

# FWDP 1000 – Web Development 1

## Project #1 – Landing Page

Due Date: January 27, 2024, end of day (11:59pm).

Marks: 15 points. See the rubric in the Learning Hub for details.

### Project Description

Create a single webpage to serve as the Landing Page for your personal domain. The landing page should include links to your projects and assignments throughout the FWD program. This project will consist of a single custom designed HTML page with an external stylesheet.

You can consider this version 1 of your portfolio website. During your Web Portfolio course, you will create a portfolio website based on your goals after the FWD program.

See examples of past projects here: <https://wp.bcitwebdeveloper.ca/landing-page-examples/>

To make the process easier, it is recommended that you do a quick sketch or wireframe of your intended design. This will not be submitted but will help you get started.

### Project Requirements

Content -- 1 point

The following content should be included on your Landing Page:

- Your name.
- A short paragraph or sentence about yourself.
- Links to your school projects. For projects not completed, use “#” as the href value or use placeholder text like “Coming Soon”. Below are some example categories and projects. You can organize the links as you see fit and include any projects you want to include.
  - HTML / CSS Projects:
    - Country Website, Business Website, etc.
  - JavaScript Projects:
    - JavaScript Game, Movie Database, etc.
  - WordPress Sites:
    - Vancouver, Mindset, School
  - Online Stores:
    - Shopify Store, WooCommerce Store

### HTML Code Quality -- 3 points

- Create one index.html file.
- Follow the best practices learned in class to write clean and organized HTML.
- Use semantic HTML elements whenever appropriate.
- *Suggestion: Use an HTML code validator before submitting.*  
Example: <https://validator.w3.org/>

### CSS Code Quality -- 3 points

- Create at least one CSS file.
- Follow the best practices learned in class to write clean and organized CSS.
- The landing page should be responsive. The design and code should be done mobile-first and work properly from 320px wide to at least 1920px wide.
- *Suggestion: Use a CSS code validator before submitting.*  
Example: <https://jigsaw.w3.org/css-validator/>

### GitHub -- 2 points

- Add me to your repository (username: gbad5) or set your repository to public, either is fine.
- Create a repository for your landing page with a main/master branch.
- Have at least 2 different commits on the main/master branch.
  - Write descriptive commit messages.
- Main/Master branch must be the latest version and must match your submitted zipped file.

### Accessibility & Performance -- 3 points

- Follow the best practices learned in class to create an accessible website.
- Any images or videos used should be optimized for the web.
- *Suggestion: Use accessibility tools and an accessibility checklist before submitting.*  
Example: <https://www.a11yproject.com/checklist/>

### Domain and Hosting -- 2 points

- Upload your Landing Page to your hosting account through FTP.
- Connect your domain to your hosting account. The Landing Page should serve as the current home page for your domain.
  - Example: name your file index.html and place it in the public\_html folder on the server.

### Reflection & Submission -- 1 point

- Zip up your landing page files and upload them to the Learning Hub.
- Answer these questions on the Learning Hub when submitting the assignment:  
(See the next page for example answers.)

1. What is the URL of your landing page?
2. What is the URL of your GitHub repository?
3. What is something you learned while working on the landing page? It can be about web development or your working process generally.
4. Was there anything you were unable to do on your landing page that you would have liked to do? Or anything you found particularly challenging?

This is an example of answers that would receive **full** points on the Reflection portion...

1. gabbiebade.com
2. <https://github.com/gbads/landing-page>
3. I learned that using the developer tools can be very helpful to debug issues I was running into. For instance, I couldn't get my projects section layout working and then checked in dev tools to see that I had display: flex on the wrong element. Also, I found that having a sketch or plan for my design made it easier to start the CSS.
4. Using flex was challenging to get the layout I wanted in my projects section but I managed to get it working. I thought about doing a more advanced layout but I didn't know how to do that yet so I am hoping we will learn it later. One thing I tried to do but wasn't able to do was adding animation to the text with my name at the top. I saw this on some examples but I couldn't get the JavaScript and CSS to work together.

This is an example of answers that would receive **half** points on the Reflection portion...

1. gabbiebade.com
2. <https://github.com/gbads/landing-page>
3. I learned how to use flex better.
4. I wanted to add images but I couldn't make them work.

This is an example of answers that would receive **no** points on the Reflection portion...

My url is gabbiebade.com