Web Development Handbook

Image Generation Site

Introduction:

This handbook covers the design, production, and implementation of an image generation website.

Design & Premise:

The purpose of this website is to receive an image input from a user alongside a prompt, process the image with use of an image-to-image model that applies the prompt as a modifier, and then return the output allowing the user to create and change images as they desire.

The image generator can be used purely recreationally or for modification of images that would potentially require photoshop such as removing a background or particular colour.

The page flow is as follows:

INPUT PAGE 🡪LOADING PAGE 🡪 OUTPUT PAGE 🡪 INPUT PAGE

The layout and styling of the webpages is fairly minimalistic as to convey a more futuristic look especially when paired with a movable 3D background.

A screen shot of a computer screen

Description automatically generated

The background is a 3D set of stars that react to mouse movement to move around the screen.

The idea for the background was to give the user an interactive and visually pleasing interface to view particularly when waiting for the page to generate the image they requested.

A black sky with white dots

Description automatically generated

1 GIF of 3D background on display

Implementation:

The server used is a python-based flask server that sends and receives to the front end on the client side. When the user first connects to the server through the page, a unique session key is created. This session key is used to ensure that images created for that user are only shown to that user.

The server routes different python flask functions based on what the user routes to on their browser. These functions complete their task and return an html template that has variables within that can be altered such as URLs for images.

A black screen with text on it

Description automatically generated

2 A Routing for the initial user connection to show upload page.

The client uploads an image that is validated for correct file types and then the user can enter a prompt that will modify the image (for example changing the colour of the background). The prompt and file upload are also validated checking for input as to not be left blank.

The client is then presented with a loading screen with a bar at the top to represent this.

A screen shot of a black sky

Description automatically generated

3 Loading Screen

During this loading time the client image is received by the server, saved and then the save location is passed into a model that uses stable diffusion to apply the modifications of the prompt.

The model loads its pipeline components and



The client side pings the server-side using JavaScript to fetch to check for completion every few seconds and upon completion navigates to the results page to display the image:

*Page on right with provided image on left, prompt: “make him into a robot”:*

A robot standing on a ledge

Description automatically generatedA statue of a person

Description automatically generated

The user is prompted to generate another image if they wish and can save the image on screen.

The model used can generally produce fairly high-quality images however due to the unpredictable nature of some modern image models and the lower end processing provided these images can sometimes not be high quality enough for use.

The 3D background is achieved via THREE.js by having a Starcreation() class that initiates star objects, gives them a shape and mesh, size, and an X,Y and Z position. These stars are randomly placed within a space of 2000 units squared with the camera placed looking towards them.

There are 45000 stars rendered on the page however due to the small size and detail of the stars they do not present any problems when it comes to performance. Equally having the stars run on the client’s side means that no processing power is taken from the model to do so.

Possible Improvements

There are some improvements that could be made to the site that could increase performance or user experience:

* Showing the user’s inputted image as a preview would help to stop wrong uploads and improve user experience.
* Showing a side-by-side of the user’s image with the generated one could go a long way to show the differences made and highlight the effects of the model.
* A more efficient model could be used possibly with more niche applications.
* A faster computer could be used to process the images, this could be done through a cloud deployment of the application where faster processing power can be purchased.
* The loading page could fetch what the model is currently processing and provide that information to the user.
* Add more image file types.

User Guide

1.Upon opening click “UPLOAD FILE” and choose a file to be uploaded to the site

2. Add a prompt that relates to the image and will change it in some way

3. Press “GENERATE IMAGE”

A screen shot of a computer screen

Description automatically generated

4. Wait for loading screen to complete

A screen shot of a black sky

Description automatically generated

5. Save image and generate another!

A robot standing on a ledge

Description automatically generated