





Search Topics

 Bookmarks	1
 Course Schedule	1
Table of Contents	4
Welcome - Start Here	✓
Syllabus	✓
Help Forum	✓
Lessons and Assignments	4
Orientation	✓
Web Development Process	✓
Debugging	✓
Enhancing the Site	1
Layouts	1
Collaborative Coding	1

Enhancing the Site



0 % 0 of 1 topics complete

Enhancing the Site

Assignment

 Due July 20 at 11:59 PM

 Available on Jul 6, 2025 12:01 AM. Access restricted before availability starts.

 Available until Jul 27, 2025 11:59 PM. Access restricted after availability ends.

Objectives

- Optimize and display images with multiple sizes and accessible markup
- Create and apply a favicon to enhance site branding
- Build a semantically structured and accessible contact form
- Use HTML tables for meaningful tabular data with accessibility features
- Apply the box model, float-based layouts, and white space for improved design clarity
- Organize CSS using multiple stylesheets for modular control

Description

Your Launch Studio training site is evolving! In this module, you'll refine the Home page with well-integrated images and build a fully functional Contact Us page using accessible and professional design techniques.


You'll start by enhancing the **Home page** with three optimized images, each with both thumbnail and full-size versions. You'll float these images inside your main

content sections to improve layout and composition. You'll also create a favicon to give your site a more complete and polished appearance.

Next, you'll design a **Contact Us page** that includes a fully accessible form and a clean, well-styled data table. You'll focus on layout best practices like using white space, centering content, and constraining widths to improve readability and professionalism. The Contact Us page will submit to the `success.html` page. Upon successful submission of the contact form, the server should return the valid HTML5 `success.html` confirmation page styled to match the rest of the site.

As always, if you need help, check in with the **Help Forum** to troubleshoot and share ideas.

Requirements

 **Tip:** The title of each module part includes a link to a lesson with important instructions and examples. Follow these links closely — they're your roadmap to building a successful project. And remember, if you're stuck, check the **Help Forum**!

Lesson 1: Adding Images (Enhance Home Page)

- Add three meaningful images, one per main section of the Home page

Image Requirements

- A full-size version
- A thumbnail version (displayed on page and linked to full-size)
- alt text for accessibility
- Float each image inside its corresponding section to improve layout and text composition
- Constrain image dimensions using CSS
- Create and apply a favicon using the `<link rel="icon">` tag in the `<head>`
- Organize all assets in an `images/` directory

Lesson 2: HTML Forms and Tables (Contact Us Page)

Create `contact.html` and link it from your site navigation.

Requirements

- Build a structured contact form with appropriate fields

Form Requirements

Required Elements

- `<form>` tag with `name="contact"`, `method="get"`, and `action="success.html"`
- `<fieldset>` and `<legend>` used for grouping
- `<label for="">` tags for each form field
- Single-line text input
- Multi-line `<textarea>`
- Radio buttons
- Checkboxes
- Dropdown menu (`<select>`)
- Submit and reset buttons

Form Styling

- Width should be 60–70% of the page
- Form should be horizontally centered using CSS
- Use box model styling: padding, margin, and borders

Form Confirmation

- Create a `success.html` page with a confirmation message for form submission

Table Requirements

Required Elements

- `<table>`, `<caption>`, `<thead>`, `<tbody>`, `<tr>`, `<th>`, and `<td>`
- `<th>` elements must include the `scope` attribute
- Table content must be true tabular data, not layout

Table Styling

- Width should be 60–70% of the page
- Apply CSS for padding, margin, borders, and text formatting

Suggested Table Ideas

Your table should present real tabular data — not be used for layout. Here are a few ideas to consider:

- A **training schedule** with dates, topics, and instructors
- A **browser compatibility chart** for web features or CSS properties
- A **team contact sheet** showing names, roles, and support hours
- A **feature comparison** table for software tools or site components
- A **project checklist** with task names, deadlines, and completion status



Lesson 3: The Box Model and Floated Elements

- Use floats on the Home page to place images alongside text in each section
- Apply clearfix or other techniques to prevent float-related layout issues
- Use the box model (padding, margin, border) to manage spacing and separation
- Keep layout-specific styles in `layout.css`, separate from `style.css`

Contact Page Layout Enhancements

- Center the content container
- Limit width to improve white space and readability


Deliverables


Use the lesson examples as your blueprint — they reflect the standards our team expects in each project.

Submit the following artifacts as links to demonstrate successful development:



Project Home Page

-  **GitHub Repository URL** — A working repository named `webtools` containing your project files and a `README.md` file.

-  **GitHub Pages URL** — A published version of your site, deployed from the main branch using GitHub Pages.

 **Be sure both links are submitted before the due date.**

Submissions without working GitHub URLs may receive no credit.

Link to GitHub repository with the following:

- `index.html` updated with floated images
- `contact.html` with a structured form and data table
- `success.html` with a confirmation message for form submission
- `style.css` (from previous module)
- A new `layout.css` file for layout-specific styles
- A favicon and an `images/` folder with optimized assets

Grading Rubric

This rubric defines grading criteria for images, forms, CSS layout, and deployment.

Each row describes expectations for Poor, Good, and Excellent levels.

Criteria	Poor (50 pts)	Good (80 pts)	Excellent (100 pts)
Adding Images	Images missing or poorly implemented; lacks alt text or folder organization; no float styling.	Some images present; thumbnails or float behavior may be incomplete; alt text may be inconsistent.	All required images included with alt text, thumbnail/full-size links, floated placement, and organized in an <code>images/</code> folder.
HTML Forms and Tables	Form or table is incomplete, invalid, or missing accessibility features.	Most required form/table elements present but with structural or labeling issues.	Fully accessible, well-structured form and semantic table meet all technical and accessibility expectations.
The Box Model and Floated Elements	Layout is disorganized; float-related issues not	Some layout improvements applied, but styles	Clear use of the box model, float styling, spacing, and a well-

Criteria	Poor (50 pts)	Good (80 pts)	Excellent (100 pts)
	resolved; spacing is inconsistent or missing.	may be mixed or incomplete; float clearing is partial.	structured layout.css separate from style.css.
Project			
Repository & Deployment	GitHub repo or GitHub Pages link		
(Required – submission cannot be graded without it)	is missing or broken; work cannot be evaluated.	—	—