

WDD 131

[Home](#) [W1](#) [W2](#) [W3](#) [W4](#) [W5](#) [W6](#) [W7](#)

W03 Assignment

Overview

Apply the concepts from the learning activities in the design and development of a web page. The subject of the page is about a **country or place** where you live or have visited or wish to visit. There are specific, functional and developmental requirements for the layout, design, and content of the page.

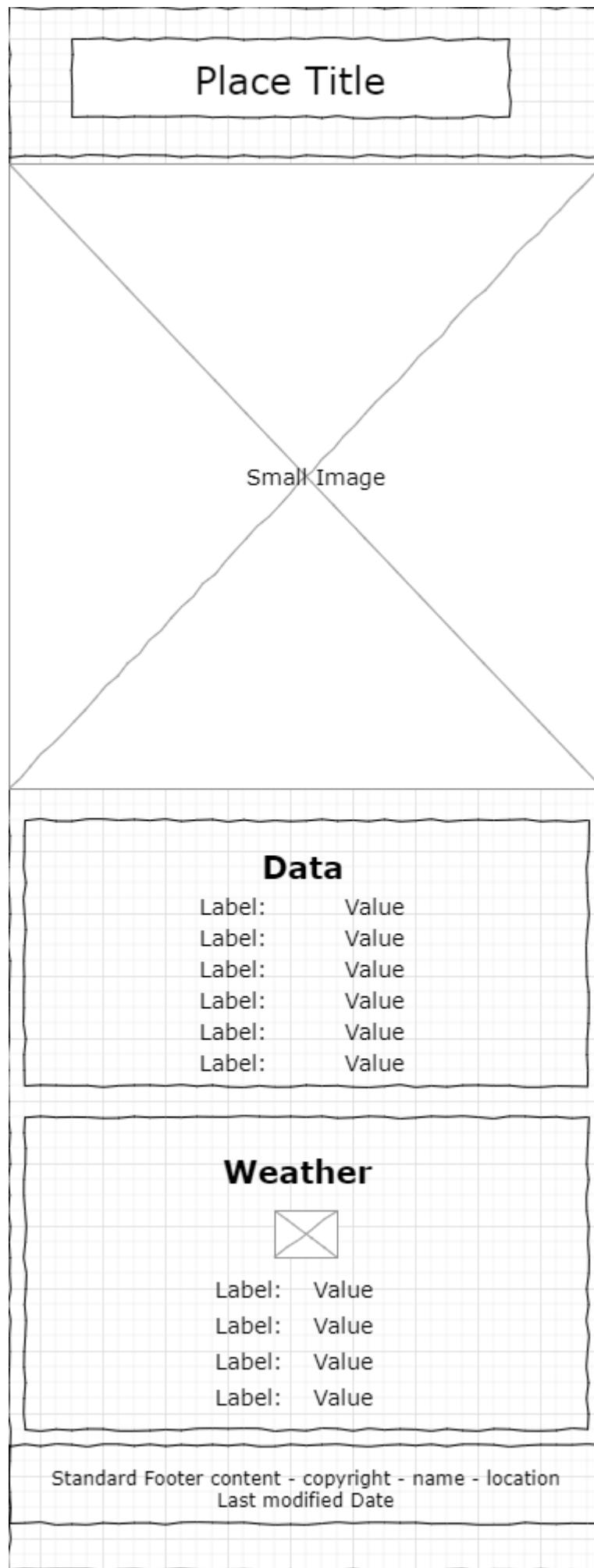
▼ Associated Course Learning Outcomes

1. Develop responsive web pages that follow best practices and use valid HTML and CSS.
2. Demonstrate proficiency with JavaScript language syntax.
3. Use JavaScript to respond to events and dynamically modify HTML.
4. Demonstrate the traits of an effective team member (such as clear communication, collaboration, fulfilling assignments, and meeting deadlines).

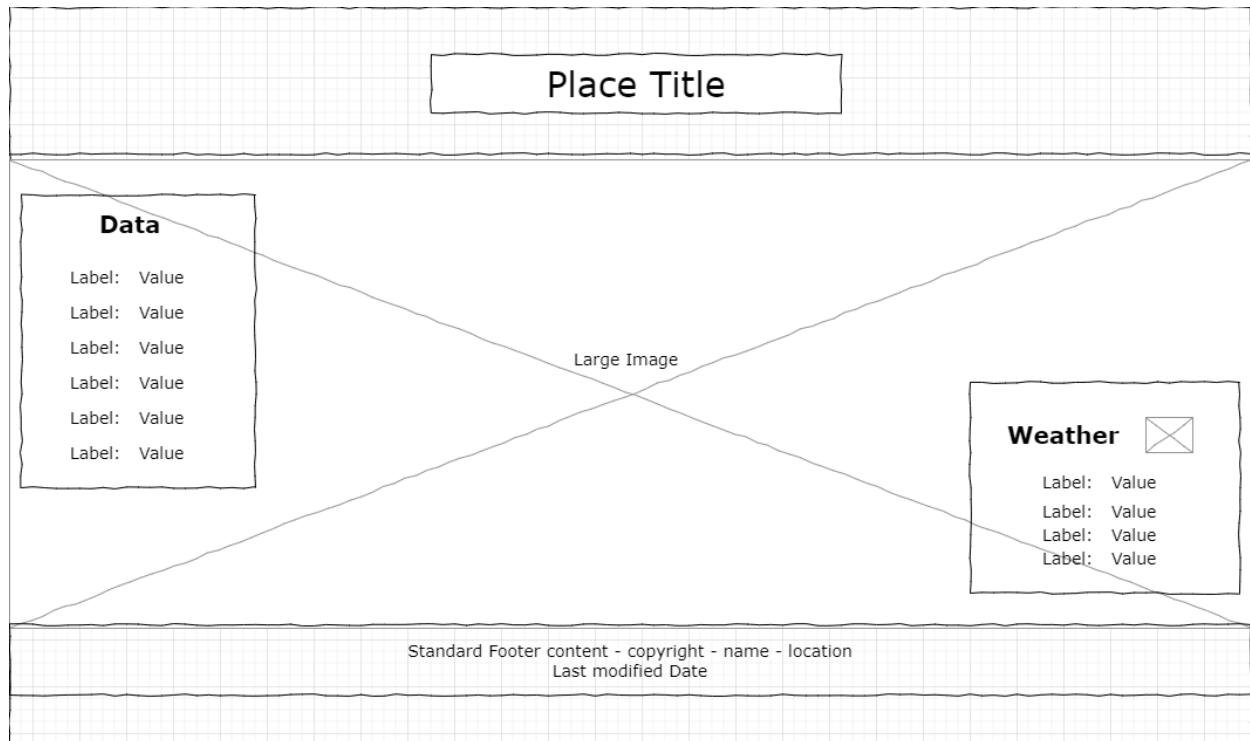
Task

Design and build a single page based upon the following **wireframe** as the layout and content location guide:

▼ Wireframe Designs and Example Screenshots



Wireframe design of the page in mobile view



Wireframe design of the page in a wider view

Madagascar



Data

Area: 587,041 sq km

Population: 27,691,018

Capital: Antananarivo

Languages: Malagasy, French

Currency: Malagasy Ariary

Time Zone: UTC+3

Calling Code: +261

Internet TLD: .mg

Weather



Temperature: 10 °C

Conditions: Partly Cloudy

Wind: 5 km/h

Wind Chill: 9.8 °C

©2024 🌸 Rubia Magdelena Francesco 🌸
Madagascar

Last Modification: 02/08/2024 22:47:47

Example Page in Mobile View

Madagascar

Data

Area: 587,041 sq km
Population: 27,691,018
Capital: Antananarivo
Languages: Malagasy, French
Currency: Malagasy Ariary
Time Zone: UTC+3
Calling Code: +261
Internet TLD: .mg

Weather

Temperature: 10 °C
Conditions: Partly Cloudy
Wind: 5 km/h
Wind Chill: 9.8 °C

©2024 🌸 Rubia Magdelena Francesco 🌸 Madagascar
Last Modification: 02/08/2024 22:47:47

Example Page in a Wider View

You can create your own, free **wireframe** designs using the online application [jGraph-Diagrams.net](https://jgraph-diagrams.net) or you can install on your own system from [JGraph](https://www.jgraph.com/jgraph.html). There

<https://byui-cse.github.io/wdd131-ww-course/week03/assignment-03.html>

5/9

are many other viable choices including [Moqups](#). Note that you will need to create your own wireframe designs later in the course.

Instructions

Step 1: File and Folder Setup

1. In your **wdd131** local repository folder, add a new HTML file named "**place.html**".
2. Add a CSS file or files in the appropriate subfolder.

You may elect to use one CSS or multiple CSS files for this assignment to organize your media queries.

3. Add a JavaScript file to the appropriate subfolder.

Step 2: HTML

1. Include the standard HTML document and **<head>** elements.

Refer to the [course development standards](#) if you need to review standards.

2. Be sure to link your **css** file(s).
 3. Reference your **js** file using a **defer** attribute.
 4. Create the structure and layout for the page by referencing the wireframe and example screenshots given above.
- Important:** It is required that you match the wireframe components and basic positioning of those components as shown in the wireframes.
5. The "**hero**" **images** must utilize the **srcset** responsive design method to provide different images based on screen size. You can choose to implement this requirement by using either the **picture** element or a basic **img** approach. Ensure that the image subjects and dimensions support the subject, design, and layout of the page.

Step 3: Assets

You have the option to use your own images or select from stock photos available on [pexels.com](#) or [unsplash.com](#), as well as other resources.

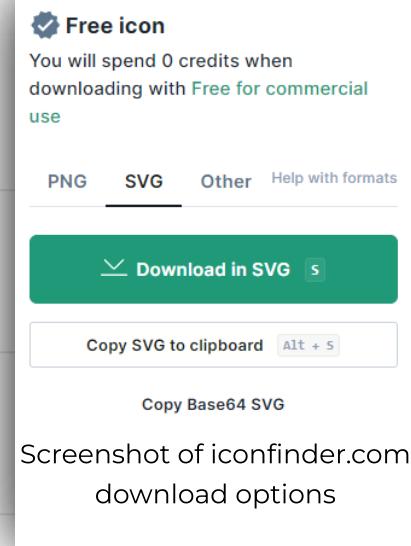
Ensure compliance with licensing requirements and provide appropriate credentials if necessary. Many images on these sites are free to use and do not require [attributions](#), however it is important to review the licensing terms.

1. Use the **WebP** image format for the hero images.
2. Use the **SVG** image format for the weather icon in the mobile view.

An svg can be downloaded via most icon service sites like [iconfinder.com](#).

3. Use a emoticon or equivalent for the weather icon ☁ in the wider view by the "Weather" heading.

The weather will be static at this point and so try to pick an image/icon that matches your stated conditions.



Step 4: CSS

Important: You are responsible to implement good design principles of alignment, color contrast, proximity, repetition, white space, and general usability in all of your work.

1. Select a color schema that supports the country or place

For example, the provided screenshots above use the official colors of the flag of Madagascar.

2. The typography choice is up to you.
3. The minimum styling requirements include:

1. Matching the **layouts** shown in the wireframes
2. Using a **media query** or queries.
3. Using **global CSS variables**.
4. Using a **pseudo element** for the weather icon in the large view located after the heading "Weather".

Step 5: JavaScript

1. The **footer** of the page must include the current year and the date the document was last modified just like the other assignments. You can copy that code into this page and either use the existing JavaScript file or use your page specific JavaScript file to support this page's footer.

2. In your JavaScript file, provide the code to support the following requirements:

1. Use JavaScript to display the **windchill factor** in the "Weather" section of the page as shown in the examples. The windchill factor should be calculated and displayed when the page loads.
2. At this point in the course, go ahead and **define variables** that use static values for the temperature and wind speed, matching the static, displayed values you have in your weather section content.

The next course will cover how to use third-party APIs to get real-time weather data.

3. Write a function named "**calculateWindChill**" that returns the windchill factor when passed the appropriate arguments (temperature and wind speed). The function should use one line of code that returns the result of the calculation of the wind. Of course, your formula should be based upon the location's preference of units ($^{\circ}\text{C}$ or $^{\circ}\text{F}$).

Using AI to help determine this formula might be a good approach.

4. Do **not** call the **calculateWindChill** function unless the conditions are met which are as follows:

Viable Wind Chill Calculations		
	Metric	Imperial (English)
Temperature	$\leq 10 \text{ } ^{\circ}\text{C}$	$\leq 50 \text{ } ^{\circ}\text{F}$
Wind speed	$> 4.8 \text{ km/h}$	$> 3 \text{ mph}$

If the conditions are not met, then display "N/A", which means not applicable.

Of course, your code is using static inputs, nonetheless, your code should have this provision coded in preparation for dynamic, real-time inputs.

Step 6: Testing

1. Continuously check your work by rendering the page locally using Live Server in VS Code.
2. Use the browser's **DevTools** to check for JavaScript runtime errors in the console or click the red, error icon in the upper right corner of DevTools.
3. Use DevTools [CSS Overview](#) to check your color contrast.
4. Generate the DevTools **Lighthouse** report and run diagnostics for Accessibility, Best Practices, and SEO in both the mobile and Desktop views.

It is best to test your page in a **Private** or **Incognito** browser window.

Do not worry if the Best Practices report comes back low for the use of a content delivery network. We are uses the Church's media library on purpose.

Audit and Submission

1. Commit your local repository and push or upload your work to your GitHub Pages enabled wdd 131 repository on GitHub.
2. Use this [audit tool](#) to check some basic HTML and CSS standards and requirements.
3. **Share** your work by posting your URL in the course's **Microsoft Teams weekly forum** and comment on your peers' work by providing constructive feedback.
4. Return to Canvas and submit your the correct GitHub Pages enabled URL.

<https://your-github-username.github.io/wdd131/place.html>