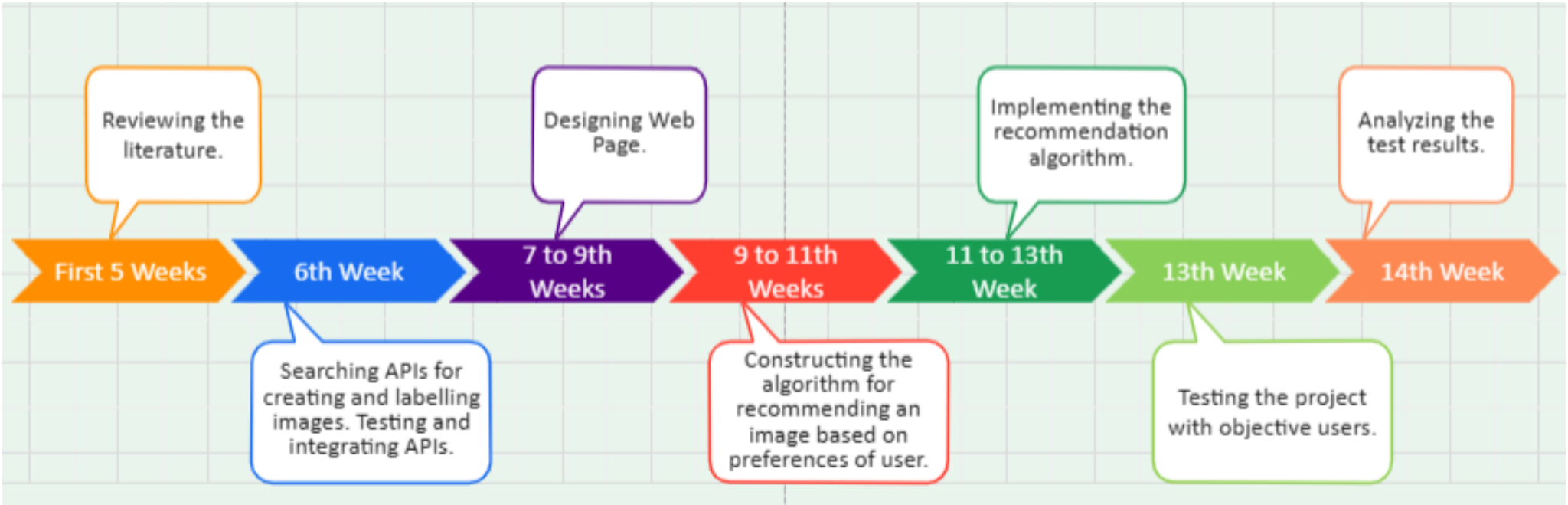
Overall System Design

Completed Tasks

- I've identified suitable APIs for image labeling (Google Cloud Vision) and image creation (API Ninjas).
- I've successfully coded the integration to add labeled images to my database.
- I've populated my database with a variety of images using this code.
- I’ve developed the front-end and back-end part of the my website.
- I’ve developed an image recommendation algorithm.
- The project is tested by objective, real users and the statistics are calculated.

Project Timeline

- The project has reached its current stage exactly as planned on the timeline.
- Documentation is made in 14 to 16th weeks.



Success Criteria

Determined success criteria for this project were;

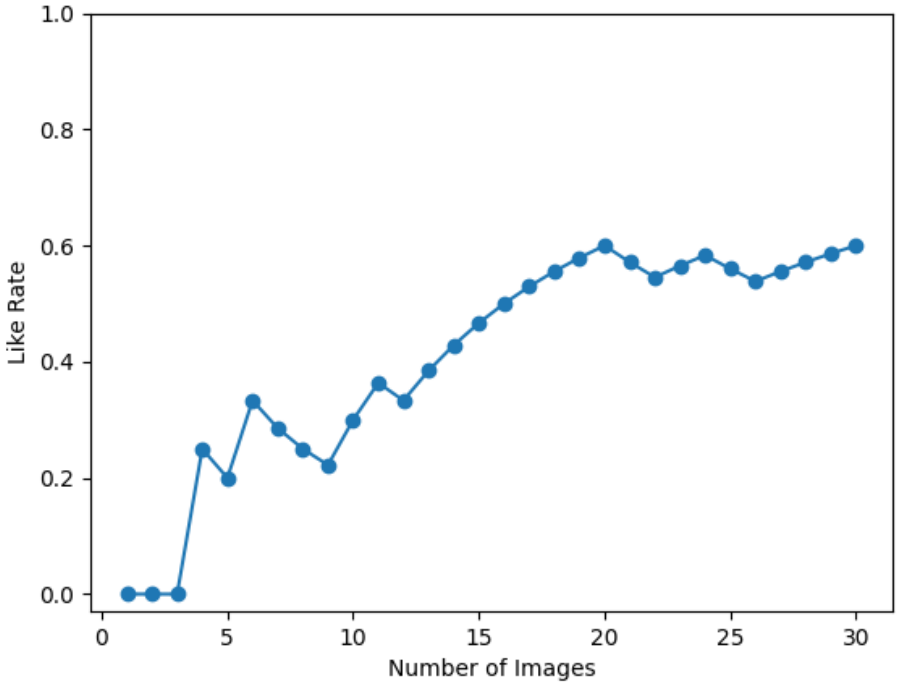
- The system must recommend **different** images from the previous images.
- Like rate must be higher than **50%**.
- Recommending an image must be less than 10 seconds.

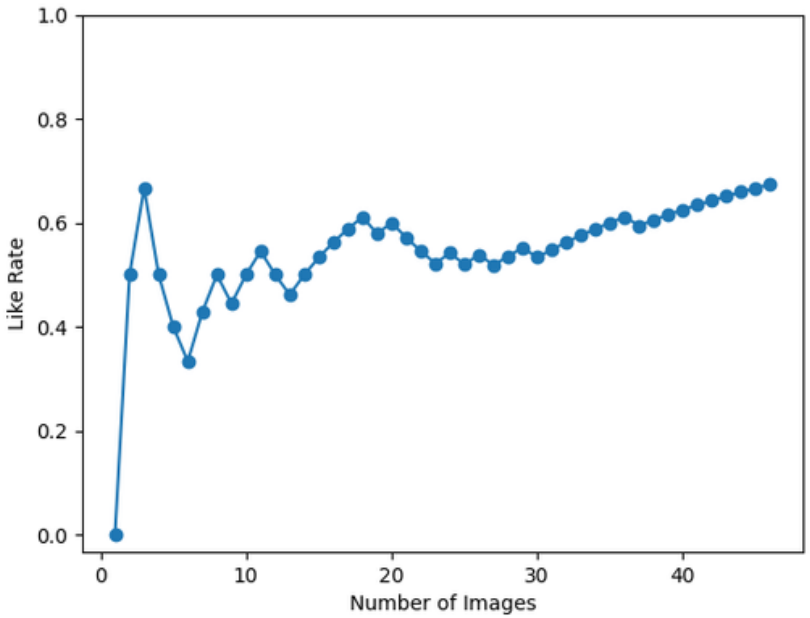
Success Criteria

- The success criteria for the recommendation algorithm were met by adjusting the algorithm to **avoid the repetition of the same image**.
- In testing the system with 20 different users across 42 accounts, the average like rate was calculated and found to be **69.19%**, surpassing the defined success criteria of a like rate higher than 50%.
- Additionally, after conducting experiments, the recommendation time was measured and determined to be **2 seconds**. These results indicate that the system not only meets but exceeds the specified success criteria, demonstrating its effectiveness and efficiency.

Success Criteria

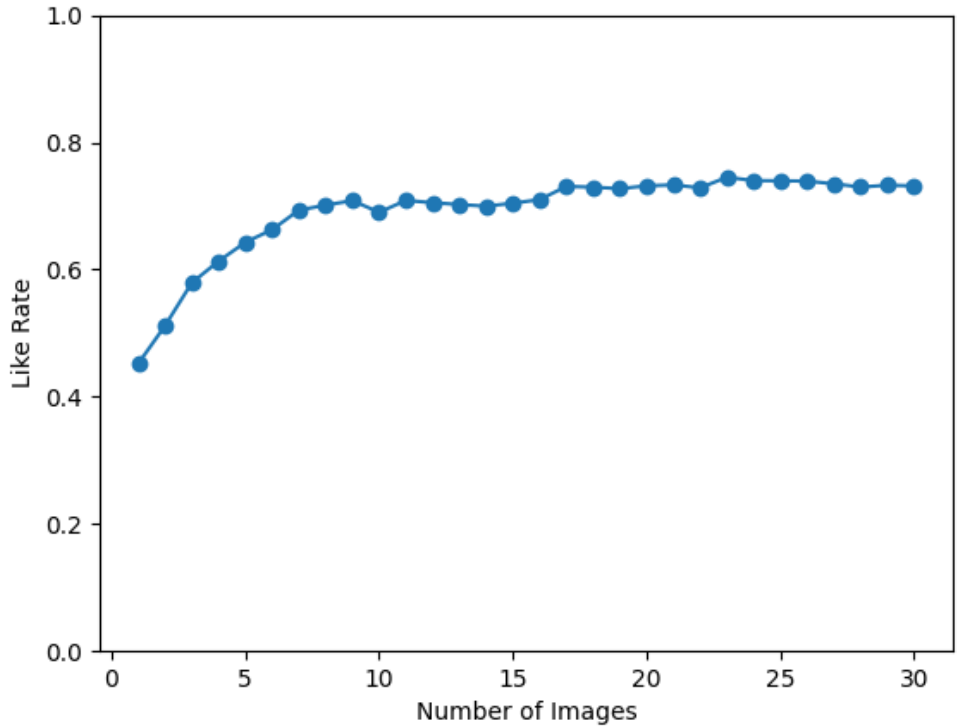
These graphs depict how the like rate of randomly selected users changes based on the number of pictures they engage with.





Success Criteria

These graph depict how the average like rate of users changes based on the number of pictures they engage with.



Sources

- http://draw.io [Drawing system design]
- https://cloud.google.com/vision?hl=tr [Labelling images]
- https://matplotlib.org/ [Drawing Graphs]

THANK YOU FOR LISTENING