# WDD 330 Personal Project

This document serves as your final course assessment.

## **Introduction**

**Name**: Austin Shiflet

**Video Link**: [Insert your video link here]

**Working Application Link**: https://austinshiflet14.github.io/wdd330/final/

**GitHub Source URL**: https://github.com/AustinShiflet14/wdd330/tree/main/final

**Trello Board URL**: https://trello.com/b/axgdchfV/my-trello-board

## **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 | Used JavaScript to fetch and display recipe data dynamically, manage user interactions, and handle DOM updates. | JavaScript powers the search bar, displays results, and updates recipe details dynamically when a user selects a meal |
|  |  |
|  |  |
| Third-party APIs | Integrated with external APIs (Spoonacular and TheMealDB) to retrieve recipes, nutrition info, and meal details. | The recipe search feature and meal detail pages use data pulled from these APIs. |
|  |  |
|  |  |
| JSON | Parsed JSON data from API responses to display recipe names, images, and ingredients. | The search results and recipe details are built from the parsed JSON returned by the APIs. |
|  |  |
|  |  |
| CSS | Styled the site to be visually appealing and responsive across devices. Used hover effects, rounded borders, and transitions. | Seen hover effects on recipe cards, and the mobile-friendly layout. |
|  |  |
|  |  |
| Events | Added event listeners for user interactions, such as searching recipes, clicking to view details, and favoriting meals. | Triggered when the user types in the search bar, clicks a recipe card, or taps the heart button to favorite/unfavorite. |
|  |  |
|  |  |
| Local Storage | Saved and retrieved user favorites so they persist even after refreshing the page. | The “Favorites” page uses local storage to display the user’s saved meals. |
|  |  |
|  |  |