

Project 3

Web development and deployment

Timing: 20 to 35 hours

Project overview

In this project, student teams use their client-approved design documents for the technical build of a website. Teams will learn about technical features to enhance the look and feel, consistency, and usability of their websites. They will also learn how to conduct technical and usability tests of their client websites and make any necessary revisions.

The main focus of this project is for students to develop reusable content, templates, and Cascading Style Sheets (CSS) based on the page layout described in their design documents and production storyboards. The emphasis is on collaboratively building websites for their clients by using web standards to complete their home pages and content pages. Each team will test and deploy its site, make any necessary revisions based on client review, and then redeploy the site. Teams will create a presentation for their clients explaining how the site accomplishes the client goals.

Student product: Client website

Note: Portions of the Web development and deployment project align to the Adobe Certified Associate, Web Communication objectives. Within the instruction steps and technical guides, the specific learning objectives for the exam are referenced with the following format: ^{1.1}

Project objectives

At the completion of the project, students will have developed the following skills:

Project management skills

- Following and executing a project plan
- Following and executing a wireframe and prototype
- Following up and following through on roles and responsibilities
- Defining and prioritizing tasks
- Producing deliverables and meeting deadlines
- Managing files and using file-naming conventions
- Executing a review and redesign cycle
- Executing quality assurance tests

Design skills

- Providing consistency and accessibility

- Providing universal navigation
- Integrating images, text, and multimedia
- Planning graphics and multimedia based on needs and audience
- Creating web pages using web standards
- Designing with CSS
- Organizing pages with CSS
- Understanding HTML
- Using basic HTML
- Creating templates and reusable designs
- Adapting content for readability and emphasis
- Designing for multiple screens
- Considering screen size and device requirements

- Incorporating layout and color consistently
- Designing a quality assurance test

Research and communication skills

- Communicating ideas clearly
- Presenting a website to a group
- Taking notes on critique
- Providing meaningful but not overly critical feedback
- Listening and interpreting information and feedback
- Demonstrating the realization of redesign goals
- Conducting usability analysis
- Creating quality assurance tests

Technical skills

Fireworks

- Optimizing images
- Creating buttons
- Creating navigation bars
- Creating pop-up menus

Photoshop

- Resizing images

- Rotating images
- Cropping images

Dreamweaver

- Understanding the Dreamweaver workspace
- Setting up a web project
- Setting document properties
- Using the Insert, Files, and Assets panels
- Inserting images
- Inserting text
- Creating lists
- Creating data tables
- Creating image maps
- Using CSS starter layouts
- Using CSS text styles
- Creating, modifying, and troubleshooting CSS
- Creating templates
- Developing a website for multiple screens
- Checking files in and out
- Inserting HTML5 video
- Checking links
- Testing in Adobe BrowserLab
- Publishing web files

Project materials

- Adobe Fireworks CS6 installed on all machines
- Adobe Photoshop CS6 installed on all machines
- Adobe Dreamweaver CS6 installed on all machines
- Dreamweaver guide: Overview of the Adobe Dreamweaver CS6 workspace
- Dreamweaver guide: How to set up a local root folder and site structure
- Dreamweaver guide: How to set document properties
- Dreamweaver guide: How to use CSS starter layouts
- Guide: File-naming conventions
- Worksheet: File management
- Dreamweaver guide: How to manage files and links by using the Files panel
- Guide: Introduction to HTML and CSS
- Dreamweaver guide: How to lay out a web page with CSS
- Dreamweaver guide: How to create and edit a CSS rule
- Dreamweaver guide: How to create lists
- Presentation: Reusable design
- Dreamweaver guide: How to use the Assets panel
- Dreamweaver and Fireworks guide: How to create interactive navigation
- Fireworks guide: How to optimize images
- Photoshop guide: How to resize, rotate, and crop images
- Dreamweaver guide: How to work with external Cascading Style Sheets

- Dreamweaver guide: How to troubleshoot Cascading Style Sheets
- Dreamweaver guide: How to use templates
- Dreamweaver guide: How to check files in and out
- Dreamweaver guide: How to add text from another document
- Dreamweaver guide: How to use CSS text styles
- Dreamweaver guide: The Dreamweaver Insert panel
- Dreamweaver guide: How to create links
- Dreamweaver guide: How to create data tables
- Dreamweaver guide: How to add rich media to a web page
- Electronic file: Sample_Flash-Movie.flu
- Electronic file: Sample_Flash-Movie.swf
- Electronic file: Sample_Flash-Video.flv
- Electronic file: Sample_HTML5-Video.webm
- Electronic file: Sample_HTML5-video.mp4
- Electronic file: Sample_HTML5-video.ogv
- Electronic file: Sample_audio.mp3
- Dreamweaver guide: How to create forms
- Dreamweaver guide: How to create image maps
- Presentation: Web design for multiple screens
- Dreamweaver guide: How to deploy for multiple screens
- Guide: Technical features and their design solutions
- Dreamweaver guide: How to check links
- Dreamweaver guide: How to preview and test a web project
- Worksheet: Design review
- Dreamweaver guide: How to publish web files
- Worksheet: How to promote a website
- Worksheet: Review and redesign

Background preparation resources

- [Technical and content information](#)
- [Key terms](#)
- [ISTE NETS*S Standard for Students](#)
- [Adobe Certified Associate, Web Communication objectives](#)

Project steps

1. Introduce students to the goals of the project. Explain that they will use their design documents, storyboards, and prototypes from Project 2 to build, test, and publish their client websites:
 - Learn web development best practices
 - Implement reusable designs
 - Build a website with web standards
 - Design for multiple screens
 - Conduct technical and usability tests
 - Publish, present, and launch a website

Web development best practices

(Suggested time: 150–250 minutes)

2. Explain that students will use Adobe Dreamweaver for the production of the website for which they defined, structured, and created a prototype in Project 2. Remind students to continually refer to and assess their project plans throughout the site build to make sure they are on task, team members are aware of responsibilities, and deliverables are completed by the expected due dates.^{1.6}
3. Ask students to start Dreamweaver. Briefly review the Dreamweaver interface that was introduced in Project 1. Review how to work with panels, how to access preset workspaces, and how to customize the workspace.

Dreamweaver guide: Introduction to the Adobe Dreamweaver CS6 workspace^{3.1, 3.6}

4. Briefly discuss the options for setting up a web project:
 - How to determine the kind of web project to create. Discuss the differences between HTML1.x, HTML4, and HTML5 DOCTYPEs.^{4.1}
 - How to create a new blank page.^{4.3}
 - How to create a new blank template, a page from a template, and a page from a sample.^{5.5}
 - How to use and modify CSS starter layouts.^{5.8}
 - How to use templates.^{5.5}
 - How to use Fluid Grid Layouts for mobile, tablet, and desktop versions of a website layout.
 - How to set up a local root folder and site structure.^{4.2}
 - How to define and modify document properties.^{5.1}

Dreamweaver guide: How to set up a local root folder and site structure^{4.2}

Dreamweaver guide: How to set document properties^{3.3, 5.1}

Dreamweaver guide: How to use CSS starter layouts^{5.8}

Note: More instruction on how to set up and use templates and Fluid Grid Layouts is available later in this project.

5. Ask students to open a new document, save it, define the site structure, and name the file index.html. Discuss the following:^{4.3}
 - Extensions for filenames and what each means (e.g., .html, .htm, .php)
 - Using *index.html* or *default.html* as the default filename for a home page
 - Rules and best practice for naming HTML files (e.g., maximum characters, forbidden characters, lowercase letters, underscores and spaces)
6. Use a class brainstorm to introduce file management, generating reasons for file organization.^{4.3} Questions to ask the class include:
 - What is file management?
 - Why is it important?
 - How might you organize a set of files?

7. Ask students to propose ways they might organize their files in Dreamweaver and conventions that might help manage their files and folders, such as using names that help others recognize the contents of a folder or file. Decide as a class what conventions you will use.

Guide: File-naming conventions ^{4.3}

Worksheet: File management

8. Briefly introduce students to the Files panel in Dreamweaver. Ask students to open Dreamweaver and look at the Files panel. Discuss and define the following terms, relating them to interface elements in the Files panel:

- Dreamweaver site map
- Get files
- Put files
- FTP
- Local host
- Testing and remote servers
- Publish

To round out students' knowledge of the Files panel, show them how to use the panel to maintain site files and links.

Dreamweaver guide: How to manage files and links by using the Files panel ^{3.5, 6.4}

9. Briefly review the principles of using HTML to code a web page and CSS to specify the layout and styles for a website. In this discussion, emphasize and demonstrate the following:
 - HTML and CSS as web standards and the current iterations of HTML5 and CSS3
 - Specific HTML tags such as HEAD, TITLE, BODY, H1-H6, P, EM, STRONG, A HREF, TABLE, TR, TD, OL, LI, UL
 - Using HTML5 best practices such as DOCTYPE so a browser knows what version of HTML the page is written in and can render it correctly, meta charset so a browser knows the character-set of the website, and semantic markup tags to enhance the website's structure
 - New HTML5 elements and their capabilities, such as section, article, aside, hgroup, header, footer, nav, figure, video, audio, embed, wbr, and canvas

Guide: Introduction to HTML and CSS ^{4.1, 5.2, 5.6}

10. Explain how to control the design and layout of a page by placing div tags and then adding content to them. The rectangles students drew for images, text blocks, and navigation on their wireframes represent div tags in their Dreamweaver documents. Explain the difference between absolute, relative, fixed, and static positioning of div tags. Explain how they can use CSS3 to create web-page layout features such as tabs, animated drop-down menus, and accordions. Using the "I do, we do, you do" method, demonstrate how to place div tags corresponding to the blocks of information students sketched in their wireframes. ^{2.5}

Dreamweaver guide: How to lay out a web page with CSS ^{5.2}

Note: You might discuss the use of tables in the past for web-page layout. Explain to students that using div tags instead of tables creates a cleaner implementation of design and that CSS-based layouts are now the standard layout method in professional web design.

11. From the specifications in their design documents, ask students to build a home page using CSS. Explain that students should create the CSS layout and add the CSS rules and styles for their website as outlined in their design document. To help them, demonstrate the following technical features:
- How to use the CSS Styles panel to create and edit CSS rules ^{5.8}
 - How to use the Property panel to apply a style to a paragraph ^{3.3, 5.3}
 - How to use the Property panel to set off a block quotation by indenting it ^{3.3, 5.3}
 - How to use the Page properties dialog box to set document properties ^{5.1}

Dreamweaver guide: How to create and edit a CSS rule ^{5.8}

Note: Depending on the level of your students, you may want to discuss vendor prefixes. Until web standards are accepted across all modern web browsers, vendor prefixes are often needed within CSS style sheets.

Implementing a reusable design

(Suggested time: 250–350 minutes)

12. Introduce and discuss the benefits of reusable design. Topics to cover include:
- How web standards such as CSS help make designs reusable and enable quick global changes
 - How reusable designs can make for less work on future projects
 - How such things as code snippets, templates, CSS Starter Layouts, and other elements can be reused
 - How reusable design is especially helpful when designing for multiple devices
 - How reusable design can help keep a website current even in older browsers

Presentation: Reusable design ^{5.8}

13. Discuss the importance of templates, assets, and libraries as tools for promoting consistency in website design. Other techniques with which students should be familiar are managing reusable assets in the Assets panel, using library objects, applying styles to text, using standard fonts, and using web-safe colors. ^{2.1}

Note: You may want to discuss includes at this point. Explain that includes are sections of HTML that include content of one or more files from another web page. They are especially helpful for items such as logos, navigation, and copyright information that are repeated on most, or all, of your web pages. You can program includes to be client-side (cached on the client and viewable locally without a web server) or server-side (files on the web server).

14. Discuss the types of elements that can be stored in the Assets panel. Briefly demonstrate how to add an element to the Assets panel and how to place an element in a document by dragging or by using the Insert button.

Dreamweaver guide: How to use the Assets panel ^{3.4}

15. Using their wireframes, storyboards, and prototypes, ask each team to decide what will be on all pages of the site. ^{2.5} Explain how using style sheets and templates for content pages helps keep the site consistent while the team produces the site. ^{2.1}

16. Explain that Adobe Fireworks enables you to design the look and functionality of interactive graphics (such as buttons, links, and rollovers) and save them as HTML so you can import them into a Dreamweaver document. Explain that virtual buttons can have four states: Up, Over, Down, and Over While Down. The first three states are the most common. Demonstrate the following:
- How to create a three-state button in Fireworks and how to set the active area for a button
 - How to create a navigation bar in Fireworks
 - How to create rollover images in Fireworks
 - How to export to Dreamweaver

Fireworks and Dreamweaver guide: How to create interactive navigation ^{4.9}

17. Explain that students will now begin to build out the assets and graphics for their client sites. Ask students to use Photoshop, Illustrator, and/or Fireworks to create the common graphic elements from their final comps, such as creating a navigation bar in Fireworks. Remind them that the look of their graphics should coordinate with their site designs.
18. Explain that students should export graphics in a web-optimized file type and size. Introduce the file types available for graphics: GIF, JPEG, and PNG. Discuss how the file types differ and when it is appropriate to use each type of file. You might also distinguish between a Fireworks native PNG and a flattened PNG image. ^{4.5}
19. Using large GIF and JPEG sample images, introduce the Optimize panel in Fireworks or the steps for resizing images in Photoshop and demonstrate to students the methods available for optimizing an image. Ask each team to optimize a JPEG or GIF. All JPEG groups should have the same image, and all GIF groups should have the same image. Have students save the images in two ways: graphic file as small as possible and graphic file as small as possible while keeping as much image quality as possible.

Fireworks guide: How to optimize images

Photoshop guide: How to resize, rotate, and crop images

20. Ask some teams to share their optimized images and discuss the benefits of optimization and the balance necessary to preserve quality.
21. Ask students to import each asset and graphic element, such as logos, buttons, and mottos, from Photoshop, Illustrator, or Fireworks into the Dreamweaver Assets panel, including the library, for later use throughout the site. Explain they will use these graphics on their home and content page templates and encourage them to create as many reusable assets and graphics as possible and use them across their content pages.
22. Demonstrate how to edit and scale images in Dreamweaver and how to edit Photoshop (using Photoshop Smart Objects) and Fireworks assets from within Dreamweaver. ^{4.5, 5.4}

Building a website

(Suggested time: 300–400 minutes)

23. Once students have built their assets and graphic elements, remind students of their previous experience using CSS. Demonstrate how to use internal and embedded style sheets. Demonstrate how to turn internal styles into a new external style sheet. Ask students to start creating external style sheets for their client sites as they follow along with the demonstration. After the demonstration, have the students continue creating the style sheets for their client sites, applying what they have learned about creating and editing styles to implement the styles required by their design comps for the home page they created.

Dreamweaver guide: How to work with external Cascading Style Sheets ^{5.8}

Dreamweaver guide: How to troubleshoot Cascading Style Sheets ^{5.8}

Note: If you want students to learn specific technical techniques to prepare for the Adobe Certified Associate, Web Communication exam, introduce the Code Navigator, CSS Inspect, and Live view as useful tools for identifying the CSS rules applied to a web page; demonstrate how to use selector types such as descendent selectors, classes, the tag selector, pseudo class selectors, and group selectors; and show students how they can troubleshoot CSS issues by using tools such as CSS layout backgrounds.

24. Ask student teams to use Dreamweaver to create a template for their content pages, following their storyboards. Their templates should incorporate the style sheets they created. Students may need to supplement their style sheets with new styles or create new style sheets for their content pages. Encourage them to use div tags, paragraph styles for headings, and indentation to create and maintain a page structure and a meaningful content hierarchy. Students should insert the graphics they created in Photoshop, Illustrator, and/or Fireworks. Make sure they add editable regions and save their files as templates.

Dreamweaver guide: How to use templates ^{5.5}

Note: If students are preparing for the Adobe Certified Associate, Web Communication exam, introduce the benefits of creating templates with sections for page content that can be edited through a web content management system such as In Context Editing. ^{4.5} This allows clients to make simple text changes to the web pages themselves.

25. Ask students to use their templates to create the rest of the pages for their client websites, inserting elements from the Assets panel as appropriate. Explain that by using a template, all team members can work on the pages while maintaining uniformity of site design.
26. Now that students have a working knowledge of Dreamweaver, explain that you will go over some best practices for helping teams have efficient collaboration and organization. Introduce and demonstrate the following:
- Explain that Subversion integration allows multiple team members to combine their individual edits into one comprehensive Dreamweaver document that has reconciled the edits made to the document.
 - Discuss how these procedures support collaboration within a team and eliminate the possibility that two team members may simultaneously make different changes to a file.
 - Discuss how subversion integration contributes to the best practices for working with a team.

- Demonstrate the procedure for checking Dreamweaver files in and out; moving, copying, and deleting shared files; synchronizing changes with a remote Subversion SVN repository; and hiding files with File Cloaking.

Dreamweaver guide: How to check files in and out ^{6.4}

Note: Knowledge of Subversion integration and check in/check out functions are part of the Adobe Certified Associate, Web Communication exam. If you are unable to install Subversion, introduce and discuss the usefulness of these features to prepare your students for the exam.

27. Students can now begin adding text, images, links, rich media, and other content to their content pages. Review how to add text to web pages, including adding lists. ^{4.4, 5.3} Demonstrate how to use CSS text styles to define text styles across their websites. Demonstrate how to access and use web-hosted fonts.

Dreamweaver guide: How to add text from another document ^{4.12}

Dreamweaver guide: How to create lists ^{5.3}

Dreamweaver guide: How to use CSS text styles ^{5.3, 5.8}

Note: If students aren't sure how to copy and paste text from a word-processing document and how to import text from a word-processing document, show them.

28. Review how to add graphics and links to content pages by using the Insert panel. Ask students to insert images, text, and other assets they may have, including those from the Assets panel and library.

Dreamweaver guide: The Dreamweaver Insert panel ^{3.2, 4.5}

29. Discuss the differences among the types of links: absolute, site-root-relative, and document-relative. Discuss link targets: `_blank`, `_self`, `_parent`, and `_top`. Discuss named anchors and ID attributes. Ask students to link some text on their home page to text on their content pages.

Dreamweaver guide: How to create links ^{4.6}

30. Demonstrate how to add, adjust, and modify data tables to content pages.

Dreamweaver guide: How to create data tables ^{4.11}

Note: If students are preparing for the Adobe Certified Associate, Web Communication exam, be sure to cover creating data tables and importing tabular data from a Microsoft Excel spreadsheet. ^{4.12}

- 31.** Discuss the best practices for adding video, audio, and other rich media elements to a web page. Discuss the difference between HTML5 and Flash video, especially for older browsers that don't support HTML5 and mobile devices that may not support Flash video. Demonstrate how to insert Flash and HTML5 video into a web page. Demonstrate how to add audio to a web page. Demonstrate how to add Flash SWF files to a web page and edit them from within Dreamweaver.

Dreamweaver guide: How to add rich media to a web page ^{4.7, 4.8}

Electronic file: Sample_Flash-Video.flv

Electronic file: Sample_HTML5_video.ogv

Electronic file: Sample_HTML5_video.webm

Electronic file: Sample_HTML5_video.mp4

Electronic file: Sample_audio.mp3

Electronic file: Sample_Flash-Movie.swf

- 32.** Students might create one or more of the technical features described in the following guides, based on the needs of their clients. You might want to demonstrate any of the technical features teams plan to implement.

Dreamweaver guide: How to create forms ^{4.13}

Dreamweaver guide: How to create image maps ^{4.10}

Note: If students are preparing for the Adobe Certified Associate, Web Communication exam, demonstrate how to build image maps. Alternatively, you can introduce your students to CSS image sprites. CSS image sprites enable multiple images to be turned into a single CSS background that is capable of revealing the correct parts of the multifaceted image.

Note: If you want students to learn specific technical techniques to prepare for the Adobe Certified Associate, Web Communication exam, you might have them build test pages or sample features to show their clients how the technical features would be implemented or as a learning exercise for students to complete as they wait for feedback from clients.

Designing for multiple screens

(Suggested time: 50–150 minutes)

- 33.** Review the concept from Project 2 of designing and developing for multiple screens in Dreamweaver. Discuss various devices for viewing web content and making design decisions based on the screens.

Presentation: Web design for multiple screens ^{1.5, 2.1, 2.2}

- 34.** Discuss the methods for dynamically delivering appropriate content for viewers' screens. Students can use Fluid Grid Layouts to create a layout that responsively adjusts to a viewer's screen size. Or students can use CSS3 media queries to define tablet, phone, and desktop form factors and specify custom CSS files for each device form factor.

- 35.** Ask your students to use either Fluid Grid Layouts or CSS Media Queries to prepare their client sites for at least one screen other than a desktop.

Dreamweaver guide: How to deploy for multiple screens ^{1.5, 2.1, 2.2}

Testing a website

(Suggested time: 150–300 minutes)

36. Introduce the topic of quality assurance by asking students about their perception of quality assurance.

- What is quality assurance?
- What steps would you take to guarantee quality?

37. Ask students to begin the process of technical testing by making a checklist—also called a test plan—of the technical features they need to test as those features are specified in their design documents and storyboards. The checklists might include the following:^{6.1}

- Test links to ensure they go to proper destinations and check for broken links by using the Link Checker panel in Dreamweaver.
- Check for layout elements that should appear on every page (for example, navigation bars and banners across the top).
- Check the spelling of all text (such as content paragraphs and button labels) on every page.
- Use Adobe BrowserLab to check that the site works in at least the three major browsers—Internet Explorer, Mozilla Firefox, and Safari—and on both Windows and Macintosh computers.
- Check that the site works on a device other than a laptop.
- Check that all images have alternative text and meet other accessibility requirements.^{1.4}

Guide: Technical features and their design solutions

Dreamweaver guide: How to check links^{6.1}

Dreamweaver guide: How to preview and test a web project^{2.2, 6.1}

38. You might want to pair teams to test each other's sites or have members of each team follow the test checklist on their own site. Make sure testers write down the bugs and problems they find so the team knows everything they need to fix.

39. Introduce the topic of usability testing by asking students about their perception of usability testing.^{6.2}

- What is usability testing?
- How would you conduct a usability test?
- Why is it important to question site visitors?
- Why is it important to watch visitors as they go through your site?
- What types of things should you look for when you watch site visitors?
- When watching site visitors, how will you know they are having trouble?
- When watching site visitors, how will you know when they like what they see on your site?

Note: If students are preparing for the Adobe Certified Associate, Web Communication exam, discuss various methods for collecting ongoing user feedback and site-evaluation information, such as forms, site reports, e-mail requesting feedback, surveys, and so on.^{6.3}

40. Usability observation looks for latent behavior, behavior of which the site visitor is unaware. Latent behavior is found in body language or facial expressions visitors are unable to verbalize. Some examples follow:

- Site visitors might wave the pointer around the screen for a short while, looking for something, but might not document that the item was difficult to find.

- Site visitors might sigh or frown because they are tired of waiting for an image to load after clicking a link. This behavior indicates that an image file is too large, but the visitor might not document this occurrence.
41. To prepare for usability testing, ask student teams to make a usability-test interview sheet for testers to fill out as they go through the site. You might want to send your students to the All Things Web site at www.pantos.org/atw/35317.html for interview worksheet ideas.
 42. For usability testing, pair students from different teams. Ask one partner to test the other's site, following the usability-test worksheet. The other partner should observe the tester and document the tester's actions, looking for latent behavior. After the first tester finishes, have partners switch roles and perform the usability test for the other student's site.^{6.2}
 43. From the peer feedback and personal notes, ask students to propose written solutions and implement them. If they do not plan to implement a proposed solution, make sure they document why they choose not to do so.
 44. Ask teams to revise their sites based on the information from the technical and usability tests.
 45. In preparation for publishing and launching their sites to their clients, ask teams to conduct an overall design review. Pair teams and have them explain their client's goals and audience; then have them conduct a design review of the client's website by evaluating the following strengths of the site's design:^{2.4, 6.2}
 - Consistency
 - Ease of use
 - Use of color
 - Universal navigation
 - Layout
 - Links
 - Usability
 - Accessibility

Worksheet: Design review

46. Allow teams time to make any necessary changes based on the design review.

Publishing a website

(Suggested time: 50–150 minutes)

47. Introduce students to the concept of publishing a website. Explain the general idea of taking files from students' local root folders on their computers and putting them in a location on a remote server.
48. Explain to students that although there are various ways to publish, they will be using Dreamweaver to publish their web files. Demonstrate how to set up a remote site. Discuss the contents of the dialog boxes you encounter as you demonstrate the steps for setting up a remote site.

Dreamweaver guide: How to publish web files^{6.5}

Note: You might ask students to generate a list of other techniques they could use to publish files. Point out to your students the differences between web servers and FTP servers as they relate to publishing a Dreamweaver site. Discuss the common practice in web design and development firms of publishing to a testing server or internal development server prior to live publication.

49. If students have access to a remote server, allow time for them to publish their sites.^{6.5}

Note: If you can't allow students to publish their own files, you might spend more time discussing the steps you demonstrate and engaging in extension activities to help students better understand the publishing process.

50. Ask students to use a web browser to open their newly published sites, navigate to each page, and make sure all files were successfully published and function as they should.

Presenting and launching a website

(Suggested time: 50–150 minutes)

51. Explain that when a client is working with a large web development firm, the client may expect that different teams within the firm will execute various aspects of the web production process. Although most web designers would not be responsible for launching and marketing a site, this process is important to understand.
52. Introduce Search Engine Optimization (SEO) and describe the primary components of a website that search engines crawl to find relevant search results. Encourage each team to identify the key terms that might be useful to include in the meta tags of their sites to make their pages easier for search engines to find. Demonstrate how to add head content to a web page and have students insert head content on their web pages to optimize discovery by search engines.^{5.7}
53. Ask students to investigate how to use web search engines to promote or market a website. They can find information on search engines on the Submission Tips page at <http://searchenginewatch.com/webmasters/>. Students should explore and contrast the following options:
- Find a search engine that offers free submission of websites. Make a list of their submission procedures.
 - Find a search engine or promotion site for which you pay to submit your site. State the cost and discuss whether the benefits of the service are worth the cost.

Worksheet: How to promote a website

54. Ask students to create formal presentations and then present the sites to their clients. Make sure at least one team member takes notes on the client's comments.^{2.6}
55. Allow students time to redesign their sites based on client feedback. Ask students to finalize and publish their websites.
-

Extension activities

You can extend the project in the following ways:

- *Multi-screen:* Have students take the websites they built and adapt them for a third device and/or screen size.
- *HTML5:* Students can extend their website content to take advantage of the new elements in HTML5, such as section, article, aside, header, footer, and canvas. For more information about these new elements, see <http://dev.w3.org/html5/markup/elements.html#elements>
- *Advanced topics in CSS:* If students seem to be more advanced, you might have them research the Dreamweaver help system and articles from the CSS Developer Center (www.adobe.com/devnet/dreamweaver/css.html) to understand how they can use CSS for positioning.
- *Peer teaching:* If some teams use a guide to learn about a technical feature, have them hold a mini-seminar to teach the rest of the class. Their presentation could include the following:
 - A working example
 - Short explanations of when and why to use the feature (design challenges it helps to solve)
 - Step-by-step demonstration of how to create the feature
- *Using local experts:* Assign students as resource specialists. For example, have four students act as local experts to help others use a software product. With these extra hands and heads, you will be able to manage the classroom without being the sole software expert. As students become more familiar with the software, more students will be able to serve as local experts.
- *Distributed skill-building:* Another way to teach Dreamweaver skills is to distribute skill expertise among the students. Demonstrate and briefly introduce skills such as creating data tables, inserting images, using text and text properties, and inserting outside content by using copy-and-paste or file import. Following this brief introduction, divide the students into small groups and ask each group to become experts for one of the skills, applying it to a content page they create. After they have mastered their specific skill, ask each group to present the application of the skill to the class and explain their methodology. This explanation and presentation will help others understand how to use the technique while they continue to work on their content pages.
- *Sample usability guideline:* Setting visitor performance and preference goals helps developers build better websites. It can also make usability testing more effective. For example, some Internet sites have set a goal that information will be found 80% of the time and in less than one minute.
- *Publishing techniques:* Form small groups of students to research techniques on publishing websites. Consider techniques students might encounter if they were to publish from a home computer, such as using FTP clients or wizards from site-hosting services such as Yahoo. After each group has learned about its technique, have the group briefly present it to the class. Their presentation might include the positive and negative points of using the technique, a short demonstration or handout explaining how to use the technique, and a description of the technology behind the technique.
- *More tags, prefixes, and includes:* To further challenge students, ask student groups to look up a website and identify the tags, prefixes, and includes and what each does in rendering the web page.

Assessment

- Project rubric

Background preparation resources

- Review the student step-by-step guides referenced in this project.
- To view video tutorials aligned with the skills required to complete this project, visit the Digital Design CS6 show on Adobe TV: <http://tv.adobe.com/show/digital-design-cs6/>
- For more teaching and learning resources on web design, planning and other topics in this project, search for resources from the community on the Adobe Education Exchange: <http://edexchange.adobe.com>
- For an overview of the interface and for more information on the technical aspects of Fireworks, see Fireworks Help.
- For an overview of the interface and for more information on the technical aspects of Dreamweaver, see Dreamweaver Help.

Reusable design

- Basic and advanced techniques of Cascading Style Sheets: the CSS Technology Center at www.adobe.com/devnet/dreamweaver/css.html
- CSS Zen Garden. A demonstration of what can be accomplished through CSS-based design: www.csszengarden.com/
- Typekit: <https://typekit.com/>
- Google web fonts: www.google.com/webfonts#HomePlace:home

Building a website

- Using Cascading Style Sheets in the CSS Support Center: www.adobe.com/devnet/dreamweaver/css.html
- Description of meta tags and how to build them when building your pages: <http://searchenginewatch.com/webmasters/article.php/2167931>
- HTML Primer: www.htmlprimer.com
- Writing HTML tutorial: www.mcli.dist.maricopa.edu/tut/

Testing a website

- Adobe's Website Production Management Techniques (read the information under Build and Test on the right side of the page): www.adobe.com/resources/techniques/
- Web usability checklist: <http://drdobbs.com/184412660>
- Adobe Website Production Management Techniques Client Survey: www.adobe.com/resources/techniques/resources/define/client_survey.rtf

Launching a website

- Adobe's Website Production Management Techniques (select Launch on the right side of the page for a checklist of tasks students might want to engage in after launching their sites): www.adobe.com/resources/techniques/
- Google Search Engine Optimization (SEO) Starter Guide: www.google.com/webmasters/docs/search-engine-optimization-starter-guide.pdf

Key terms

- absolute link
- accessibility
- alternative text
- Cascading Style Sheets (CSS)
- check in
- check out
- Class style
- CSS3
- document-relative link
- Dreamweaver library
- Dreamweaver template
- file cloaking
- FTP
- HTML5
- hyperlink
- ID style
- link
- link target
- meta tags
- named anchor
- pathname
- search engines
- server
- site-root-relative link
- subversion integration
- synchronizing
- technical testing

ISTE NETS*S Standard for Students

This project is aligned to the ISTE NETS*S Technology Standards. Depending on the subject and content area the student selects you may research your own state content standards to see how this project aligns to your state requirements.

ISTE NETS*S: Curriculum and Content Area Standards – NETS for Students

1. Creativity and Innovation

Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:

- a. apply existing knowledge to generate new ideas, products, or processes.
- b. create original works as a means of personal or group expression.

2. Communication and Collaboration

Students use digital media and environments to communicate and work collaboratively, including at a distance, to support individual learning and contribute to the learning of others. Students:

- a. interact, collaborate, and publish with peers, experts or others employing a variety of digital environments and media.
- b. communicate information and ideas effectively to multiple audiences using a variety of media and formats
- d. contribute to project teams to produce original works or solve problems

3. Research and Information Retrieval

Students apply digital tools to gather, evaluate, and use information. Students:

- b. locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media.
- c. evaluate and select information sources and digital tools based on the appropriateness to specific tasks

4. Critical Thinking and Problem Solving

Students use critical thinking skills to plan and conduct research, manage projects, solve problems and make informed decisions using appropriate digital tools and resources. Students:

- b. plan and manage activities to develop a solution or complete a project.
- d. use multiple processes and diverse perspectives to explore alternative solutions

5. Digital Citizenship

Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior. Students:

- a. advocate and practice safe, legal, and responsible use of information and technology.
- b. exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity.
- d. exhibit leadership for digital citizenship.

6. Technology Operations and Concepts

Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:

- a. understand and use technology systems.
- b. select and use applications effectively and productively.
- c. troubleshoot systems and applications.
- d. transfer current knowledge to learning of new technologies.

Adobe Certified Associate, Web Communication objectives

1.4 Demonstrate knowledge of website accessibility standards that address the needs of people with visual and motor impairments.

1.5 Make website development decisions based on your analysis and interpretation of design specifications.

1.6 Understand project management tasks and responsibilities.

2.1 Demonstrate general and Dreamweaver-specific knowledge of best practices for designing a website, such as maintaining consistency, separating content from design, using standard fonts, and utilizing visual hierarchy.

2.2 Produce website designs that work equally well on various operating systems, browser versions/configurations, and devices.

2.4 Identify basic principles of website usability, readability, and accessibility.

2.5 Demonstrate knowledge of flowcharts, storyboards, and wireframes to create web pages and a site map (site index) that maintain the planned website hierarchy.

2.6 Communicate with others (such as peers and clients) about design plans.

3.1 Identify elements of the Dreamweaver interface.

3.2 Use the Insert bar.

3.3 Use the Property inspector.

3.4 Use the Assets panel.

- 3.5 Use the Files panel.
- 3.6 Customize the workspace.
- 4.1 Demonstrate knowledge of Hypertext Markup Language.
- 4.2 Define a Dreamweaver site.
- 4.3 Create, title, name, and save a web page.
- 4.4 Add text to a web page.
- 4.5 Insert images and apply alternative text on a web page.
- 4.6 Link web content, using hyperlinks, e-mail links, and named anchors.
- 4.7 Include video and sound in a web page.
- 4.8 Add animation and interactivity to content.
- 4.9 Insert navigation bars, rollover images, and buttons created in a drawing program on a web page.
- 4.10 Build image maps.
- 4.11 Import tabular data to a web page.
- 4.12 Import and display a Microsoft Word or Microsoft Excel document to a web page.
- 4.13 Create forms.
- 5.1 Set and modify document properties.
- 5.2 Organize web page layout with relative and absolutely-positioned div tags and CSS styles.
- 5.3 Modify text and text properties.
- 5.4 Modify images and image properties.
- 5.5 Create web page templates.
- 5.6 Use basic HTML tags to set up an HTML document, format text, add links, create tables, and build ordered and unordered lists.
- 5.7 Add head content to make a web page visible to search engines.
- 5.8 Use CSS to implement a reusable design.
- 6.1 Conduct basic technical tests.
- 6.2 Identify techniques for basic usability tests.
- 6.3 Identify methods for collecting site feedback.
- 6.4 Manage assets, links, and files for a site.
- 6.5 Publish and update site files to a remote server.

Assessment

	0 - Does not meet expectations	3 - Meets expectations	5 - Exceeds expectations
Client website	Absent or incomplete.	Students complete a client site that conveys the site's goals, purpose, and audience. The client site follows the wireframes, design comps, storyboards, and prototypes the client approved. The client site implements reusable design elements such as CSS and templates and adheres to web standards. The client site is published and works across browsers.	Students complete a client site that clearly conveys the site's goals, purpose, and audience. The client site follows the wireframes, design comps, storyboards, and prototypes the client approved. The site clearly employs design principles. The client site successfully implements reusable design elements such as CSS and templates and adheres to web standards. The client site is successfully published and works across browsers, as well as for multiple screens.
Technical test	Absent or incomplete.	Students create test plans to test technical aspects of the client's site. Students test sites in Adobe BrowserLab, document any bugs, and fix the bugs.	Students create detailed test plans to test all technical aspects of the client's site. Students test sites in Adobe BrowserLab, document any bugs, and fix the bugs.
Usability test	Absent or incomplete.	Students create user feedback forms and run usability tests with peers. Students document observations of user subjects and gather reports for user feedback from the test. Design teams implement all feasible suggestions from user test.	Students create detailed user feedback forms and run usability tests with multiple peers. Students clearly document observations of user subjects and gather detailed reports for user feedback from the test. Design teams implement all suggestions from user test and provide clear articulation of the impact these changes had on the overall site.

	0 - Does not meet expectations	3 - Meets expectations	5 - Exceeds expectations
Client presentation	Absent or incomplete.	Students present the client site to the client, highlighting how the client site accomplishes the client goals, meets audience needs, and addresses client content and technical requirements and propose the launch plan.	Students present the client site to the client, highlighting how the client site accomplishes the client goals, meets audience needs, and addresses client content and technical requirements and propose the launch plan. Students evaluate successful strategies employed during the production and offer a plan for how the client can go forward with managing and maintaining their site.
Teamwork	N/A	Student teams track milestones and due dates in project plan. They work with all members of the team to distribute work equally and meet deadlines.	Student teams track milestones and due dates in project plan. They work with all members of the team to distribute work equally and meet deadlines. Student teams use technologies such as check in and check out and develop workflows that help all team members work efficiently.