# WDD 330 Personal Project

This document serves as your final course assessment.

## **Introduction**

**Name**: [NEHIKHARE EFEHI]

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

**Application Link**: [https://nehiz.github.io/my-final-project/]

## **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 | I used javaScript to add a click event listener to the forecastButton. | This is seen when the button is clicked, the getForcast() function is called.. |
| I used javaScript to process and Display Data | In the getForecast(), I process the data by extracting relevant information(date, temperature, weather condition) from each forecast item. This is displayed in the forecastResult element. |
| I used javaScript to add custom Clothing Recommendation. | The getRecommendations(weatherCondition) function returns clothing recommendations based on the weather condition. This is displayed in the weather-appropriate clothing. |
| Third-party APIs | OpenWeatherMap API | In the fetchWeather(location), I use the OpenWeatherMap API to get weather data for the specified location. This can be seen when a user input a city and click on the get weather. The api takes the city and search for its current weather. It then returns that weather for us which is then displayed in the weather overlay session that is seen below the get weather button. |
| API location and API KEY:  https://api.openweathermap.org/data/2.5/weather?q=${location}&units=metric&appid=${apiKey} | It is used to fetch current weather data from the OpenWeatherMap API. It is displayed as the current weather output. |
| API location and API KEY:  https://api.openweathermap.org/data/2.5/forecast?q=${location}&units=metric&appid=${apiKey} | It is used to fetch 5-days forecast data from OpenWeatherMap API. It is displayed as the 5-days forecast result. |
| JSON | I used JSON to create a response. | If the fetch data request is unsuccessful I.e (!response.ok), an error message is thrown. If successful, it returns the weather data in JSON format. |
|  |  |
|  |  |
| CSS | I used css to add an svg of weather to the page to create a sychronization to the purpose of the app. | Can be seen at the hero session |
| I Used css to add colorto the page | This can be seen in the hero session of the page were we have weather forecast. |
| I Used css for the forecast styling to give the .five-days-forcast section a fixed height, scrollable content and a dark background.  Also, selectors were used to target specific elements. | Can be seen in the 5-days-forcast  For example, .flex targets elements with the class name “flex”, .wrapper targets elements with class name “wrapper” |
| Events | I Used javaScript to the event handler getWeatherAndRecommendation() | This fetches weather data and clothing recommendations based on user input. The result is display in the current weather. |
| I used javaScript to create getForecast() | It fetches a 5-day weather forecast and output it in the dashboard. |
| Applied event listeners to the buttons to listen to events that may happen to them. | It is seen on the page when we click on the get weather button, it displays results in the get weather session. |
| Local Storage | Used local storage to store date and timefor the next five days forecast. | Whenever we click on the next five days forecast, it is being stored in the API local storage.it gives us the time alongside the weather forecast. |
|  |  |
|  |  |