Assignment 1

In this assignment, you will create a web server with endpoints that execute server side logic.

Coding Guidelines

- When creating your solution, you must use the coding practices and conventions demonstrated in class. A solution that does not reflect what was taught in class will not be accepted (0 grade) and/or be subject to an academic integrity review.

- Javascript syntax rules:
  - Variables must be declared using let/const, not var
  - Functions must be declared using arrow function syntax, not function() syntax
  - When checking equality, use strict equality (triple equals ===), **not** double equals (==)
  - Do **NOT** use higher order array functions: forEach, map, reduce, filter, closest, etc.

Submission Requirements:

- ❏ Create a zip file containing your entire project folder. Name your name FIRSTNAME_MUSIC.zip, where you replace FIRSTNAME with your first name, example: CARLOS_MUSIC.zip.

- ❏ A screen recording as described in the instructions. If your screen recording is too large for the submission dropbox, then you should upload the screen recording to Microsoft OneDrive and share the link in the submission comments. Ensure you update the link sharing settings so that it is accessible by anyone in the college with the link.

How to Create your Project:

1. Create NodeJS project and setup an Express server.

- ❏ Create a folder called FIRSTNAME_MUSIC. Replace **FIRSTNAME** with your name, example: CARLOS_MUSIC
- ❏ Inside the folder, create a new NodeJS project. (`npm init -y`)
- ❏ Install Express, Handlebars and any other required dependencies (`npm install express, npm install express-handlebars`)
- ❏ After installing Express, create a Javascript file called `server.js`. Add the template code for creating an Express web server in this file.
- ❏ After setting up the Express server code, then write your solution code.

2. When you are ready to submit:

- ❏ Create a zip file containing your entire project folder.
- ❏ Rename your zip file FIRSTNAME_MUSIC.zip. Replace **FIRSTNAME** with your name, example: CARLOS_MUSIC.zip.
  - ❏ Ensure you use a zip file. Rar and 7zip files are **not** accepted.
- ❏ Create a screen recording showing the app execution with all the functionalities. Rename the file to Firstname_Music.mov/mp4. Upload the video recording or host the video on Google Drive and share the link with the submission.

Academic Integrity

- You are responsible for familiarizing yourself with the college's Academic Integrity Policy.
- This is an individual assessment
- Situations which often cause academic integrity issues:
    - Reposting any part of the assessment to online forums or homework help websites
    - Contract plagiarism: Purchasing a solution, or completing a solution for financial compensation
    - Sharing or receiving source code, references, or assistance from others

Problem Description:

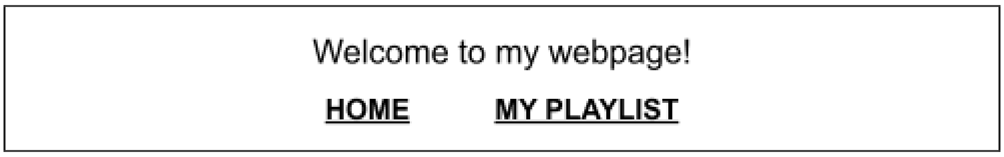
Using NodeJS, Javascript, Express, and Handlebars develop a server side web application about songs.

The server has the following endpoints:

- / → displays a HBS page that shows a list of songs
- /playlist → displays a HBS page that shows the songs on a user's playlist

Both pages must have a common <header> with two menu links:

- Home → clicking on this link navigates to the / endpoint
- My Playlist → clicking on this link navigates to the /playlist endpoint



Create an array of songs called **music**

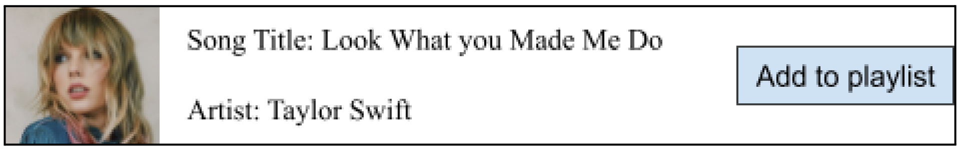
- Each song must be represented as an object literal
- Each song must have a title, artist, artist image, id
- Initialize your array with 4 songs of your choice.

Create an array called **playlist**

- Initialize this array as an empty array.

When the user visits the / endpoint in the browser:

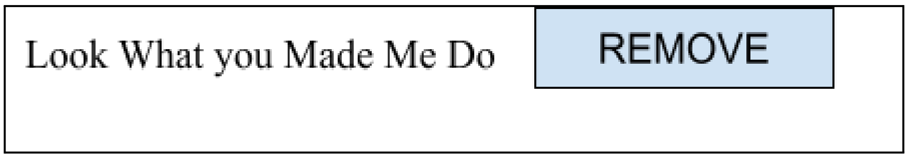
- Respond to the browser with a handlebars template.
- The handlebars template should show all the items in the music array



- You must use CSS Flexbox properties to create the layout for each song
- Clicking on the ADD TO PLAYLIST should add the song to the **playlist** array
  - A song *can* be added more than once to the playlist
  - The POST endpoint should accept the id of the song that should be added as a url parameter (example: /add-song/3)
- After a song is added, use res.render() to show the page updated with the list of songs.

When the user visits the /playlist endpoint:

- Respond to the browser with a handlebars template
- The handlebars template should show all items in the **playlist** array. You only need to show the song title.
- Each song should be displayed as follows:



- Use CSS flexbox to create
- Clicking on the REMOVE button removes the selected song from the playlist.
  - The POST endpoint should receive the id of the song as a url parameter (example: /delete-song/3) that should be deleted from playlist.
- After a song is deleted, use res.render() to show the page updated with the list of songs.

END OF ASSESSMENT