7 Roadmap

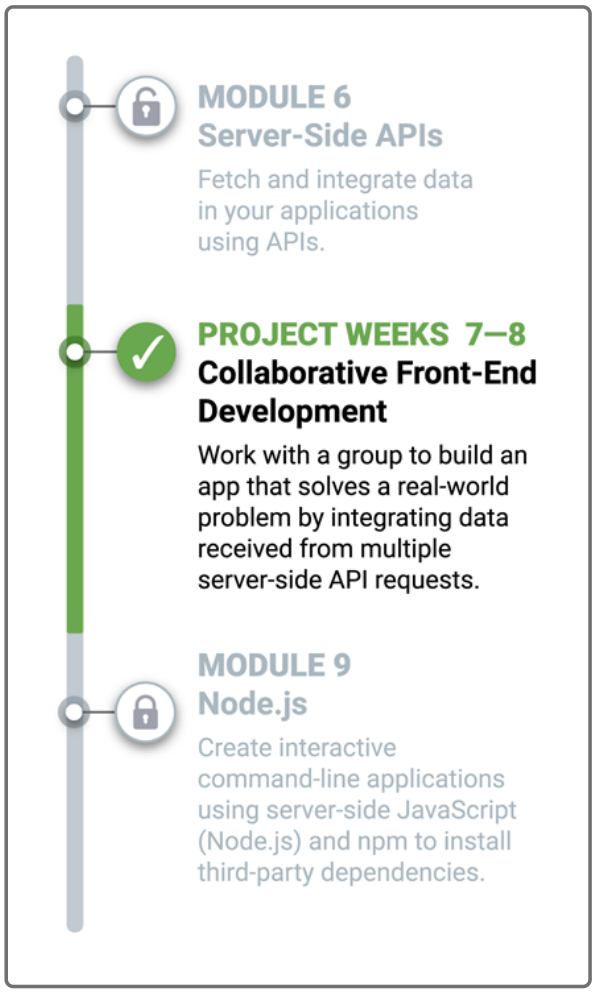
You've reached an important milestone in your journey to become a web developer! As you work on your project, here's how you should navigate the next two weeks.

Virtual Classes (Recommended) and Virtual Classes (Required)

Your virtual classes will look a bit different over the course of the next two weeks as you complete Modules 7 and 8. Although your recommended virtual classes remain recommended, you are **highly encouraged** to attend both of these classes. This week's class is especially important, as you will learn more about agile project management, including the following concepts:

- Iterative development
- Minimum Viable Product (MVP)
- User stories
- Acceptance criteria

- Stand-up meetings
- Kanban boards
- GitHub issues



You'll also have the opportunity to devote most of your time in these classes to working on your projects directly with your group members with the added benefit of having direct access to your instructors and TAs for help.

This is what your schedule will look like for the next two weeks:

Module 7: Virtual Class (Recommended)

You'll start this week by learning the agile project management concepts necessary to working collaboratively on your project. You'll also work with your instructor and fellow group members to finalize your project idea. Even though this class is recommended, you are **highly encouraged to attend**.

Module 7: Virtual Class (Required)

Your group will collaborate closely on your project, also using this time to ask for help from your instructor, TAs, and fellow students. This class time gives you a guaranteed block of time to work on your project with your group.

Module 8: Virtual Class (Recommended)

Your group will continue to collaborate on your project, also using this time to ask for help from your instructor, TAs, and fellow students. Even though this class is recommended, you are **highly encouraged to attend**, as it

gives you a guaranteed block of time to work on your project with your group.

Module 8: Virtual Class (Required)

Presentation day! You and your group members will join your fellow students in presenting your project. Your group will give a 10-minute presentation on your project, with about 7 minutes devoted to the presentation itself, followed by a 3-minute Question & Answer session.

You won't be submitting your code for review until the Sunday following your presentations. This means that your code doesn't have to be 100% ready, but it should be as close as possible. Build your presentation to show a typical user experience that includes all of the working functionality of your application.

Think of your presentation as your chance to display your minimum viable product: the product with just enough features to satisfy early customers and provide feedback for future product development.

Sunday Following Presentation Day

Each member of your group is required to submit your deployed application, your GitHub repository, and your portfolio with your project added to it on the **Sunday following presentation day**.

Finally, remember that you and your group should be working and meeting outside of class too as you move towards finishing your project. It's also important that you find time to rehearse your project presentation prior to presentation day.

Working With Your Group

This might be your first time working on a group project in an online course. It's important that you find time to meet with your group and communicate regularly with your group members. Use the following suggestions to help you make the best use of your time:

- Plan for significant amounts of collaboration time outside of your virtual sessions.

- Decide at the beginning how you're going to communicate with your group members. Create a Slack channel, exchange phone numbers, and ensure that the group knows each group member's working hours (the hours of the day they are available to collaborate).

- Set up an agile project using GitHub Projects so that your group can track interim tasks.

- Create internal milestones to ensure that your group is on pace. Set due dates for these milestones, so that you have a timeline for the completion of the project. Because this is a two-week project, you'll want to make sure that you have a certain portion of the site finished by the end of the first week to stay on track. Some of these milestones might include:

    - Project ideation
    - Project decided
    - CSS color scheme decided
    - Site structure outlined
    - User interface created
    - API integration
    - Testing
    - Creating documentation
    - Presentation created

- As you work on your project, you will divide the work among your group members, but it's important to collaborate and communicate while working on different parts of the application. Don't silo yourself

and then return hoping that everything will work together. Instead, check in regularly and offer support. Make sure you are in sync, and help each other get unstuck when a blocker arises.

Support Available

You will have all of the same support available to you over the next two weeks that you've had up until this point. Your instructor and TAs will provide help during the Virtual Sessions, and your TAs will be there for you during office hours.

You also have access to Learning Assistants on Slack to help unblock you on issues you can't resolve, as well as Tutoring to help you catch up on any topics you might need help with. Be sure to take advantage of these resources, as well as your community of fellow learners and developers—especially your group members—as you complete your first group project.

Where to Find Help in the Curriculum

As you work on your project, you may find yourself trying to remember where you learned some of the skills required to meet the requirements. Use the following tips to help you unblock any challenges you might encounter:

Deploying to GitHub Pages

Review Module 1: HTML, CSS, and Git, Lesson 1: Set Up the Project (especially 1.1.8 Publish with GitHub) to get a refresher on deployment.

Quality Code

For best practices on quality code conventions, review Module 1: HTML, CSS, and Git, Lesson 1: Set Up the Project and Lesson 2: Build the Header and Footer.

Quality Repos and READMEs

For best practices on creating a clean repo with a high-quality README, see Module 2: Advanced CSS, Lesson 1: Set Up the Git Workflow (especially 2.1.5: Create a README File).

Responsive Design

Look in Module 2: Advanced CSS, Lesson 2: Flexing Our Muscles to review flexbox, Lesson 3: Going Mobile to review media queries, and Lesson 4: CSS Grid to review CSS Grid.

Client-Side Storage

For a reminder on how to work with client-side storage, review Module 3: JavaScript, Lesson 5: Resolve Bugs and Add Final Enhancements (especially 3.5.8: Save and Load High Score from localStorage) and Module 4: Web APIs, Lesson 5: Task Persistence.

Interactivity and User Input

To review interactivity and user input, review Module 3: JavaScript, Lesson 1: JavaScript Basics and Module 4: Web APIs, Lesson 1: The DOM, Lesson 2: Work with Forms, and Lesson 3: Update and Delete Tasks.

UI Enhancements

For a refresher on how to improve the UI of your applications, review Module 2: Advanced CSS, Lesson 5: Add Visual Enhancements and Lesson 6: Add Animation and Deploy, and Module 4: Web APIs, Lesson 4: Add Drag and Drop.

Modals

For a refresher on modals, look into Module 5: Third-Party APIs, Lesson 2 Bootstrapped (especially 5.2.5: Add Bootstrap Components).

CSS Frameworks

For alternative CSS frameworks, take a look at the Module 5: Dessert Menu.

Server-Side APIs

Look in Module 6: Server-Side APIs, Lesson 1: Get Started with Requests (and the rest of the module!) to review how you can integrate server-side APIs into your application.