index.html

```
div#app > div.product-display > div.product-container > div.product-info > p
```

```
15      <div class="product-display">
16        <div class="product-container">
17          <div class="product-image">
18            <img v-bind:src="image">
19          </div>
20          <div class="product-info">
21            <h1>{{ product }}</h1>
22            <p>In Stock</p>
23            <p>Out of Stock</p>
24          </div>
25        </div>
26      </div>
27    </div>
28
29    <!-- Import App -->
30    <script src="./main.js"></script>
31
32    <!-- Mount App -->
33    <script>
34      const mountedApp = app.mount('#app')
35    </script>
36  </body>
```

main.js

```
main.js > app > data > inStock
1  const app = Vue.createApp({
2      data() {
3          return {
4              product: 'Socks',
5              image: './assets/images/socks_blue.jp
6              inStock: true
7          }
8      }
9  })
```

MMED-5006

MULTIMEDIA AUTHORING 4

3,936 contributions in 2022 — Contribution settings

TABLE OF CONTENTS

PLEASE NOTE:

Use of **.getElementById()** and **DOMContentLoaded** will result in a 10% penalty for each instance found in any assignment.




MMED-5006

COURSE DESCRIPTION

This course will reinforce and expand on the range of tools and applications covered in MMED-3039, as well as continued techniques used for the development of web based interactive applications. Additional API's and Libraries will be introduced in this term.

COURSE LEARNING OUTCOMES

Upon successful completion of this course, you will be able to reliably demonstrate the following Course Learning Outcomes which will be taught and evaluated:

1.) Understand the purpose and benefits of using development tools like MAMP/WAMP for local server environments and Node.js/NPM for JavaScript development.
2.) Learn how to set up and configure MAMP/WAMP for local web development and Node.js/NPM for JavaScript package management.
3.) Understand the role and benefits of JavaScript build tools in modern web development.
4.) Practice using a bundler to set up a project, bundle JavaScript modules, and handle assets.
5.) Implement AJAX form submissions to send and process form data asynchronously.
6.) Understand different JavaScript paradigms, including Object-Oriented Programming (OOP) and Functional Programming (FP).
7.) Understand the concept of reactive programming and its application in JavaScript.
8.) Learn about reactive libraries such as Vue.js,, React, and Angular and their features.
9.) Understand the fundamentals of Vue.js, a popular JavaScript framework for building user interfaces.
10.) Learn about Vue's core concepts, including data binding, directives, components, and the Vue instance.
11.) Practice retrieving data from a PHP/MySQL/JSON API and displaying it in a web application
12.) Understand the importance of code reviews in maintaining code quality, improving collaboration, and ensuring best practices.
13.) Understand the concept and benefits of Test-Driven Development (TDD) in software development.
14.) Present proposals and products to clients;

GRADE BREAKDOWN

Assignments/Projects - 70%
Final Integrated Project(FIP) - 30%

Prepared by Marco De Luca- m_deluca3@fanshaweonline.ca
All "*" represents a week with a project assignment or submission.

COURSE BREAKDOWN

MODULE 1 - CLIENT PROJECT LAUNCH/PORTFOLIO REVIEW

WEEK 1 *
- Teams Announced
- Client Project Kickoff Presentation
- FIP - Assigned (Due week 13), Value 30%

WEEK 2 *
- Review and correction/builds of common portfolio issues
- Portfolio - Assigned (Due week 7, Value 25%

MODULE 2 - AJAX REVIEW/RESTFUL APIS

WEEK 3 *
- AJAX/Fetch Review
- Homework #1 - Assigned (Due week 5), Value 15%

MODULE 3 - FORMS, NODE, JS BUILD TOOLS, JS PARADIGMS

WEEK 4 *
- AJAX Form Build
- Intro to Node (Basics)
- FIP Checkpoint (25% of FIP)

WEEK 5 *
- NPM (Node Package Manager), package.json
- CORS Review/JS Modules
- Homework #1 - Submission

WEEK 6 *
- JS build tools - Parcel
- Intro to JS paradigms (OOP & Functional programming)
- Homework #2 - Assigned (Due week 10), Value 15%

MODULE 4 - REACTIVE JS, VUE.JS

WEEK 7
- Intro to Reactive Javascript/Data Reactivity with Vanilla JS
- Discuss and review popular reactive libraries (Vue, Angular, React)

WEEK 8 * * *
- Intro to Vue,
- Text interpolation
- Directives
- Attribute bindings
- Homework #3 - Assigned (Due week 12), Value 15%
- FIP Checkpoint (35% of FIP)
- Portfolio Due

WEEK 9
- Vue directives continued
- List rendering
- Filters

WEEK 10 *
- Vue event handlers
- Methods
- Computed properties
- Homework #2 Due

MODULE 5 - LUMEN API, TDD, CODE REVIEWS

WEEK 11
- Work with the Lumen API built in web class

WEEK 12 *
- TDD - Test Driven Development (Mocha and Chai)
- Code Reviews
- Homework #3 Due

WEEK 13 *
- Open Lab
- FIP submission (40% of FIP)

WEEK 14
- No classes (Time to polish work for industry night)

WEEK 15
- Industry Night

ADDITIONAL LEARNING MATERIALS

Learning Vue.js

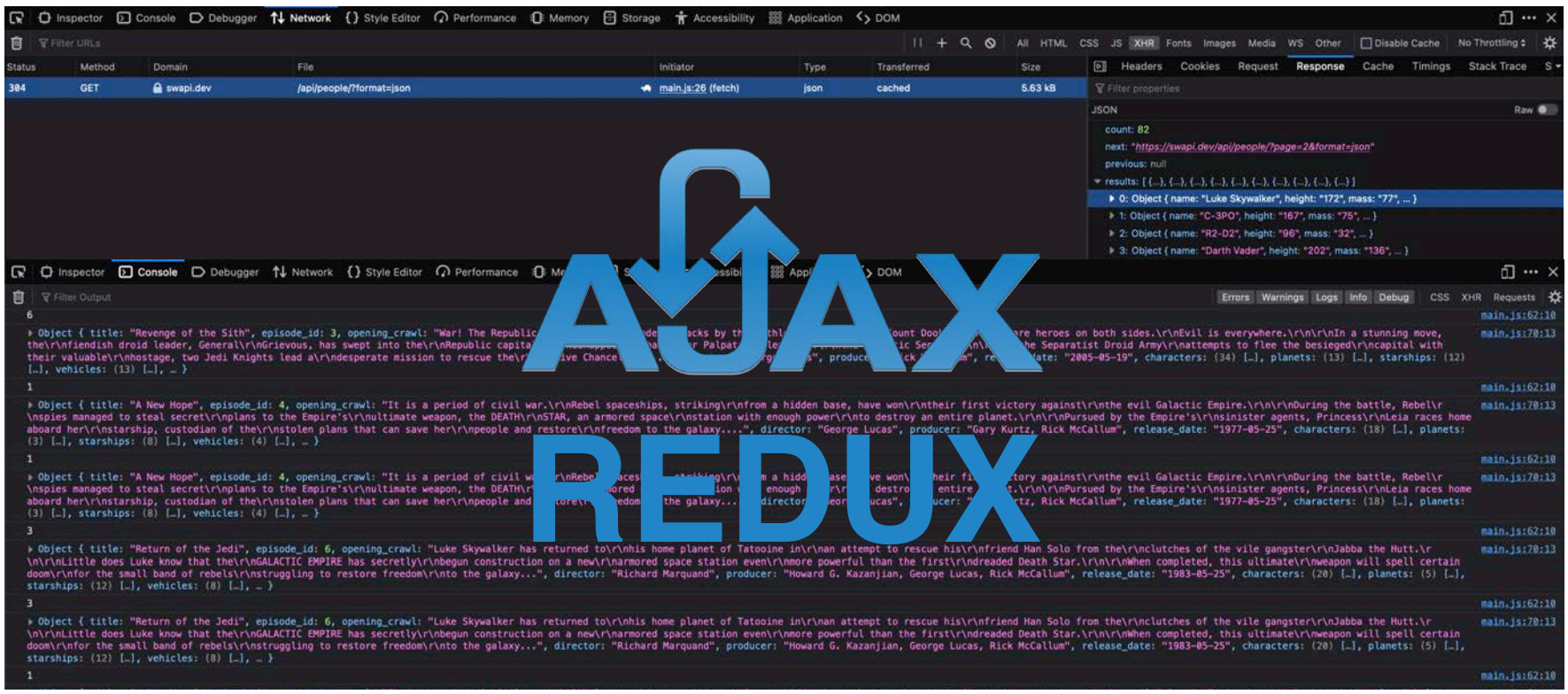
Learning npm: A Package Manager

Additional Reference

JavaScript Essential Training

LinkedIn Learning is available for free through https://fcsw2.fanshawec.ca/portal/index.htm. Look for LinkedIn Learning under My Bookmarks and Campus Links on the right side of the page. You will be prompted to login, using your FOL login credentials to access LinkedIn Learning. You will need to login first before you can click or copy and paste the above links.

HOMEWORK BRIEFS



HW #1 – AJAX REDUX

Assignment Description

Working in pairs, you will research the Star Wars API (https://swapi.dev) and create a character/movie guide. Look through the documentation to learn how to access the information as JSON. You must populate an unordered list with 10 or more characters.

Each list Item should contain the name of a character that acts as a link to one of the movies the character was in. When the character link is clicked a separate AJAX call/function will be called and will return a movie title, the opening crawl and movie poster to a separate section on the page (you will have to source the poster images and store them in your images folder).

The guide must be responsive from mobile to desktop. Additional GreenSock enhancements and features are strongly encouraged. You must also also make use of the HTML Template element or template literals as covered in class. Your Fetch functions should handle the stages of an AJAX request gracefully - what happens if the request fails (use separate functions / handlers for each stage of an AJAX request). You should also implement a loading icon to indicate the content is loading.

Submission Requirements

Create a project repo and a Readme.md document for the repo with detailed information about the project.

Use branches as appropriate with your project. NEVER work directly on the master branch.

Merge everything to the master branch and submit a link to the dropbox on FOL.

DUE DATE & RUBRIC

Homework must be submitted by Week 5 (Check dropbox date and time): ONLY THE MAIN BRANCH WILL BE GRADED. NO ZIPPED FILES!
- Github repo named correctly (lastName_firstInitial_HW1): readme.md file, main branch, development branch(es)
- Name the branches appropriately per feature IE des.yourinitials.artwork, dev.initials.script

GRADING RUBRIC /15
HTML/CSS /2 Design /2 GitHub Best Practices /1 First Data Call(Grab list of characters) /2 Populate List /3
Second Data Call (Grab Movie) /2 Populate Second Box (Movie Info, Opening Crawl, Poster Image) /3

HW #2: OOP JS, MODULES AND BUNDLING

Assignment Description

Using the principals of OOP covered in class you will work in pairs to create a custom class and then instantiate objects using that class and the new operator. The object that is created by the constructor function must contain at least 2 properties and 2 methods and be able to take in parameters/arguments.

The object(s) must be used on a web page in a practical manner. Examples include a basic RPG game or photo gallery.

The page must be responsive from mobile to desktop. Additional GreenSock enhancements and features are strongly encouraged.

You must make use of modules and Parcel.js to bundle the modules.

Please treat each assignment as a potential portfolio piece.

Submission Requirements

Create a project repo and a Readme.md document for the repo with detailed information about the project.

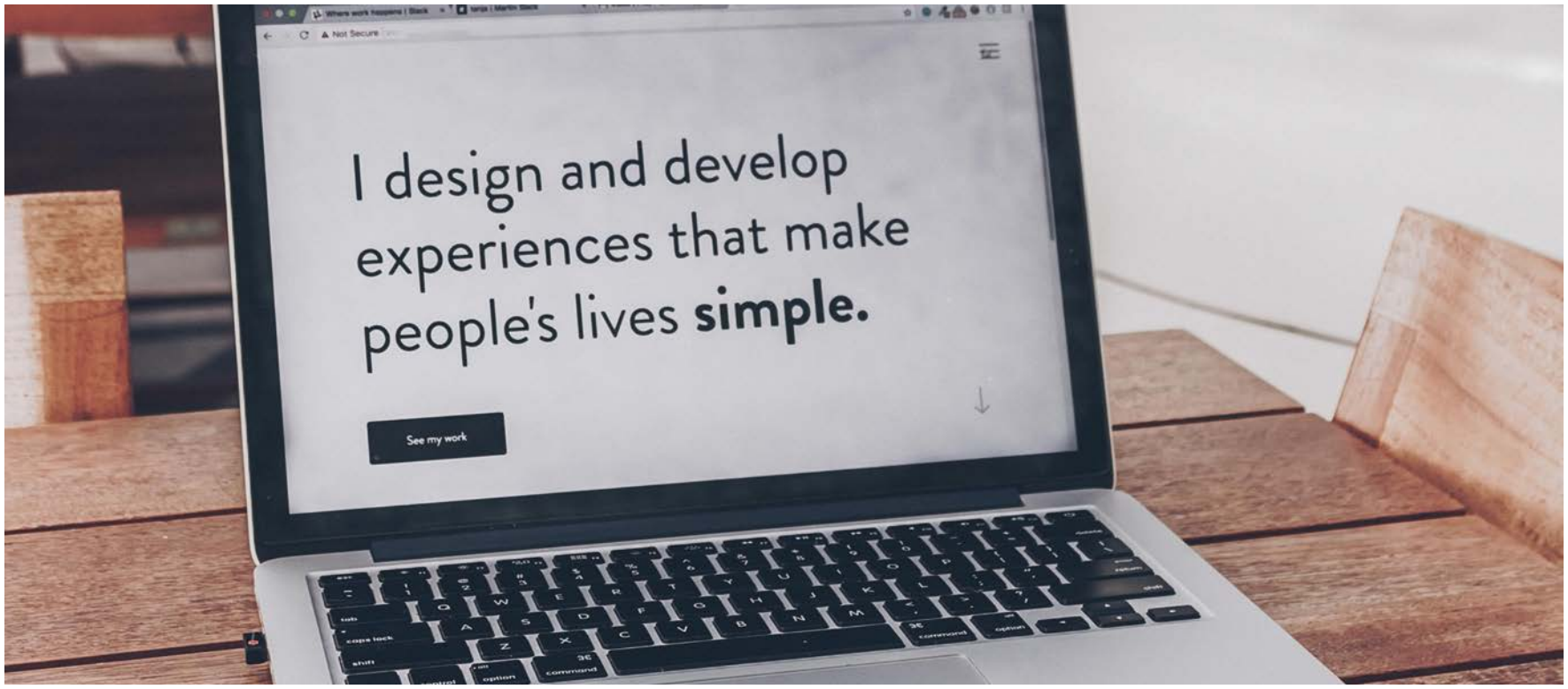
Use branches as appropriate with your project. NEVER work directly on the master branch.

Merge everything to the master branch and submit a link to the dropbox on FOL.

DUE DATE & RUBRIC

Homework must be submitted by Week 10 (Check dropbox date and time): ONLY MAIN BRANCH WILL BE GRADED. NO ZIPPED FILES!
- Github repo named correctly (lastName_firstInitial_HW2): readme.md file, main branch, development branch(es)
- Name the branches appropriately per feature IE des.yourinitials.artwork, dev.initials.script

GRADING RUBRIC /15
Valid HTML/CSS /2 Design /2 GitHub Best Practices /1 OOP implementation /5 Modules /3 Bundling /2



PORTFOLIO – COMPLETION

Assignment Description

The portfolio site you finalize this term is ultimately what you will use to present your work and your skill set (both technologically and graphically) to potential clients and employers. As such, the design and presentation of that design is up to you entirely, within the technical parameters listed in each class.

The site must have pages for all main sections (no empty links in the global navigation, no Lorem Ipsum text).

The site must render appropriately in Firefox (Windows and Macintosh), Safari, Chrome (Windows and Macintosh) and Edge for Windows along with the major mobile platforms (Android and IOS).

Assignment Requirements

Your portfolio site must include an HTML5 mobile-first responsive layout, with CSS3 and GreenSock elements (animation, scroll effects, transitions etc) used to whatever extent they can be.

Use Javascript to create some additional features on your site. For example, You can build some type of gallery (lightbox, slider, popover, video player, etc.) for your portfolio work that uses your own custom functionality.

ALL of the JavaScript code MUST be your own. Libraries (other than what is covered in class) are not allowed and will not be considered when graded.

Your JavaScript code should follow best practices as outlined in class (no inline JavaScript in the HTML, no jQuery, use IIFEs as required). Use of modules and bundling is also required.

Use a CSS preprocessor (SASS) for your CSS files. Use SCSS modules with files relevant to sections of your website (ie _nav.scss, _portfolio.scss) and compile them to a minified production file.

You'll need to include a contact form that uses a DB to store user information in an appropriate table via an AJAX call. Be sure to handle any errors and success states returned from your PHP file and update the view as appropriate using JS. For example, consider how to handle invalid emails, missing fields, etc.

The Readme is a place to show some of your personality - take advantage of this to set yourself apart from your peers. Prospective employers will look at your repos to evaluate you as an employee / part of their team - give them a reason to remember you.

You are required to buy/register a domain name and arrange web hosting for your portfolio. Research web hosting options and choose one that fits your requirements. Your site must be "live" - deployed to your host and reachable at your URL - for it to be graded.

SUBMISSION REQUIREMENTS

- Github repo: readme.md file, master branch, design branch and development branches
- correct structure for a web project (css, js, sass, images, video folders and index.html)
- Name the dev branches appropriately per feature IE des.yourinitials.artwork, dev.yourinitials.script
- Submit the repo link via FOL dropbox ONLY THE MAIN BRANCH WILL BE GRADED
-Submit your url to the FOL Dropbox as well.

DUE DATE & RUBRIC

The Portfolio is due on Friday, February 16th at 5pm EST(Eastern Standard Time). It is your responsibility to check with each course FIP breakdown (course book) to see the specific submission requirements for each individual course in IDP

25% of Term Grade

Javascript Functionality (GreenSock Animation, Scroll Effects, ScrollTrigger, custom js, etc.): 5 marks
JS Modules and Bundling: 5 marks
SASS (Modules, Minified): 5 marks
Github Workflow / Readme: 5 marks
AJAX enabled form: 5 marks

Folder Structure

• .html pages

• images folder

• SASS directory

• styles/css folder

• video

DO NOT INCLUDE THE working_files FOLDER IN YOUR REPO!

PLEASE NOTE: Your project will not be graded unless you adhere to what has been outlined for you in this document. Individual practices that were given to you during the semester are applicable here as well.

i.e: A photoshop/XD/AI file with improperly labeled layers, code not in between body tags, lack of naming conventions, etc.

Naming convention for submission:

LastName_FirstName_Portfolio - e.g.) smith_joe_portfolio

HW #3 – VUE AND AJAX ASSIGNMENT

The objective of this assignment is to introduce students to the basic concepts of connecting a Vue.js frontend to a Lumen API they will create in their web development class. The API will contain a sorted listing of entries and a single entry endpoint. Working in pairs students will gain hands-on experience in creating a simple web application that fetches and displays data from their own API.

Assignment Requirements

Create a basic Vue.js application with a single HTML file and a Vue instance.

Use the Vue instance to fetch data from the Lumen API.

Display the sorted listing of entries on the webpage.

Implement a feature that allows users to click on an entry to view the details of that specific entry.

Display the details on the webpage without navigating to a different page.

Ensure proper error handling for failed API requests.

The page must be responsive from mobile to desktop.

Additional GreenSock enhancements and features are strongly encouraged.

SUBMISSION REQUIREMENTS & DUE DATE

- Homework must be submitted by Week 12 (Check dropbox date and time)
- Github repo: readme.md file, master branch, development branch(es)
- Name the branches appropriately per feature IE des.yourinitials.artwork, dev.yourinitials.script
- Submit the repo link via FOL dropbox
- DO NOT SUBMIT ZIPPED FILES, ONLY THE MASTER BRANCH WILL BE GRADED

GRADING RUBRIC /15
HTML/CSS /2 Design /2 GitHub Best Practices /1 First Data Call /2 Populate List /3
Second Data Call /2 Populate Second Box /3

FINAL INTEGRATED PROJECT

The objective of this project is for teams of students to design, develop, and pitch a professional marketing and advertising campaign and website for their Final Integrated Project's subject matter - Foundation Sixty6

ASSIGNMENT REQUIREMENTS

Create a mobile-first responsive site adhering to all modern web standards. Coding best practices must be followed e.g. SASS modules, minifying CSS, using JS modules, and bundling.

Use a front-end framework (Vue.js) or the template element, when building the project. Use dynamic data where required with the Fetch API; use modern web techniques (CSS, JavaScript, etc) to implement site enhancements for improved UX and UI. Use the GreenSock Library to add visual enhancements and interactivity.

You'll need to include a form that uses your Lumen API to store user information in an appropriate table via an AJAX call. This could be an input for a mailing list, a signup page for users, etc. Be sure to handle any errors and success states returned from your Lumen API and update the view as appropriate using JS. For example, consider how to handle incorrect usernames/passwords, invalid emails, successful submissions/logins, etc.

SUBMISSION REQUIREMENTS & DUE DATE

The due dates for the presentation are listed below. All time is in Eastern Standard Time (EST)

Week 4: 'Show and Tell' Meeting – Friday, January 26th, 2024, 1:00 pm

Week 8: Alpha Stage – Friday, March 1st, 2024, 1:00 pm

Week 13: Beta Stage – Friday, April 5th, 2024, 1:00 pm

Groups will be graded with a single grade for all group members (plus anonymous peer evaluation)

Please see the FIP document for additional information about grading and rubrics.