

# CP1295 Project

Scott R. Young

July 25, 2023

## Abstract

This project will cover material taught in this course and its prerequisites. Your objective is to make a functional Image Board.

## 1 Operation

Your project will be composed of a “webserver.js” file, as well as any “css”, “js”, and “html” files you deem necessary to complete the objective. Your “webserver.js” file, when run with node.js, should accept GET requests for the files you will use to make the Image Board as well as any API requests that the front-end will use to fulfill its duty. An image board is a way to search and post images and associate them with tags, strings that describe the image, and also allow users to comment under those images. Searching for tags allows the site to present a list of all images matching the tags that were provided.

Your site will have 4 views:

- A homepage with a search bar and search button;
- A view that shows a list of pictures, a search bar with a search button, and a list of tags under the search bar from the tags of the images in the list;
- A view that will present an image, a list of tags associated with that image, and a set of comments associated with that image;
- A page for posting a new image, it will contain a text-input for a URL, a text input for the tags for this image, and a button to send the URL and the list of tags for that image.

## 2 Your Task

Your task will be to:

- Create a web-page that:

- Presents a search bar with a search button
- When the button is pressed it should send a request to the backend with the list of tags from the search bar.
- Create a web-page that:
  - Presents a search bar with a search button
  - When the button is pressed it should send a request to the backend with the list of tags from the search bar.
  - Presents a grid of images that match the provided tags.
  - Presents the list of all tags that match any image in the grid.
- Create a web-page that:
  - Presents a search bar with a search button
  - When the button is pressed it should send a request to the backend with the list of tags from the search bar.
  - Presents an image that match the provided tags.
  - Presents the list of all tags that match the image in on the page.
- Create a web-page that:
  - Presents a post button
  - Presents a text input for an image URL
  - Presents a text input for tags to use for the posted image
  - When the button is pressed it should send a POST request to the backend with the list of tags and URL for this image.
- Implement the following API:
  - POST /search
 

<ul style="list-style-type: none"> <li>* body: tags: [&lt;TAG&gt;]</li> </ul>	<ul style="list-style-type: none"> <li>* return a list of all Image-IDs who's JSON files contain all tags from the body not starting with “-” and contain none of the tags starting with “-”.</li> </ul>
---	--
  - GET /image/<Image-ID>
    - \* return an image's JSON object from its JSON file, using the unique Image-ID to find the image.
  - POST /image/<Image-ID>

- \* body: url: <Image URL>
  - \* Create a new image JSON file for this Image-ID and make the provided URL from the body a field of this object.
- GET /image/<Image-ID>/comment
  - \* Get the list of comments associated with this Image-ID
- POST /image/<Image-ID>/comment
  - \* body: content: <Post-Content>
  - \* Add a new comment to the list of comments for an Image-ID. The contents of the comment will be in the body.

### 3 Limitations

- You are limited to the techniques that have been demonstrated in this course through its assignment, handouts, lecture notes and the course textbook unless otherwise directed by the instructor.