# WDD 330 Personal Project

This document serves as your final course assessment.

## **Introduction**

**Name**: Nelson Muchonji Bifwoli

**Video Link**: <https://www.loom.com/share/9b6c0f457f734feaa72b0b3327c0e757?sid=9c9dc3f1-f277-4a75-a0bb-e65698e4df04>

**Application Link**: <https://weatherappwdd330.netlify.app/>

## **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 | Creating a function that will inject data into the results page. | On the results page to display the weather info such as temperature, appropriate weather icon and city and country. |
| Creating a function that will pull data and suggest to the app user what to wear | On the result page, it will display the icon of the cloth one should wear with a line of recommendation, such as wear something light. |
| Async function to show the weather forecast | This will show the weather for the day, and this will also be displayed on cards for a whole week capturing sunset, sunrise, city, country, weather description and weather icon. |
| Third-party APIs | OpenWeather API to fetch data for a city | Use longitudes and latitudes to get weather for a city and display them on the results page. |
|  |  |
|  |  |
| JSON | Local JSON file for weather comments | Used this to suggest what one was to wear based on the weather of a city they had searched for. |
|  |  |
|  |  |
| CSS | Button colors, font style and enlarges on hover | This is on the home screen. |
| When someone enters a city that doesn’t exist, the error message displayed background and font style | On the home screen when one searches for a city that doesn’t exist. |
| Weather cards displaying results for the week. The weather info captured for the week including the sunset and sunrise. The cursor changing on pointer and results for each card captured when selected. | This is shown on the results page. |
| Events | Use event listener to get the latitude and longitude co-ordinates of a place and then display them on a new window showing the weather forecast based on those weather of the co-ordinates provided. | Can be seen both on home screen when you click button to search and the result page to display the results. |
| Event listener for when a city that doesn’t exist is searched. This is on click of the button and capturing the entered text on the location search box. And the close button to close the information displayed. | On the home screen when wrong information is entered of a city. |
| Event listener to the alert button when wrong city entered | On the home screen to close the alert message. |
| Local Storage | setLocalStorage for the weather forecast which will be a list capturing the information provided on the location search. | Home screen for the city search and the results provided on result page |
| getLocalStorage for the weather forecast | Displayed on the results page. |
|  |  |