DALL-E

Image-Prompt Database

Team 3:

Jalesa Jones,

Adrian Torres,

Shintaro Osuga,

Samantha Zygmont

Syracuse University

IST 659: Data Admin Concepts & Database Management

Professor Gregory Michael Zink

Date: 6/18/24

[Project Overview 3](#_Toc989030813)

[Problem Statement 4](#_Toc512123446)

[Business Utility 4](#_Toc1074132066)

[High Level Business Requirements 5](#_Toc659388993)

[Stakeholder Benefits 5](#_Toc580395369)

[Business Use-Cases and Scenarios 5](#_Toc801177281)

[Narrative and Impact 7](#_Toc1174465747)

[Implementation Considerations 7](#_Toc1813080404)

[Technical Requirements 8](#_Toc1329831233)

[Functional Requirements 9](#_Toc38993838)

[Systems & Tools Involved 9](#_Toc984154109)

[Attributes 9](#_Toc142243517)

[Data Requirements 10](#_Toc1137237143)

[Relationships 11](#_Toc235267151)

[Data & Entity Model 12](#_Toc250188337)

[Logical & Conceptual Model 12](#_Toc347068816)

[Database Normal Form 13](#_Toc605412246)

[Technical Documentation 13](#_Toc2090081453)

[SQL Script 14](#_Toc679995020)

[Python Script 20](#_Toc223800216)

[General Script Use 20](#_Toc437997810)

[Database Connection Script 20](#_Toc2000418996)

[Web App Script 21](#_Toc1078480977)

[App Demonstration 22](#_Toc1576065225)

[Data Question & Answers 25](#_Toc742448621)

[Conclusion & Reflection 33](#_Toc82989766)

[Appendix & Team Meeting Logs 34](#_Toc1005567704)

# Project Overview

## Problem Statement

The DALL-E system is a form of Artificial Intelligence that takes a written prompt and generates an image from it. In its current form, when generating images on the DALL-E system, there is a lack of predictability and consistency between the output images. These inconsistencies have caused prompt engineers to struggle when recreating images in the same style each time. The solution proposed is to create a database application capable of saving prompts and displaying visual cues per keyword or completed prompt. In the application, the keywords stay constant and other variables such as keywords can be varied to create unique images while maintaining art style consistency.

The development of an organizational tool for collecting, storing, and retrieving previously AI (Artificial Intelligence) generated images, with information regarding the prompt utilized to create it would revolutionize the use of the DALL-E system.

# Business Utility

## High Level Business Requirements

1. Provide a user-friendly platform for users to look up and upload AI-generated images along with prompts
2. Allow for community-based sorting and ranking of prompt and image pairs

## Stakeholder Benefits

The stakeholders involved in the project are the technical developers, maintainers of the product, and the project's business managers. There are also stakeholders outside of the project, including but not limited to the businesses which utilize the product either for use in their own work or for connecting to their own business.

## Business Use-Cases and Scenarios

The objective of the DALL-E Image-Prompt Database is to streamline the process of creation, storage, and re-creation of AI images. The process of creating AI generated images is black boxed, with no two same images are created from a single prompt. This from the perspective of the AI engineers may be greatly commendable, in the perspective of the users it is a very limiting part of the process, as replicating images or similar images become very difficult and near impossible without proper care and intent in the prompting. Such is the case where the DALL-E Image-Prompt Database shines the brightest. With the rise in use of AI image generation, a centralized and streamlined process in which users can create an image and then upload it to the database for other users to see, understand, and rate the image and prompt will greatly facilitate re-creation of images. The database’s functionality can be summed up as follows: **Uploading**, **Rating**, and **Re**-**creation**.

**Uploading**

As this product is merely a database infrastructure for facilitation of image re-creation, the images which the database utilizes will be completely up to the users. This is a key part of the process as there must be images to rate and re-create from, but also to allow for more informed decisions about prompting to be made. With the increase in popularity in AI generated art, there will also be an increase in demand for a more streamlined process for uploading and rating these pieces of art. Not only will this alleviate the market need for a central location for AI generated art, but it will further increase popularity, productivity, and efficiency in the creation process of the AI generated art, leading to more intent-full and well-constructed pieces.

**Rating**

As the database is utilized and populated with more images and prompts, there will be a need for organization and differentiation of the images depending on their tags, categories, and their quality. User ratings will be a natural way for images to be quality ensured, as image-prompt quality is not an exact science and therefore cannot be measured using a specific metric. Having a rating will allow users to inform other users of quality prompts for re-creating specific image types.

**Re-creation**

With the many different categorizations, tags, keywords, and ratings, users will be able to easily find, and identify prompts for creating specific images. Just by knowing the types of keywords, phrases, and categories a person has used for a specific image will help future users to create even more quality images. Furthermore, as the database application gains traction, the quality of prompts will increase proportionally. This improvement will occur as prompts which were used to re-create an image can be further rated and filtered, so the most influential phrases are left for creation of specific images.

Use-Case and Scenarios

The use cases for the product include but are not limited to:

1. **Marketing and Advertising:**
   * **Customized Campaigns:** DALL-E 3 can generate custom images tailored to specific marketing campaigns, allowing companies to create highly targeted and visually appealing advertisements. This customization can enhance brand recognition and engagement.
   * **Visual Content Generation:** Marketers can use DALL-E 3 to produce a variety of visual content for social media, websites, and promotional materials, reducing the need for extensive graphic design resources and accelerating content production cycles.
2. **E-commerce and Retail:**
   * **Product Visualization:** E-commerce platforms can leverage DALL-E 3 to generate high-quality images of products from text descriptions, aiding in the creation of virtual catalogs and enhancing online shopping experiences.
   * **Personalized Shopping Experiences:** Retailers can offer personalized product suggestions and visualizations based on user preferences and browsing history, driving sales and customer satisfaction.
3. **Entertainment and Media:**
   * **Content Creation:** Film studios, game developers, and media companies can use DALL-E 3 to generate concept art, storyboards, and promotional materials, streamlining the creative process and reducing production costs.
   * **Interactive Experiences:** DALL-E 3 can be integrated into interactive applications, such as virtual reality (VR) and augmented reality (AR), providing users with unique and immersive experiences.
4. **Interior Design and Real Estate:**
   * **Virtual Staging:** Real estate firms can use DALL-E 3 to create virtual staging images of properties, helping potential buyers visualize furnished spaces without the need for physical staging.
   * **Design Inspiration:** Interior designers can generate visual concepts and mood boards based on client requirements, accelerating the design process and offering diverse creative options.
5. **Education and Training:**
   * **Educational Materials:** Educational institutions can utilize DALL-E 3 to create engaging and illustrative materials for various subjects, enhancing the learning experience.
   * **Simulation and Training:** In fields such as medical training or technical skills development, DALL-E 3 can generate realistic simulations and visual aids, improving training effectiveness.

#### Narrative and Impact

The integration of DALL-E 3 into business processes can drive significant value through enhanced creativity, efficiency, and personalization. By leveraging the power of AI-generated visuals, businesses can transform their operations in several keyways:

1. **Enhanced Creativity and Innovation:**
   * DALL-E 3 democratizes the creative process by enabling individuals without extensive design skills to generate high-quality artwork. This fosters a culture of innovation and experimentation within organizations, encouraging employees to explore new ideas and concepts.
2. **Operational Efficiency:**
   * The ability to quickly generate visual content reduces dependency on traditional graphic design workflows, leading to faster turnaround times and cost savings. Businesses can allocate resources more effectively and focus on strategic initiatives rather than routine content creation.
3. **Personalization at Scale:**
   * DALL-E 3’s capability to produce customized images based on specific inputs allows businesses to deliver highly personalized experiences to their customers. This level of personalization can enhance customer satisfaction, loyalty, and ultimately drive revenue growth.
4. **Competitive Advantage:**
   * Early adopters of DALL-E 3 technology can differentiate themselves in the market by offering unique and innovative visual experiences. This competitive edge can attract new customers and strengthen brand positioning.

#### Implementation Considerations

To effectively integrate DALL-E 3 into business operations, companies should consider the following strategic steps:

1. **Invest in Training and Development:**
   * Ensure that employees are well-versed in using DALL-E 3 and understand its capabilities. Providing training and development programs can maximize the utility of technology and foster a culture of continuous learning.
2. **Develop Robust Data Management Practices:**
   * Implement data management and storage solutions to handle the large volumes of generated images and associated metadata. This will facilitate easy retrieval, analysis, and reuse of visual content.
3. **Collaborate with AI Experts:**
   * Partner with AI specialists and technology providers to customize DALL-E 3 for specific business needs. Collaboration can lead to tailored solutions that align with organizational goals and enhance overall effectiveness.
4. **Monitor and Evaluate Outcomes:**
   * Regularly assess the impact of DALL-E 3 on business performance through metrics such as customer engagement, conversion rates, and operational efficiency. Use insights from these evaluations to refine strategies and optimize the use of AI-generated visuals.

The adoption of DALL-E 3 presents a transformative opportunity for businesses across various sectors. By harnessing the power of AI-generated imagery, companies can drive innovation, enhance customer experiences, and achieve operational excellence. As technology continues to evolve, its potential applications and benefits will only expand, making it an asset for forward-thinking organizations.

# Technical Requirements

## Functional Requirements

1. The ability for users to upload images linked with prompts
2. Retrieve images from the database
3. Users can rate images and prompts

## Systems & Tools Involved

* Python including libraries
  + pandas
  + pyodbc
  + openai
  + os
  + streamlit
  + load\_dotenv
  + logging
* Azure Data Studio
* Azure Blob Storage
* Azure SQL
* SQL ODBC

## Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Attribute Name | Type | Key Type |
| Users | User\_ID | Int, Identity | PK, FK |
| Username | Varchar(50) |  |
| User\_Firstname | Varchar(50) |  |
| User\_Lastname | Varchar(50) |  |
| User\_Email | Varchar(50) | U |
| User\_Registration\_Date | Date |  |
| Prompts | Prompt\_ID | Int | PK |
| Prompt\_Category | Varchar(250) |  |
| Prompt\_Description | Varchar(1000) |  |
| Prompt\_Tags | Varchar(250) |  |
| Prompt\_Creation\_Date | Date |  |
| Prompt\_Creator\_User\_ID | int | FK |
| Images | Image\_ID | Int | PK |
| Image\_Prompt\_ID | Int | FK |
| Image\_Path | Varchar(100) |  |
| User\_Feedback | Feedback\_ID | Int, Identity | PK |
| Feedback\_Image\_ID | Int | FK |
| Feedback\_User\_ID | Int | FK |
| Feedback\_Score | Float |  |
| Feedback\_Comment | Varchar(100) |  |
| Feedback\_Date | Date |  |

## Data Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Entities & Attributes** | | | |
| Entity | Attribute | Props | Description |
| **Users** | User ID | RU | Unique ID associated with each user |
| Username | RU | Name associated with a user’s account |
| Firstname | RC | First name of the user |
| Lastname | RC | Last name of the user |
| Email | RU | Unique email address for the user's account |
| RegistrationDate | R | Date when the user registered for an account |
| **Prompts** | Prompt ID | RU | ID associated with each unique prompt |
| Category | R | The categories associated with the image/prompt |
| Description | R | The description of the prompt that was used to generate the image |
| Tags | R | Keywords or phrases associated with the prompt |
| Creation Date | R | The date that the prompt was used to create the image |
| Creator UserID | R | The User ID of the creator of the image |
| **Images** | ImageID | RU | The ID associated with each image |
| PromptID | R | ID associated with the prompt |
| ImagePath | RU | Path/URL to the image stored in the cloud |
| **UserFeedback** | CreatorUserID | RU | Unique ID associated with each user |
| FeedbackID | RU | ID associated with each feedback |
| ImageID | R | The ID associated with each image |
| UserID | R | ID associated with each user |
| Score | R | User rating of an image/prompt |
| Comment | R | User comment on image/prompt |
| FeedbackDate | R | Date when feedback is made |

## Relationships

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Relationships** | | | | | |
| Relationship | Entity | Rule | Min | Max | Associated Entity |
| Prompts to Images M-M | Prompts | Generates | 1 | M | Images |
| Images | Generated By | 1 | M | Prompts |
|  |  |  |  |  |  |
| Images to Feedback M-M | Images | For | 1 | M | Feedback |
| Feedback | About | 1 | M | Images |
|  |  |  |  |  |  |
| Users to Feedback M-M | Users | Writes | 1 | M | Feedback |
| Feedback | Written By | 1 | M | Users |
|  |  |  |  |  |  |
| Users to Prompts 1-M | Users | Creates | 1 | M | Prompts |
|  | Prompts | Created By | 1 | 1 | Users |
|  |  |  |  |  |  |

## Data & Entity Model

## Logical & Conceptual Model

Logical model:

A diagram of a program

Description automatically generated

Conceptual Model:

A diagram of a program

Description automatically generated

### Database Normal Form

The database is in the third normal form. This can be seen by examining each table. First, the “**Users**” table has a primary key and 5 non-key attributes. This table is in third normal form because only atomic values are in each column, all non-key attributes are fully functionally dependent on the primary key, and no attribute is dependent on another non-key attribute as all 5 non-key attributes are dependent only on the User ID and not another attribute.

Second, the “**Prompts**” table has a primary key, a foreign key, and 4 non-key attributes. The table is in zero normal form as the table contains values for “Category” and “Tags” which are not atomic.

Third, the “**Images**” table has primary key, a foreign key, and a single attribute. This table is also in third normal form as there are only atomic values, the single attribute is fully functionally dependent on the primary key, and the single non-key attribute can only be dependent on the primary key.

Fourth, and lastly, the “**Feedback**” table has a primary key, 2 foreign keys, and 3 attributes. This table is also in third normal form as there are only atomic values contained in each column, the 3 non-key attributes are fully functionally dependent on the primary key, and there are no non-key attributes which are dependent on one another. Therefore, every table except for “**Prompts**” is in third normal form.

# Technical Documentation

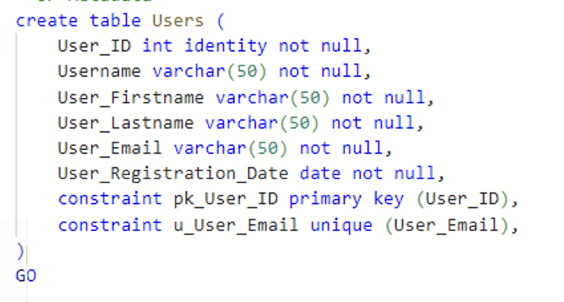
This section covers the technical documentation of the SQL and Python script utilized in creating the database, managing data flow, and the webapp. This section’s main purpose is to aid future developers and maintainers in understanding the database and the internal systems for more efficient transition and continuation of the development.

## SQL Script

**Database structures creation:**

Users Table:

The "Users” table metadata up script is as follows:



Attribute Information:

* User\_ID: A primary key which is a non-null integer identifier for each user which auto-increments for each new user.
* Username: A non-null varchar which is the username for the user, limited to 50 characters.
* User\_Firstname: A non-null varchar which holds the users’ first name information, limited to 50 characters.
* User\_Lastname: A non-null varchar which holds the users’ last name information, limited to 50 characters.
* User\_Email: A unique non-null varchar which holds the users email address used for signing up, limited to 50 characters.
* User\_Registration\_Date: A non-null date value which holds the date when the user signed up.

Prompts Table:

The "Prompts” table metadata up script is as follows:

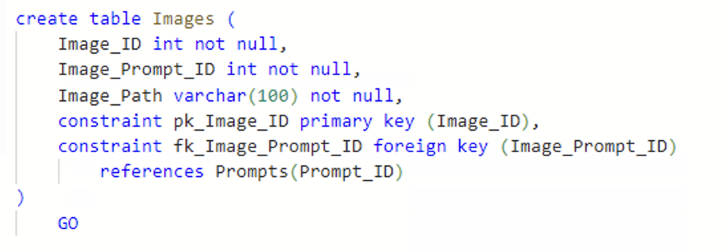


Attribute Information:

* Prompt\_ID: A primary key which is a non-null integer.
* Prompt\_Category: A non-null varchar which holds the prompt category information which is limited to 250 characters.
* Prompt\_Description: A non-null varchar which holds the description of corresponding prompt which is limited to 1000 characters.
* Prompt\_Tags: A non-null varchar which holds the “Tag” information for the prompt which is like “Prompt\_Category” but is for further differentiation between the images. This is limited to 250 characters.
* Prompt\_Creation\_Date: A non-null date value which holds the date information of when the prompt of uploaded.
* Prompt\_Creator\_User\_ID: A non-null integer value which is a foreign key referencing the attribute “User\_ID” in the “Users” table.

Images Table:

The "Images” table metadata up script is as follows:

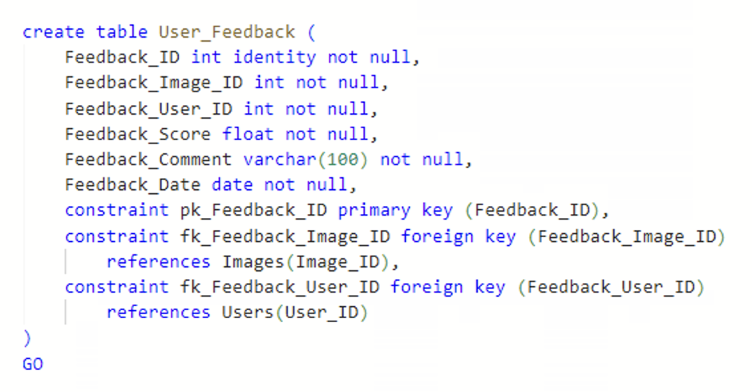


Attribute Information:

* Image\_ID: A non-null integer primary key, which is the associated “ID” value for each image.
* Image\_Prompt\_ID: A non-null integer value which is a foreign key referencing the “Prompt\_ID” attribute from the “Prompts” table.
* Image\_Path: A non-null varchar which holds the link address to the image stored in blob cloud storage. Limited to 100 characters.

User\_Feedback:

The "User\_Feedback” table metadata up script is as follows:



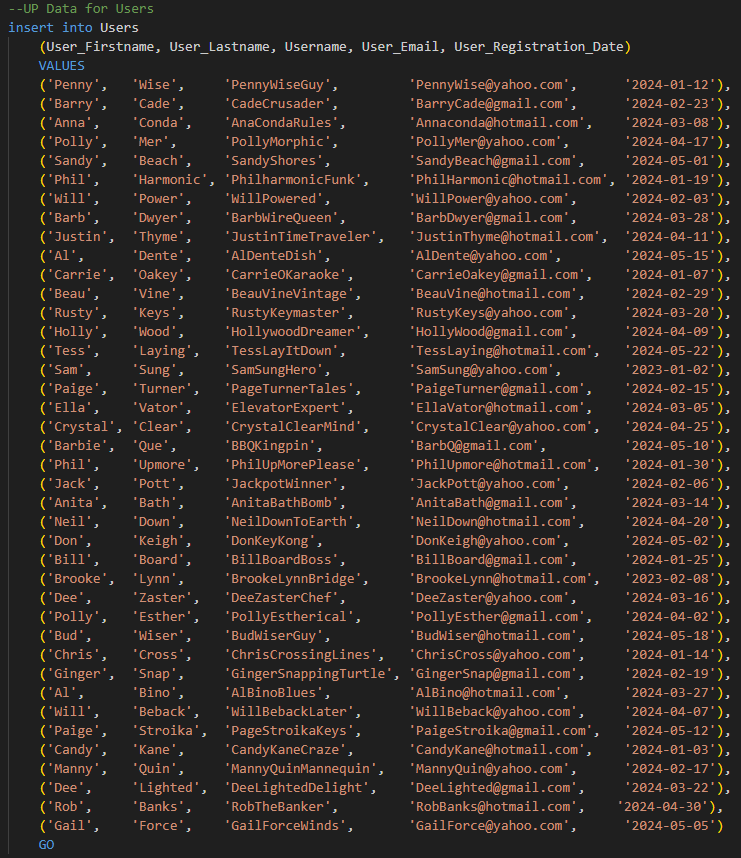
Attribute Information:

* Feedback\_ID: A non-null integer identity primary key which is an auto-incrementing integer value associated with each Feedback.
* Feedback\_Image\_ID: A non-null integer value which is the image id value that is a foreign key referencing the “Image\_ID” attribute in the “Images” table.
* Feedback\_User\_ID: A non-null integer value which is the ID of the user which made the feedback. This is a foreign key which references the “User\_ID” attribute in the “Users” table.
* Feedback\_Score: A non-null float variable which is the rating/score the user gave to the image-prompt. This is limited to floating point number.
* Feedback\_Comment: A non-null varchar which is a comment a user leaves on the image-prompt. This is limited to 100 characters.
* Feedback\_Date: A non-null date value which holds the date information for when the feedback on the image-prompt was made.

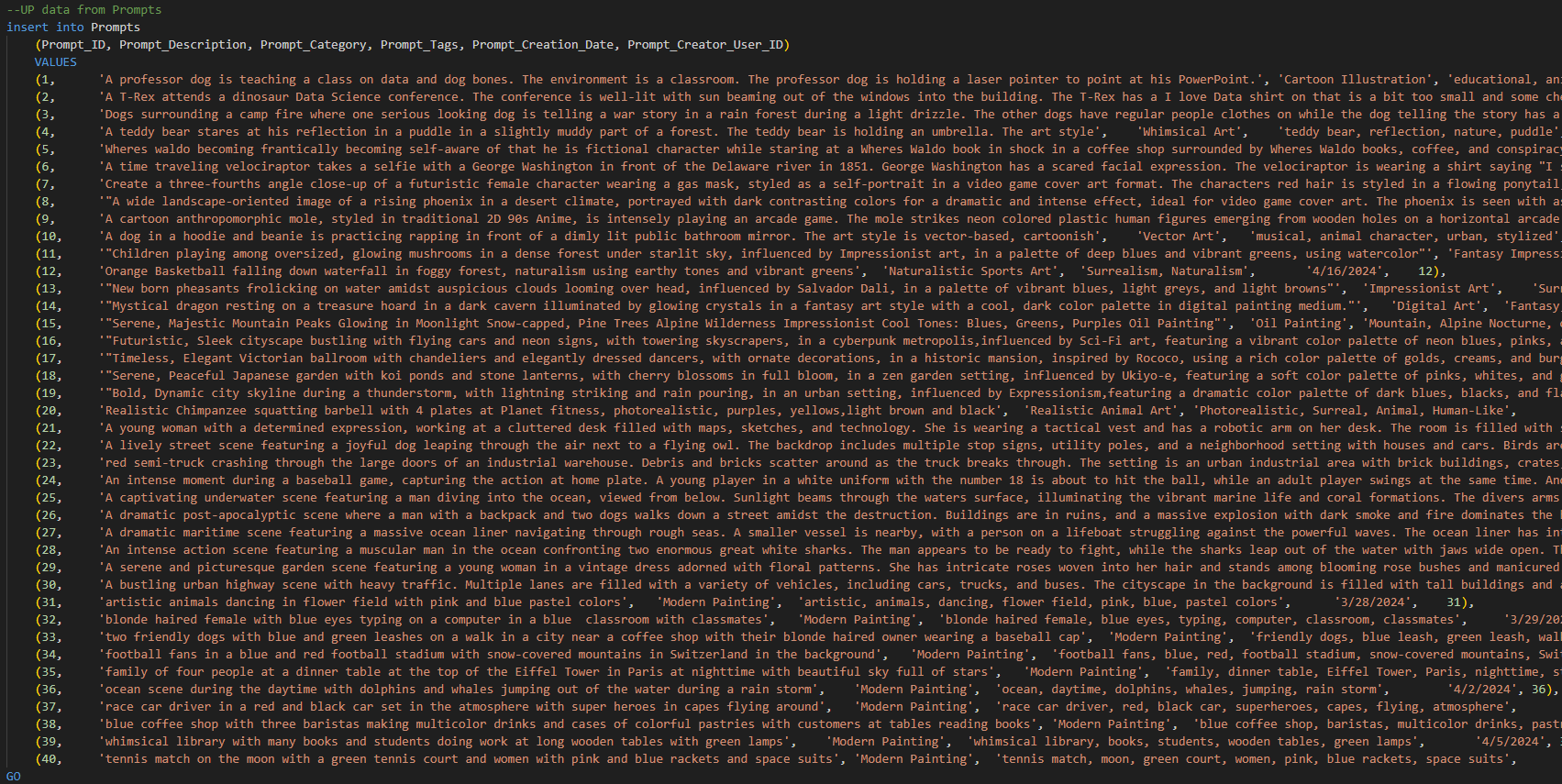
**Data Up Script:**

Up Script:

Users Table Data Up Script



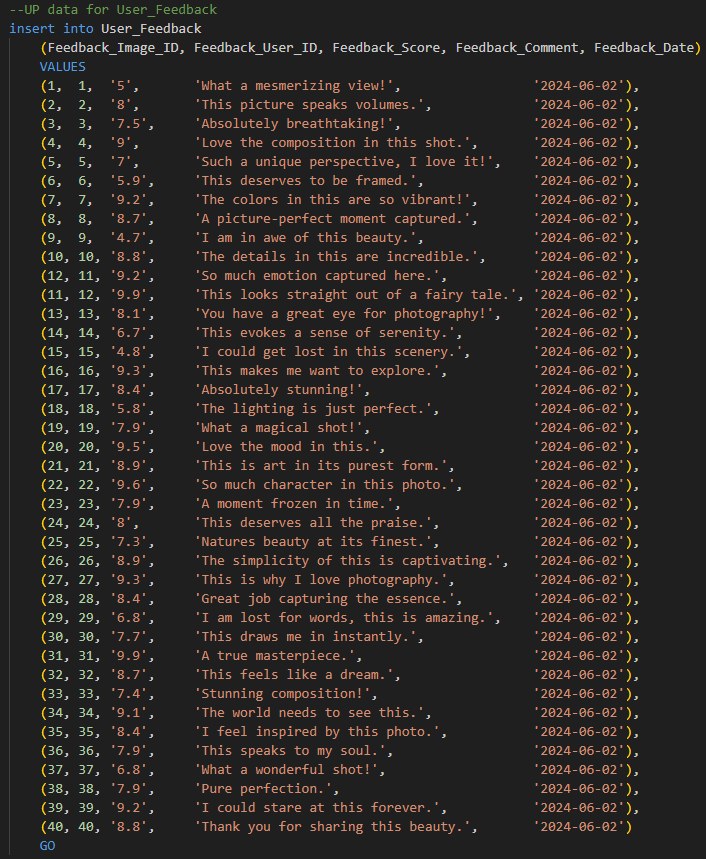
Prompts Table Data Up Script



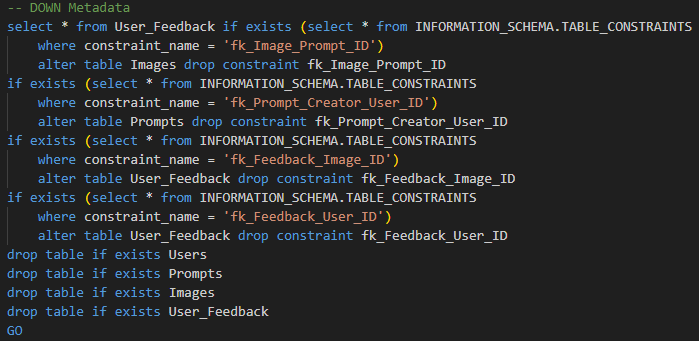
Images Table Data Up Script

Images Table Up Script


User\_Feedback Table Data Up Script



**Down Script:**



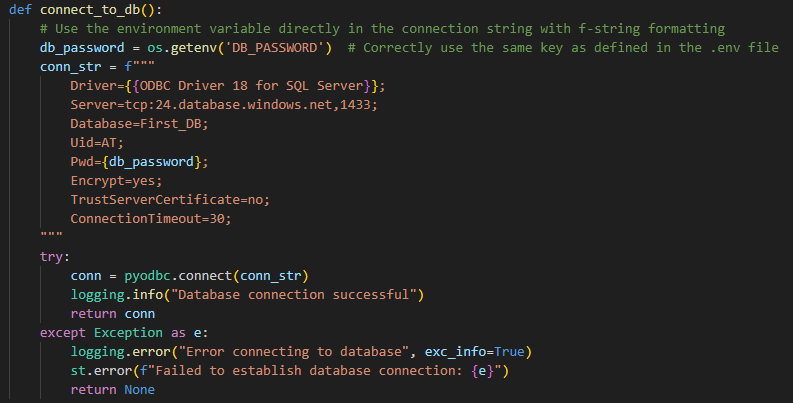
## Python Script

### General Script Use

The Python script in general is used to make the webapp which will host and is a sample use case of a graphical user interface (GUI) for the database. Though the main purpose of the python script is to facilitate and intermediate the connection to the database and the webapp, a few more functionalities are built in to showcase the ability and the true use of the database, facilitation of AI image generation.

### Database Connection Script

The connection to the database is done through a library called "pyodbc”, which allows for a simple and intuitive way to connect to a database using a few parameters like server address, database name, encryption type, and of course password.

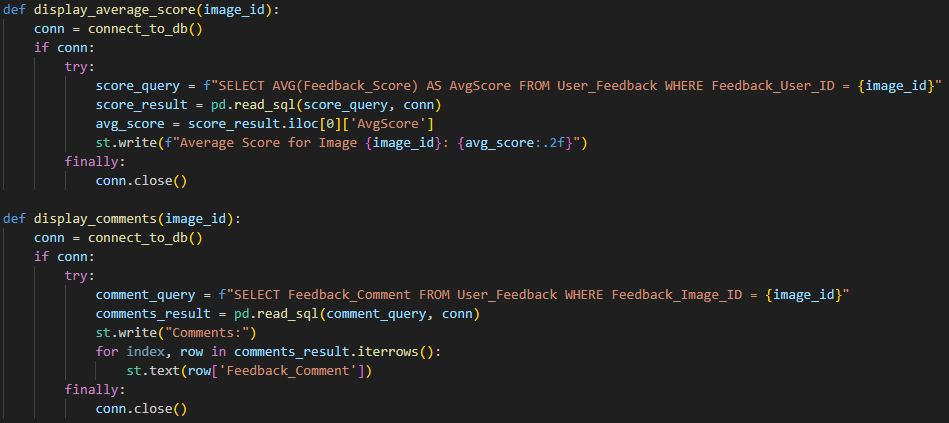


### Web App Script

The web app script which connects to the database, displays the images and querying of the DB using user input, which is also made through python. This connection and query are handled via first establishing a connection with the database using the above script and then querying the database for the image information like Image\_ID, Image\_Path, Prompt\_ID, Prompt\_Description, and Prompt\_Tags. This information is then organized and displayed as a GUI where the user can see the image, and prompt/select an image tag type.



Additionally, the average score and comments for the image is displayed via two helper function queries. This is also a key part of the database’s functionality, as knowing which prompt, tag, or keywords are able to produce the type of image, is not an exact science and therefore having user input not only in creating and uploading the images to the database, but also in rating and commenting on the images help curate which prompts are truly good and which are not.



# App Demonstration



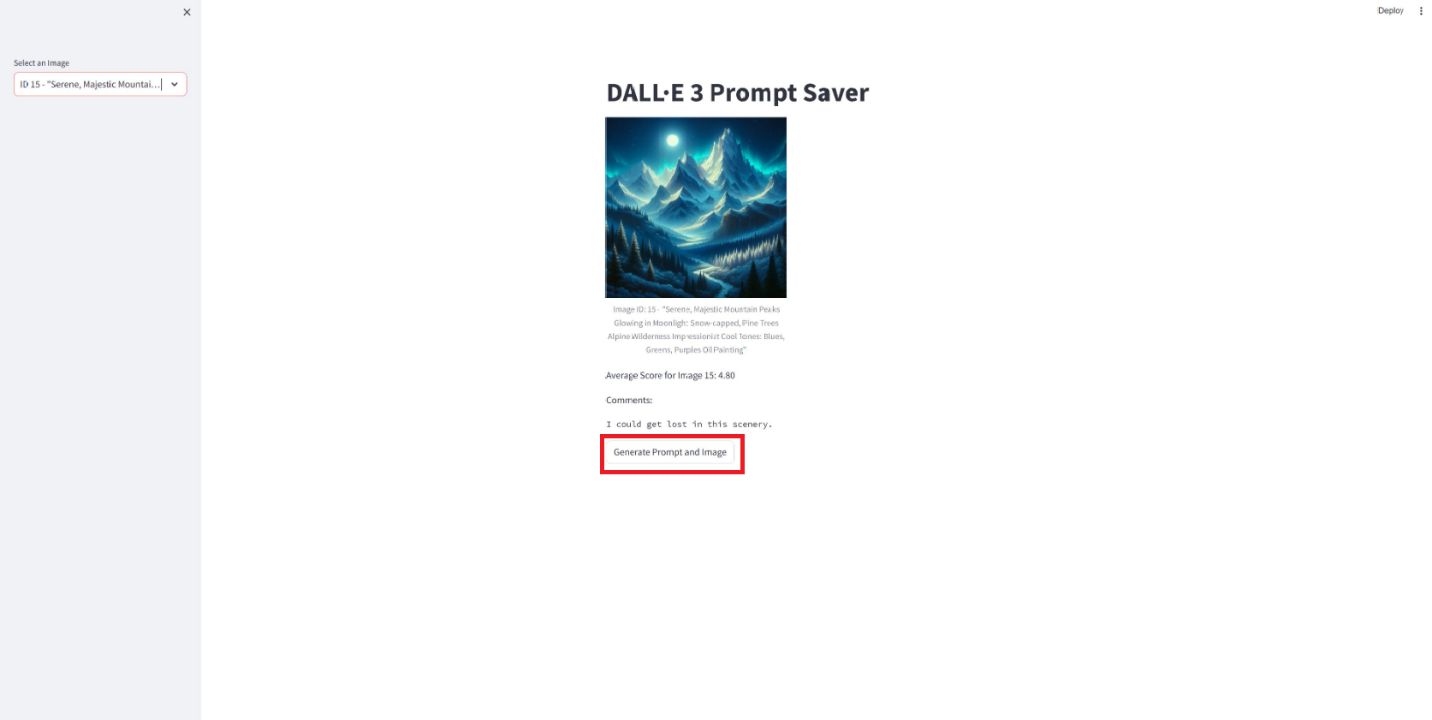
The above image is the basic layout of the webapp and how the stored information in the database will be presented and utilized.

The section that is highlighted in red is where the user will be able to select a specific image based on the prompt and ID of the image.

The image selected is shown in the box highlighted in red above. Under the displayed image in a small light grey font is the image ID and the prompt used to generate the image.

The average score for the selected image can be seen in the section highlighted in red above. This simply displays the previously computed and stored mean value of the rating given by the users.

The comments left by other users can be seen in the section highlighted in red above. Here, there may be multiple comments from multiple users which allow for more information about generating the image and prompting the model, or just general opinions about the image shown.

The last functionality given in the webapp is the generate a similar image using the prompt for the selected image. This will allow users to easily and quickly make new images. A future business may incorporate connections between images created using a prompt to show a parent-child like relationship to show the range of images a prompt style could make.

# Data Question & Answers

1. *Rank the users from highest to lowest Feedback Score*

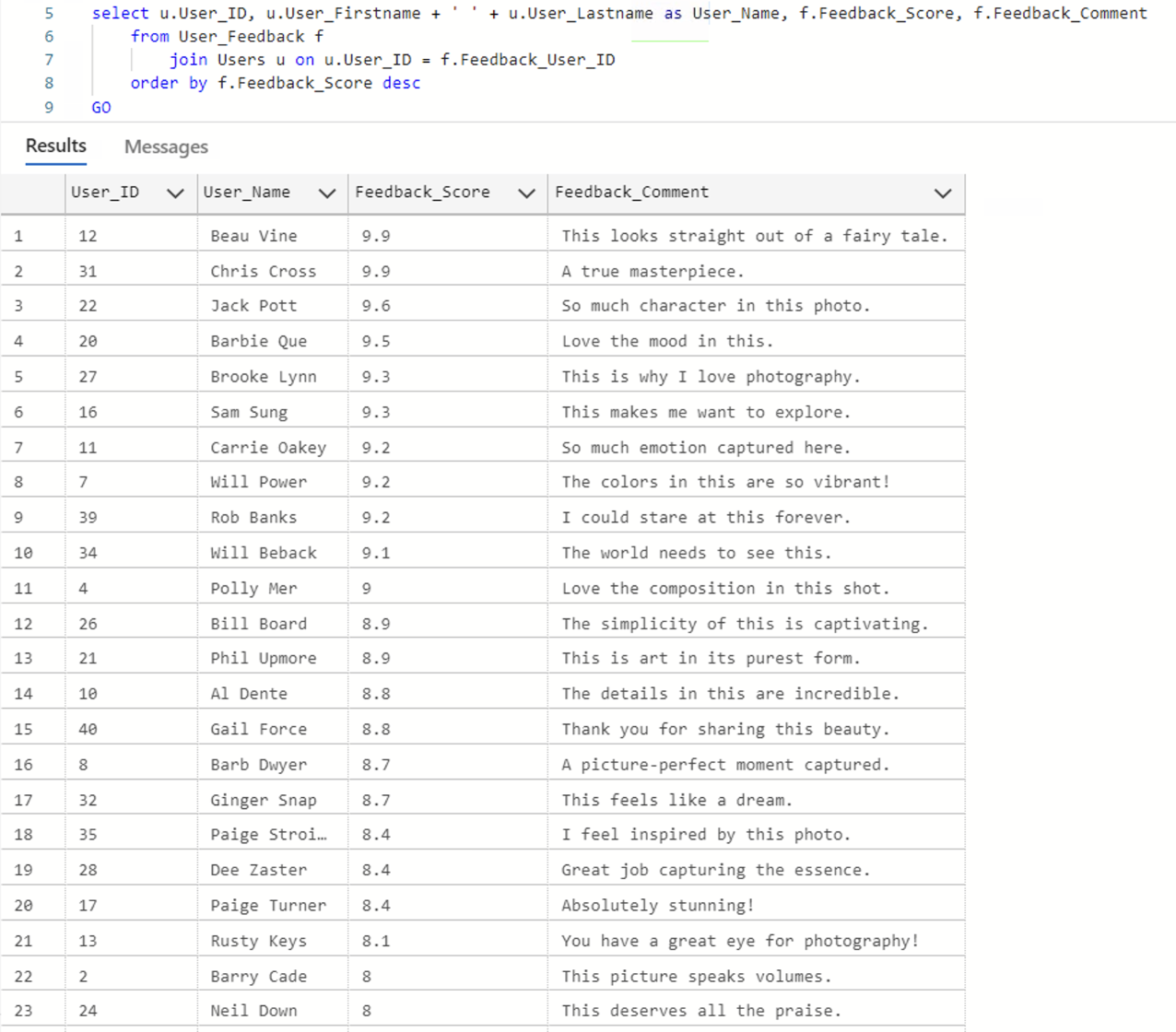
*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, f.Feedback\_Score, f.Feedback\_Comment*

*FROM User\_Feedback f*

*JOIN Users u ON u.User\_ID = f.Feedback\_User\_ID*

*ORDER BY f.Feedback\_Score DESC*

*GO*



1. *Which users had a Feedback Score greater than 8?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, f.Feedback\_Score, f.Feedback\_Comment*

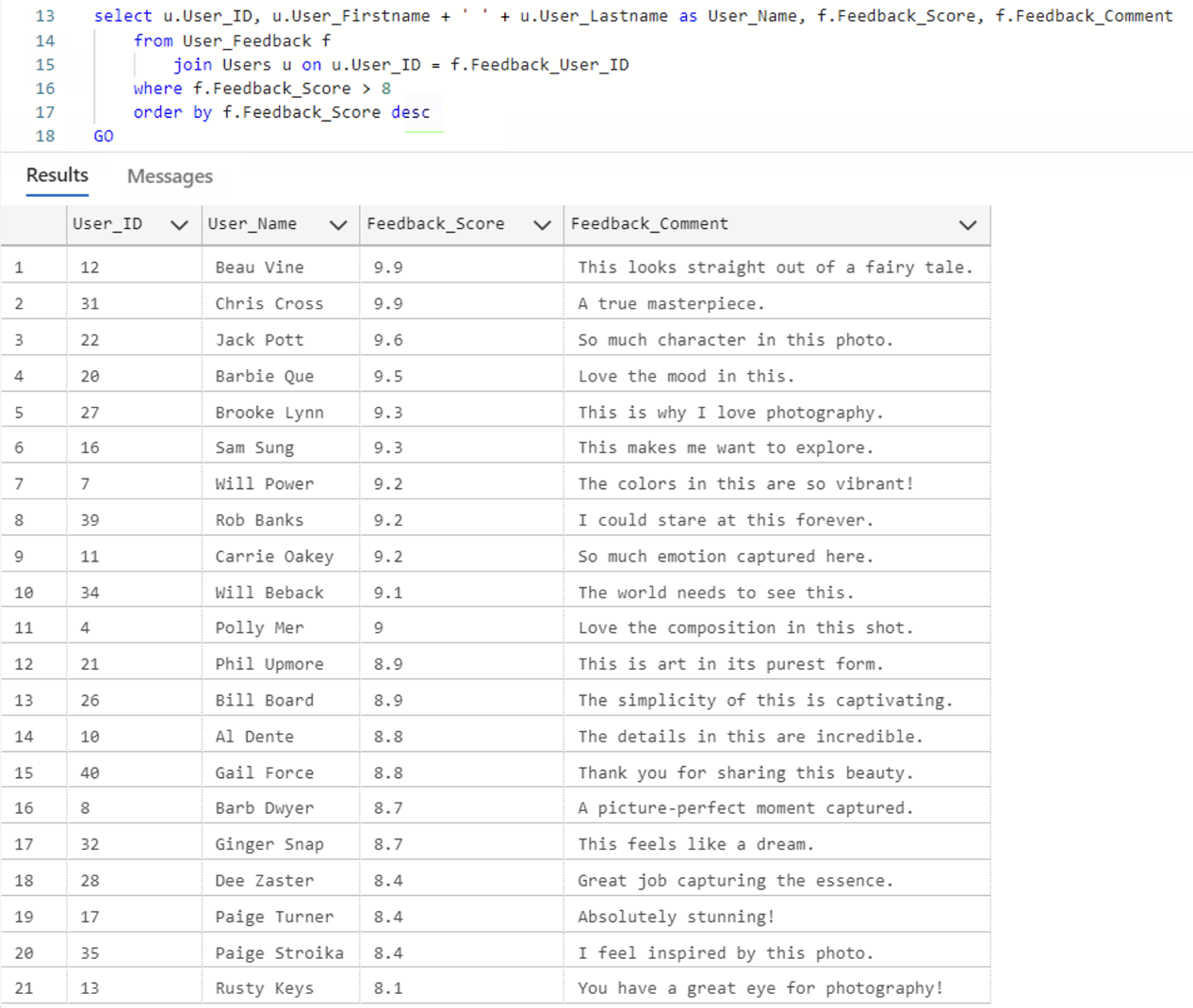
*FROM User\_Feedback f*

*JOIN users u ON u.User\_ID = f.Feedback\_User\_ID*

*WHERE f.Feedback\_Score > 8*

*ORDER BY f.Feedback\_Score DESC*

*GO*



1. *Which users had a high, medium, or low Feedback Score where high ≥ 8, low ≤ 6, and medium is any score between?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, f.Feedback\_Score, f.Feedback\_Comment*

*CASE*

*WHEN f.Feedback\_Score >= 8 THEN 'High Feedback Score'*

*WHEN f.Feedback\_Score <= 6 THEN 'Low Feedback Score'*

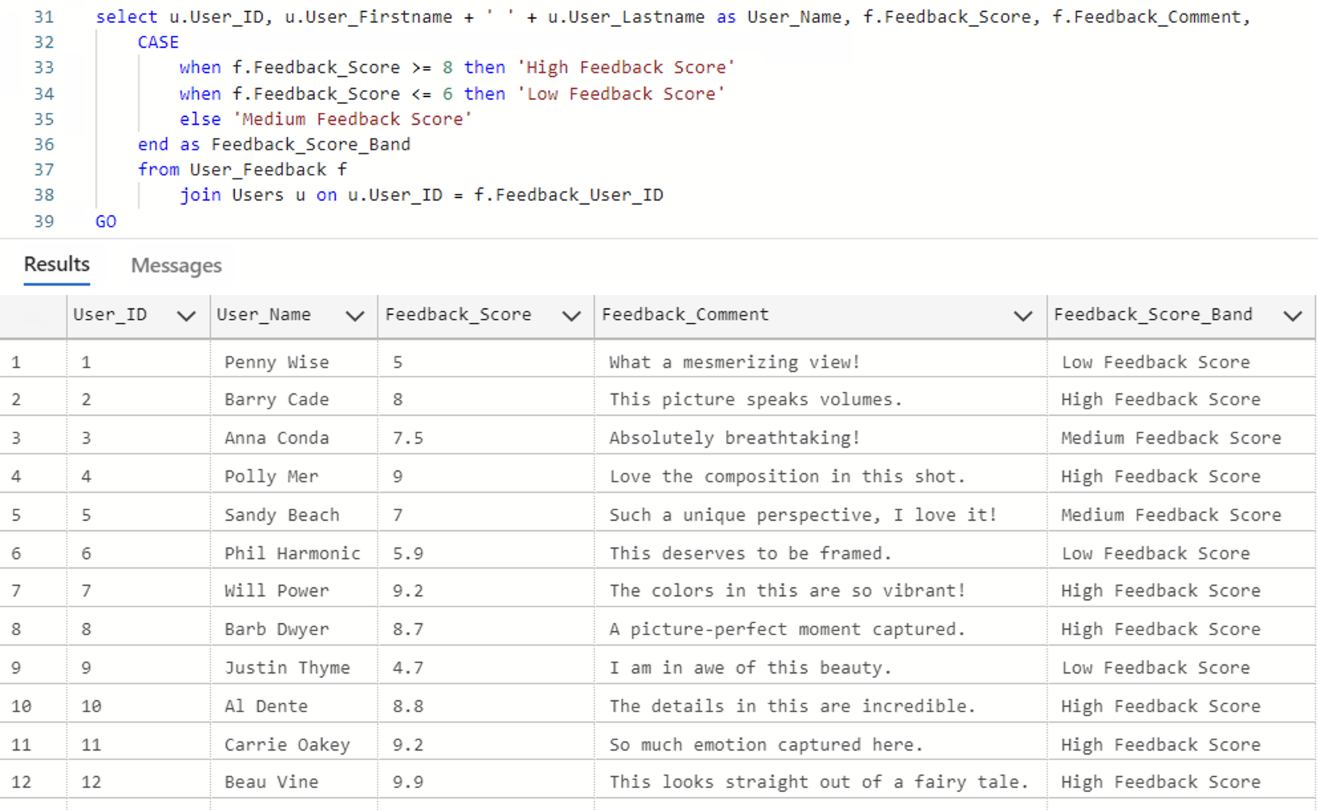
*ELSE ‘Medium Feedback Score'*

*END AS Feedback\_Score\_Band*

*FROM User\_Feedback f*

*JOIN Users u ON u.User\_ID = f.Feedback\_User\_ID*

*GO*



1. *Which users had a Feedback Score less than the Average Feedback Score?*

*SELECT \* FROM(*

*SELECT User\_Firstname + ' ' + User\_Lastname AS User\_Name, Feedback\_Score*

*(SELECT AVG(Feedback\_Score) FROM User\_Feedback) AS Avg\_Feedback\_Score,*

*COUNT(\*) AS count\_rating*

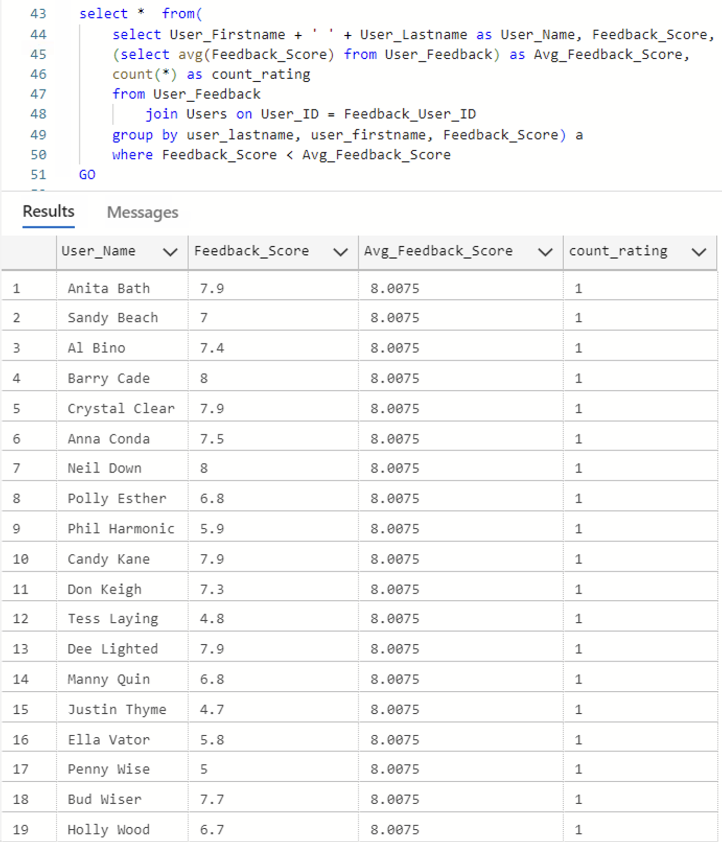
*FROM User\_Feedback*

*JOIN Users ON User\_ID = Feedback\_User\_ID*

*GROUP BY user\_lastname, user\_firstname, Feedback\_Score) a*

*WHERE Feedback\_Score < Avg\_Feedback\_Score*

*GO*



1. *Which users created images within the category of Modern Painting?*

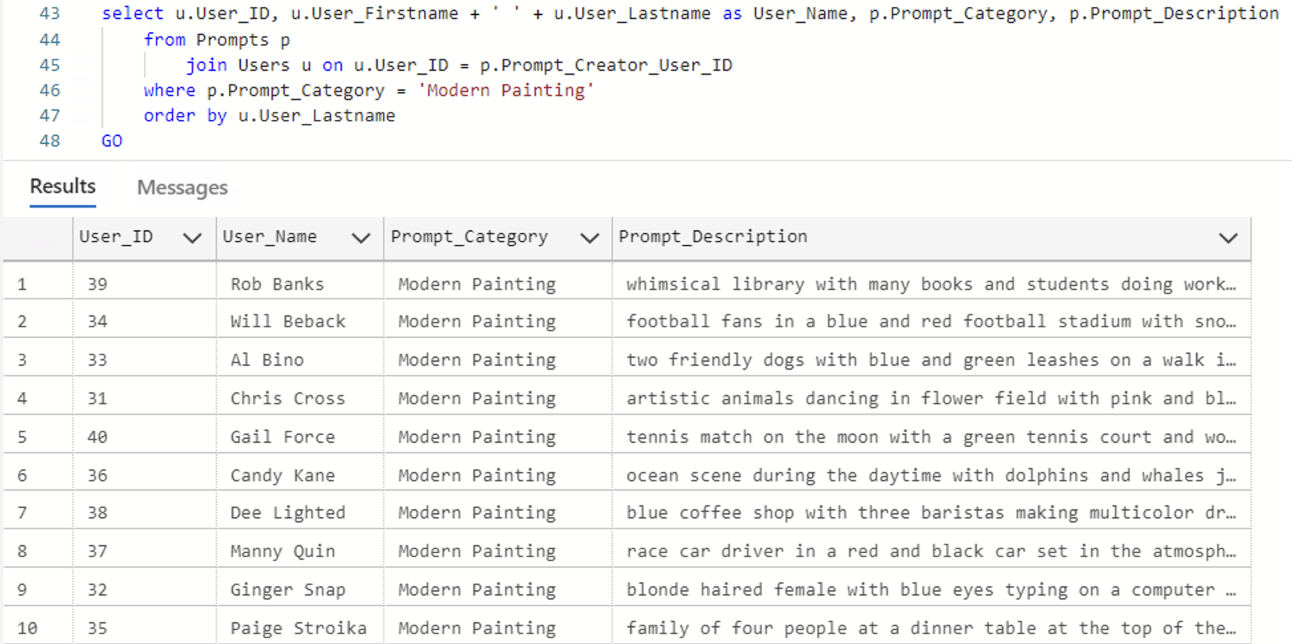
*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, p.Prompt\_Category, p.Prompt\_Description*

*FROM Prompts p*

*JOIN Users u ON u.User\_ID = p.Prompt\_Creator\_User\_ID*

*WHERE p.Prompt\_Category = 'Modern Painting’*

*ORDER BY u.User\_Lastname*

*GO*

1. *Order the users and prompts alphabetically based on the prompt category*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, f.Feedback\_Score, f.Feedback\_Comment, p.Prompt\_Category*

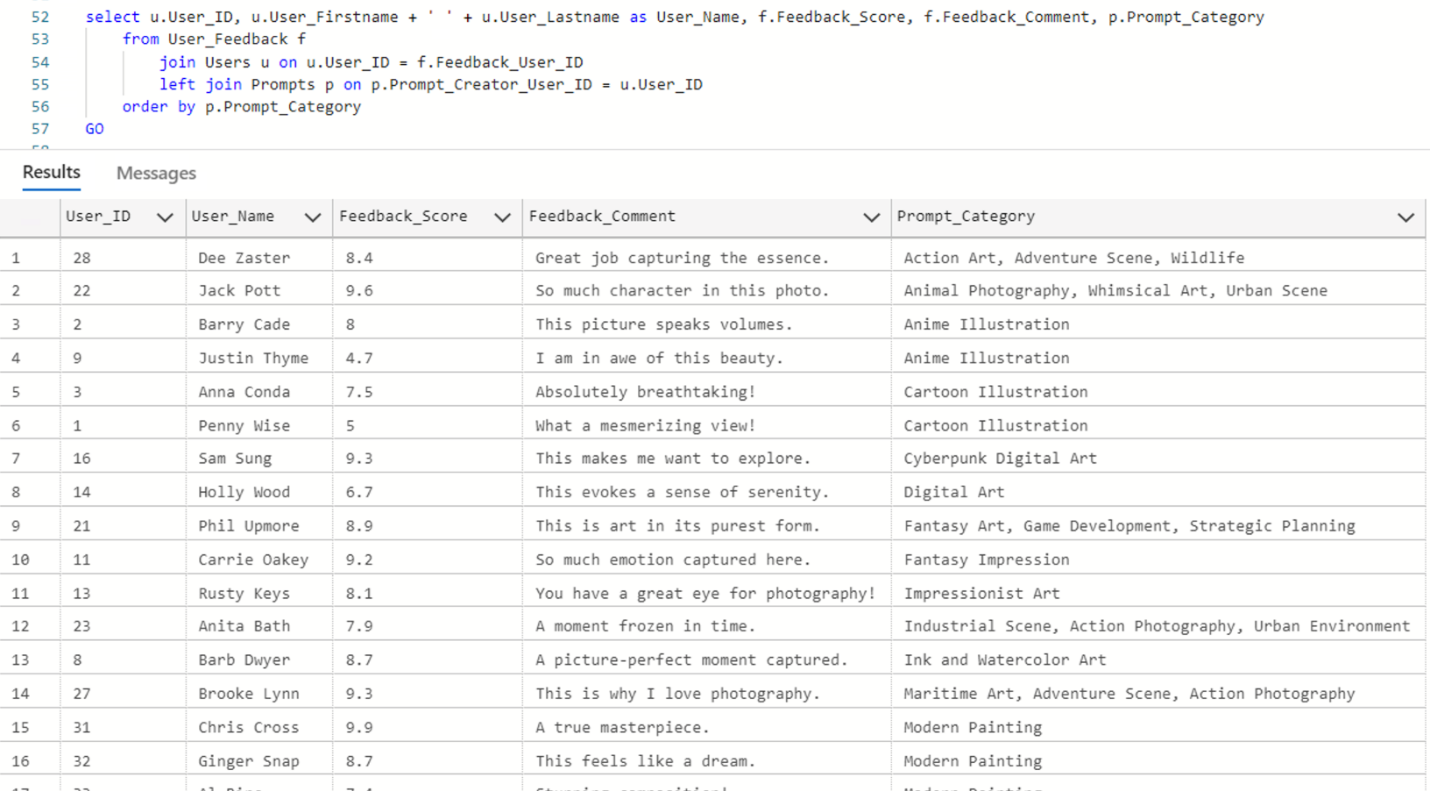
*FROM User\_Feedback f*

*JOIN Users u ON u.User\_ID = f.Feedback\_User\_ID*

*LEFT JOIN Prompts p ON p.Prompt\_Creator\_User\_ID = u.User\_ID*

*ORDER BY p.Prompt\_Category*

*GO*



1. *Which users and their respective prompt used a tag for the color blue?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, p.Prompt\_Tags, p.Prompt\_Description*

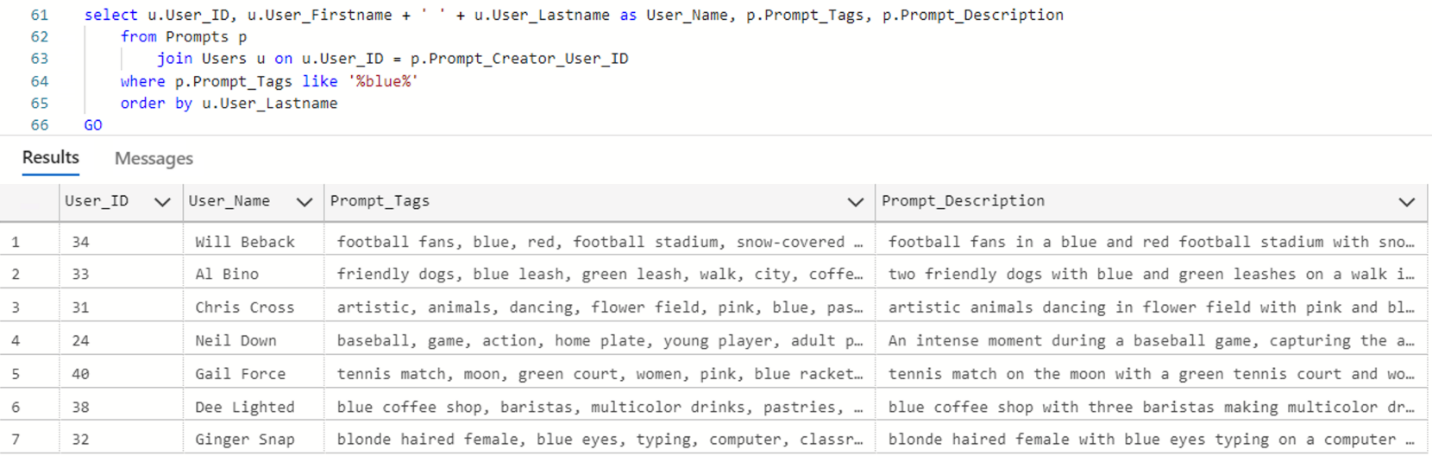
*FROM Prompts p*

*JOIN Users u ON u.User\_ID = p.Prompt\_Creator\_User\_ID*

*WHERE p.Prompt\_Tags LIKE '%blue%'*

*ORDER BY u.User\_Lastname*

*GO*



1. *Which users and their respective prompt used a tag for the ocean?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, p.Prompt\_Tags, p.Prompt\_Description*

*FROM Prompts p*

*JOIN Users u ON u.User\_ID = p.Prompt\_Creator\_User\_ID*

*WHERE p.Prompt\_Tags LIKE '%ocean%'*

*ORDER BY u.User\_Lastname*

*GO*



1. *Which users and their respective prompts used the word game in the description?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, p.Prompt\_Description*

*FROM Prompts p*

*JOIN Users u ON u.User\_ID = p.Prompt\_Creator\_User\_ID*

*WHERE p.Prompt\_Description LIKE '%game%'*

*ORDER BY u.User\_Lastname*

*GO*



1. *Which users and their respective prompts used the word dog in the description?*

*SELECT u.User\_ID, u.User\_Firstname + ' ' + u.User\_Lastname AS User\_Name, p.Prompt\_Description*

*FROM Prompts p*

*JOIN Users u ON u.User\_ID = p.Prompt\_Creator\_User\_ID*

*WHERE p.Prompt\_Description LIKE '%dog%’*

*ORDER BY u.User\_Lastname*

*GO*

# 

# Conclusion & Reflection

The DALL-E 3 Image-Prompt database application shows promise and has proved not only to be an exceptional tool for prompt-to-image database management, but also as a business utility and market piece. With the commercial use and understanding of AI generated text and images seeing a large increase in use, a systemic and use specific approach to managing the vast number of images created while retaining prompt, tag, and genre information, allows for vastly different uses in all sectors from commercial to research from a single database structure.

Throughout the development process, our team collaborated and maintained communication of completed and in-process tasks using Agile methodologies, which included use of a Kan Ban board, weekly meetings, and an efficient centralized communication platform. Many challenges throughout the development process were circumvented with proper communication and careful planning. Many invaluable lessons were learned from the challenges we faced and the steps we took to overcome them, which will become even greater tools and steppingstones for future projects.

Furthermore, with our original idea being completed, implemented and fully developed, we as a group still found places we could reflect upon. Namely, the use of generative AI to create solutions faced by the advent of future technology. There were also additional improvements we wished to see for the project, to introduce further utility into the database, but also to distinguish and differentiate this database from any other. One such idea was to implement a way for users to use a specific prompt in the database to create an image, or a similar image, and have a tree like relationship between the parent prompt-image and the child prompt-image. This will allow for further analysis and understanding of how certain prompts and words reflect and affect the image being created.

# Appendix & Team Meeting Logs

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Length | Members Attendance | Topics Discussed |
| 4/28/24 | 1 hour | Adrian, Jalesa, Samantha, and Shintaro | Discussed ideas for the project, plan of execution, started a Teams account to communicate and share documents, agreed to each have 10 prompts and photos generated from NightCafe (total of 40) for next meeting |
| 5/5/24 | 1 hour | Adrian, Jalesa, Samantha, and Shintaro | Reviewed prompts and photos generated, set up a database in Azure Studio for everyone to access the database, created a KanBan through Teams, split up and assigned tasks to each person for the execution of the project, created skeleton of final paper and presentation |
| 5/14/24 | 1 hour | Adrian, Jalesa, Samantha, and Shintaro | Everyone gained access to the database, reviewed model for the SQL metadata structure, began writing the introductory statements for the paper |
| 5/21/24 | 1 hour | Adrian, Samantha, and Shintaro | Logical and Conceptual models completed and reviewed with the team; Python script finalized to connect database with the app for use |
| 5/30/24 | 0.5 hour | Adrian, Samantha, and Shintaro | Discussed any updates and missing data needed for final input into SQL |
| 6/2/24 | 1 hour | Adrian, Jalesa, Samantha, and Shintaro | SQL script finalized for the metadata and data and reviewed with the team; app tested |
| 6/8/24 | 0.5 hour | Adrian, Jalesa, and Shintaro | Discussed final steps for preparation and review of paper, any missing elements still needed for the paper, SQL, and/or presentation |