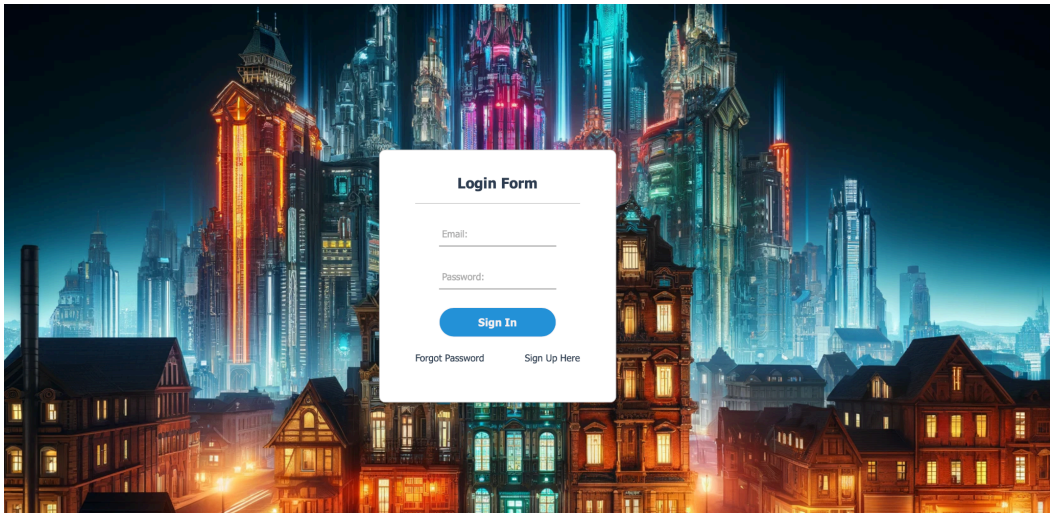
Architexa users manual:

Architexa is a CGAN neural network capable of generating images of buildings based on your prompts.

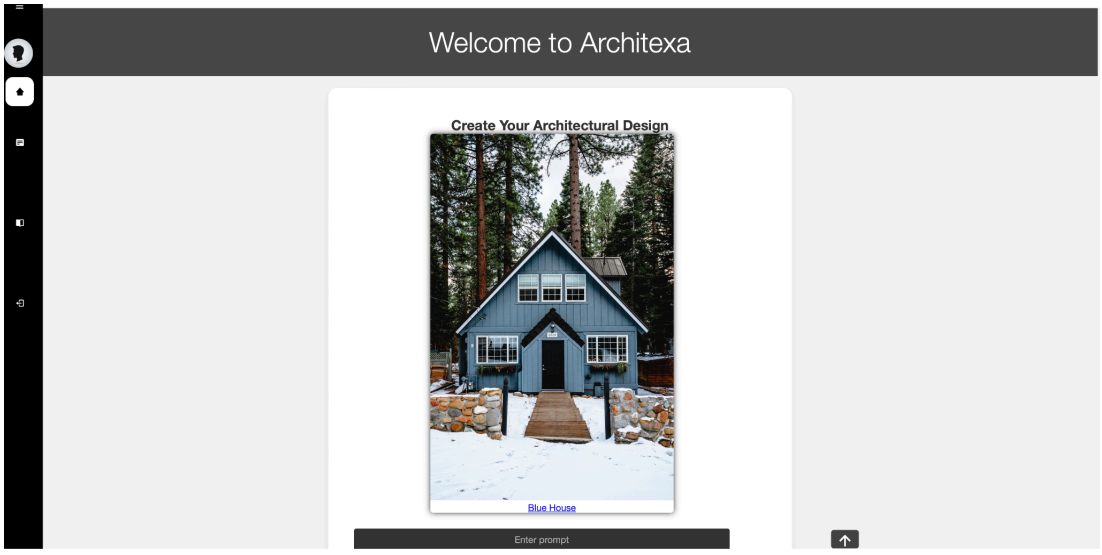
Common users:

To use Architexa navigate to this website: Architexa.io

Where you will see this is the login screen. Please select the Sign up here and create an account. Once you have done this, return back to the main login form and sign in.



Once you have logged in, you will be presented this screen:



(our website also has an about us section if you are interested)

From here you will need to input a short sentence broadly describing the image you want generated and then click submit.

After allowing 75 seconds for generation the UI will then deliver four generated images matching your prompt.

Case tool users:

Our neural network optimisation tool, and our dataset-building tools can be found in the following repository:
Ash237333/208-Project (github.com)

Once there you will need to download either CGAN.py (from the multi-prompt-branch folder), for the optimisation tool

Or the contents of data_preprocessing/generated_images/code for the database building tools

In the network optimiser case you will then need to install python IDLE 3.11, and the tensorflow library for python:

Python Release Python 3.12.2 | Python.org

tensorflow · PyPI

In the database builder case, you will need:

- Google Chrome.
- A discord account.
- A subscription to mid-journey.

Running the network optimiser:

For the network optimisation tool, you will want to attach your CGAN or WCGAN-GP build to the tool by changing the names of files (discriminator and generator) imported in the HyperCGAN script to the names of your networks.

Then feed alltasks.json file some best guess values for hyperparameters before running worker.py (wait about a day then select what you believe to be the best output image from its output) the script will then generate more sets of hyperparameters to test. Then create as many separate instances of the script as your computer can run, and feed each of them one of the

generated test suites, after about a day each of them will terminate with images generated from each test suite value. Select the best images out of all these images and feed that back into the original (or master) script which will then produce new test suites.

After each iteration of this, your neural networks should perform better, and eventually, the hyperparameter values will converge (you will know they are completely optimised when a new test suite consists only of the same values you put into it). However, you may not have time to run to this point, but wait for at least some significant convergence if you want to see a significant improvement in accuracy.

Running the database builder:

Generate Images.py (Setting up the script to automate your own images)
The Generate Images.py script automates the process of generating images. Below is a detailed breakdown of its functionality:
Input Collection

- Album Name: Users are prompted to enter a name for the image album.
- Image Prompt: Users provide a base prompt that defines the theme or concept for the images.
- Number of Images: Users specify how many images they want to generate.

Prompt Construction

- The script constructs a list of unique prompts for each image by appending an index to the base prompt.

Configuration and Output

- Configuration: All user inputs and settings are compiled into a configuration dictionary.
- YAML Configuration: The configuration is saved to a YAML file, allowing for repeated use or modification.
- Bot Compatibility: The script is tailored to interact with the midjourney bot.
- Options:
    - Download: Enables downloading of generated images.
    - Upscale: Allows for the enhancement of image resolution.
    - Variation: Offers variation in image generation.
    - Thumbnail: The option to generate thumbnails is available but disabled by default.
    - Suffix: Custom suffixes can be added to commands as needed.
    - Wait Time: Configurable wait time between issuing prompts to manage API calls or bot interactions.

The YAML configuration file location is set by default to ~INSERT FILE PATH/bulkai.yaml, and users are notified of the file path upon successful creation.

Generating images using BulkAI

After having downloaded the contents of data_preprocessing/generated_images/code open your computers command prompt and run the following:

```
export PATH=$PATH:$(go env GOPATH)/bin
```

Then change the directory for command prompt to the location where the .yaml files from the download are stored (this can just be /downloads)

Then run the following:

```
bulkai create-session
```

This will open a chrome tab at discord, and you will need to login to your discord account (do it via the pop up in that windows or it can cause problems)

Finally declare a command in discord then type and submit the following:

```
bulkai generate --config bulkai.yam
```

(you can stop the generation process by pressing cntrl+c)

For example, it creates this: