WDD 330 Personal Project

This document serves as your final course assessment.

Introduction

Name: Chelsea Thomson
Video Link:
Application Link: WDD330 Portfolio (snowmoto.github.io)

Course Outcomes

The following are the course outcomes of WDD 330:

1. Become more efficient at applying your innate curiosity and creativity.
2. Become more dexterous at exploring your environment.
3. Become a person who enjoys helping and learning from others.
4. Use a divide and conquer approach to design solutions for programming problems.
5. Finding and troubleshooting bugs you and others will have in the code you write.
6. Developing and debugging HTML, CSS, and JavaScript programs that use medium complexity web technologies.

To complete this course, you need to demonstrate your skill in these areas. Outcomes #1-5 demonstrate your personal development and are most easily shown through self-assessment and sharing experiences. Outcome #6 demonstrates your programming skill and is shown through code and experience in projects.

Skill Development Outcome

Developing and debugging HTML, CSS, and JavaScript programs that use medium complexity web technologies.

This outcome is demonstrated by your skill in the following learning objectives:

Objective	%	Description
JavaScript	25%	Robust programming logic is demonstrated. For example, validating the screen data, looping through an array of JSON data to display to the screen, creating and using events, changing element styles with JS, changing element classes to use different CSS rules.
Third-party APIs	15%	APIs are used effectively, including APIs that provide rich JSON data.
JSON	15%	Demonstrate skill processing JSON data to dynamically update the website.
CSS	15%	Appropriate use of Transforms and Transitions. For example: Add round the edges to DIV, add shadows. enlarge an input field on focus, and shrink it on blur, Add borders. CSS should subtly add style to a page.
Events	15%	Use events to enhance the user experience. For example, increase the size of the input field on focus or add a shadow. React to a button click. Initialized the page with data once the onload event triggers.
Local Storage	5%	Local storage is used effectively.

These learning objectives are rated on the following scale:

Rating	Description
Unsatisfactory	Very little if any work was shown in this area.
Developing	The learning objective was shown in very basic ways.
Proficient	Effective use of the learning objective was shown in multiple places.
Mastery	Extensive use of the learning objective was shown in non-trivial ways in many places in the code.

For each learning objective, discuss how the topic was used in your application. List several examples of places where the topics are demonstrated.

The following is an example of what is expected:

Learning Objective	Description	Where can this be seen in your application?
CSS	*I spent a lot of time choosing colors that would complement each other.* *I used CSS to make the input field bigger when it received the focus and to shrink it when it lost focus.*	*This can be seen on the home screen for each input field.*
	Images are enlarged on hover.	*The recipe detail pages have this effect.*
	The search results have alternating colors for the rows for readability.	See the home page after a search is successfully run.

In the following table:

1. Describe how the topics are used.

Have someone test your links to make sure they are accessible by the grader. These links will be to your final personal project.

Feel free to add more rows to this table if needed.

Learning Objective	Description	Where can this be seen in your final personal project application?
JavaScript & Local Storage	Fetching API for poster, title, year, and plot of the movies and creating the function for searching for information. Within the function you will see that I have used a DOM incasing a div for the information to load at once and not stagger loading.	Located in the folder scripts file name movies.js It will be lines 1-61. On the page you will see the search bar and be able to enter the title of the movie and click search.
	Creating a local storage function for likes and favorites. This is clickable function to like and unlike the chosen favorite movies. It includes a number to see how many likes there are for that movie.	Located in the folder scripts file name movies.js. It will be lines 63-116. On the page once the images have loaded in you will see the heart and number below each poster.
Third-party APIs	I used rapid API for movies that allows 1000 times of used per day.	Located in the folder scripts file name movies. Lines 1-7.

JSON	Rapid API has a JSON file that I am pulling information from to display the movies.	Located in the folder scripts file name movies. Lines 1-7.
CSS	Created 3 screen views	Located in folder styles and files names: base.css, tablet.css, desktop.css and normalize.css from a default file used from previous classes.
	I created image animation that is similar to a slide show that fades out and then displays the next image. Embedded a gif as a background for the search area and title of the HTML.	Located in folder styles and files names: base.css. Lines 52-108 Located in folder styles and files names: base.css Lines 205-220
	Created a card flip animation for the posters to display the information on the backside.	Located in folder styles and files names: base.css Lines 135-175
Events	Click the search to find movies.	Located in the folder scripts file name movies. Line 9
	Click on the heart for liking favorite movie.	Located in the folder scripts file name movies. Line 116
	Flipping posters over to see backside of information.	Located in folder styles and files names: base.css Lines 135-175