



DOUGLAS COLLEGE

COMMERCE AND BUSINESS ADMINISTRATION

CSIS 3380: ADVANCED WEB PROGRAMMING WITH JAVASCRIPT & AJAX

Assignment 2

Due date: November 23, 2021

(5 % towards your Final Grade)

Total Marks: 100

In the first part of this assignment, you are going to integrate content into web page from an API using Ajax.

Introduction:

Ajax is a technology that allows developers to make asynchronous HTTP requests without the need for a full-page reload. While we are not bound to use jQuery, you may find it easier to use jQuery's shorthand Ajax methods like \$.ajax() or \$.get() to perform XMLHttpRequests.

The jQuery [\\$.ajax\(\)](#) method provides core functionality of Ajax in jQuery. It is used to perform an asynchronous HTTP request. The typical signature of the function is:

```
$.ajax(url,[options]);
```

Using Fetch

The Fetch API provides an interface for fetching resources. The fetch() method takes one mandatory argument, the path to the resource you want to fetch, and returns a promise that resolves to the response to that request (successful or not). Fetch API is built into the window object, and therefore doesn't need to be installed as a dependency or imported in client-side code. A basic fetch request looks like this:

```
fetch(`http://example.com/task.json`)  
  .then(response=>response.json())  
  .then(data => console.log(data))
```

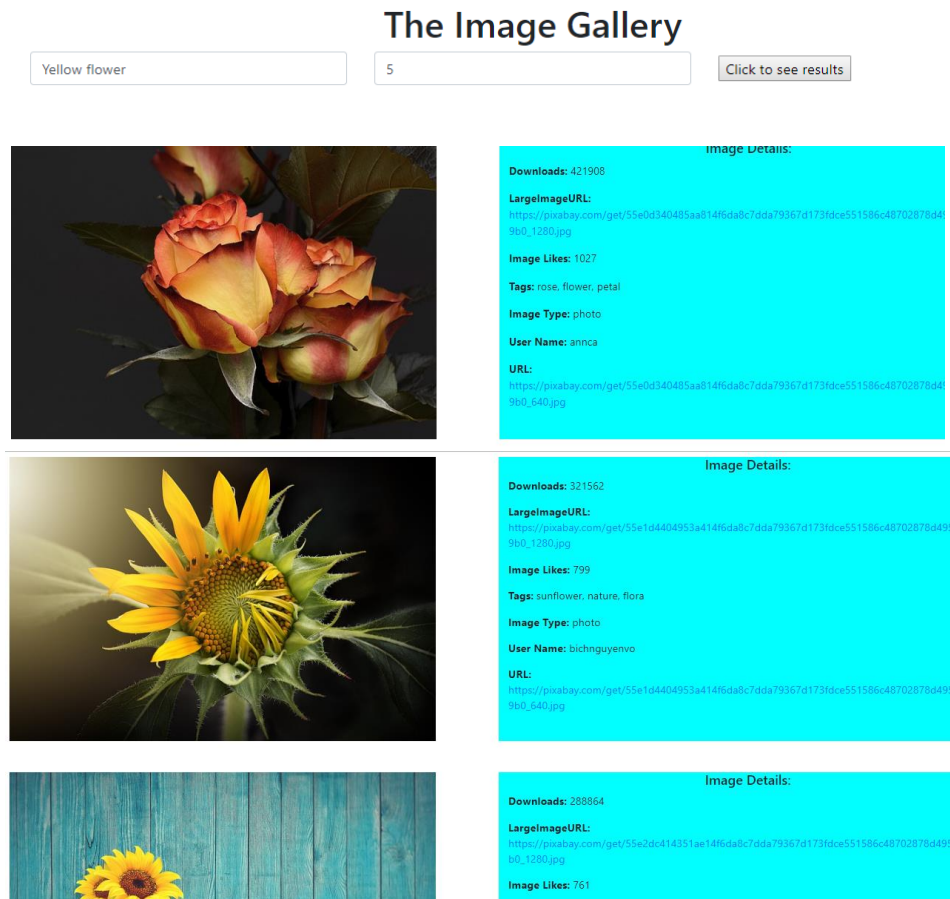
Part 1:

Demonstrate the use of **jQuery ajax and Fetch API** to fetch data from a third-party API and update the part of the HTML page. You are provided with HTML template named "*imageGallery.html*". Use pixabay API <https://pixabay.com/api/docs/> to bring the desired images and image details on your web page (You will have to generate API key to use the API.).

Create two scripts "jqajax.js" (using ajax) and "fetchapi.js" (using fetch API). Both scripts should accomplish the following task:

When the user enters the search keyword for the image and the number of images to be shown, and hits "**Click to see results**" button, the images and corresponding details for each image like **Downloads, LargeImageURL, Image Likes, Tags, Image Type, User Name, URL** are displayed. Make sure the output looks similar to what is shown in Excerpt on the next page. You are free to use your own colors and font styles for displaying the relevant details.

The following excerpt shows a portion of the resulting webpage after the user enters the search keyword “**yellow flower**” and number of images to be 5.

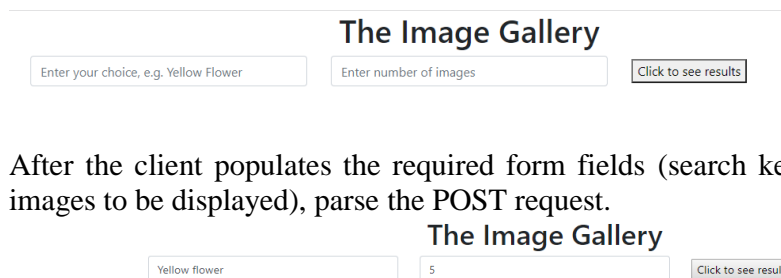


In the second part, build a server using Node/Express and use Express to render a website with the API data.

Part 2:

Complete the following five tasks:

- Handle a GET request from the client at the root route `localhost:3000/`
- Send back the following HTML file to the client as the response:



- After the client populates the required form fields (search keyword and the number of images to be displayed), parse the POST request.

- Use the parsed data to structure the URL to fetch the image data from an external server.
- Next, make a GET request to the external Server (at Pixabay API) with the node HTTPS Module.
- Get the dynamic image data based on what user typed into the input fields in `app.post` and send it back to the client in the same form as shown in previous excerpt.

Submission:

- You are provided with HTML page template that you would use to accomplish the task.
- The assignment is to be completed individually. Any form of cheating or sharing of work may have serious consequences.
- Compress/zip your Assignment_2 folder as LastName_StudentID and upload it to the Blackboard. NO LATE SUBMISSION will be allowed.

Rubrics for grading:

Part 1: 50 Marks jQuery ajax 25 + fetch API 25

Part 2: 50 Marks for five tasks with each task for 10 marks