

San Francisco State University  
CSC 317 – Introduction to Web Development Software  
Assignment Four

## Introduction

For assignment four, you will be adding JavaScript to your registration form and working with a fake API to populate a home page with fake data. Use of bootstrap and jQuery is **NOT ALLOWED**. This assignment will get your feet wet using JavaScript. We will mostly work with the DOM API for this assignment. During the Term Project you will get more experience with the advance features of JavaScript. You must write the additional HTML, CSS, and JavaScript needed for this assignment yourself. **PLEASE NOTE, YOU ARE FORBIDDEN TO USE ANY JAVASCRIPT LIBRARY OR HTML5 Tag Attributes THAT DO INPUT VALIDATION UNLESS STATED OTHERWISE.**

For you HTML forms to know what JavaScript code you will be calling, we need to link our JavaScript files to our HTML files. You may link your JavaScript files to your HTML pages by adding the following to the head tag of each HTML page:

```
<script type="text/javascript" src="path/to/file.js"></script>
```

The **type** attribute will define the MIME type of the file which tells the browser this is a JavaScript file. The **src** attribute specifies the URL of an external script file.

If you need your JavaScript file to load AFTER the HTML is loaded, you may add the **defer** attribute to the HTML tag. This will cause the file to load after the html is parsed

## Setup

Please create a new branch in your repository named “assignment4”. This branch should be created off the master branch. When the assignment is completed you will merge the assignment4 branch back onto master. Please DO NOT delete branches when assignments are completed.

Make sure you are on the master branch of your repository.

```
git checkout master
```

While on the master branch, create a new branch named assignment4

```
git checkout -b assignment4
```

## Requirements

### Form Validation

ALL forms should be in <form> tags. If your form is built using <table> tags this needs to be removed.

**FORM VALIDATION DONE VIA HTML5 IS NOT ALLOWED UNLESS OTHERWISE STATED, LATER WE CAN ADD THEM IN TO SIMPLIFY THE VALIDATION.**

Students are required to add JavaScript to their registration form per the specifications below:

#### ❖ registration.html – for guests to register accounts

- require the user to enter a username that begins with a character ([a-zA-Z]).
- require the user to enter a username that is 3 or more alphanumeric characters.
- require the user to enter a password that is 8 or more characters AND contains at least 1 upper case letter AND 1 number and 1 of the following special characters ( / \* - + ! @ # \$ ^ & \* ).
- require that the password and confirm password inputs are the same.
- require the user to enter an email that is valid.
  - This one CAN BE done with the type attribute set to “email”
- require the user to select that they are 13+ years of age.
  - This one CAN BE done with the HTML attribute require
- require the user to select TOS and Privacy rules.
  - This one CAN BE done with the HTML attribute require

When implementing the above requirements think about what happens when these requirements are not met. Some of these requirements can be verified as the user types and some can be verified when the user clicks the submit button. These design choices I leave up to you. If the data is invalid the form SHOULD NOT BE submitted. If the data is valid, simply let the page refresh or show a message saying the form was submitted.

## Fetching Posts from an API route

For this portion of the assignment you will need to create and style a new HTML page named `home.html`. This page will be used to show results from an AJAX call made to a fake API. We will use the following API route for this assignment:

<https://jsonplaceholder.typicode.com/albums/2/photos>

This API route will return an array of JavaScript objects that represent photos. Here is an example of such an object:

```
{
  "albumId": 2,
  "id": 51,
  "title": "non sunt voluptatem placeat consequuntur rem incidunt",
  "url": "https://via.placeholder.com/300/8e973b",
  "thumbnailUrl": "https://via.placeholder.com/150/8e973b"
}
```

With this object we are going to use the `title` and `url` to show a series of photos on our newly created HTML page. These photos will need to be organized neatly onto the page. Please make sure to use the proper CSS and HTML to correctly display the photos. **DO NOT STORE THESE PHOTOS IN A HTML TABLE. CANNOT USE THE `<table>` tag.** Please show all photos on your newly created `home.html` page with the following specifications:

- No more than 5 photos per row but must have at least 3.
- All photos shown should be the same size.
- Each individual object should show the photo and title. How these are organized is up to you.
- Photos can either be fetched with a button press or when the html page is loaded.
- You may use either the [XMLHttpRequest object](#), [fetch API](#) or [axios](#) to retrieve the photos.
- Page should be able to handle any number of posts.
- At the bottom or top of the page (just after the photos) display the number of photos being displayed
- Add an onclick event to each photo container that when clicked, the photo fades out over time. You may choose the speed of the fade out. Once the fadeout is complete, remove the element from the DOM.
- After removing the faded out div from the DOM, decrement the photo count by 1 to show the correct current number of photos being shown.

## Submission

When you are done with the assignment please do the following:

- ❖ Save all work done.
- ❖ Commit all HTML, CSS and JavaScript files to the assignment4 branch.
- ❖ Push new commits to GitHub (these should be on the assignment4 branch NOT master, assignment2, or assignment3 branch).
- ❖ Verify the commits are pushed to GitHub by browsing your repo in your favorite browser.
- ❖ Once verified, merge all the work done from your assignment4 branch to master.

## Grading

Your assignment will be graded on the following criteria's:

- ❖ Correctness of required form validation.
- ❖ Correctness of fetching and displaying results from fake API.
- ❖ Amount of duplicated code. Some form validation can be used for multiple forms.

## GitHub Submission

When you have completed the assignment please commit all code to the assignment4 branch and then merge the changes back into the master branch. You may use the following commands.

### Committing Changes

```
git add .
```

```
git commit -m "some message"
```

```
git push origin master
```

### Merge Changes into Master

While on the assignment4 branch, please verify all your changes are on GitHub. You can do this by going to your repository on GitHub and browsing your repository looking for the files.

Once you have verified your changes are present on GitHub, switch to the master branch on your local repository (this is on your computer). Once switched execute the following command:

```
git merge assignment4
```

This will most likely produce a fast-forward, meaning no new commits will be made. Once the merge is completed, push the updated branch to GitHub with the following commands:

```
git push origin master
```