7 Project Overview

What Is a Project?

At the conclusion of each phase of this course, you'll work with a group of your fellow students to create a **project**. A project is collaborative work among a group of developers to create an application that solves a real-world problem. Projects model the experience you'll encounter in every development role at any company, from large multinational businesses to small startups. Coding is collaborative.

A project is a bit different than the Challenge assignments you've worked on for the last six weeks. One of the biggest differences is that you'll no longer build an application by yourself! This has some advantages—you won't have to do all of the work, you can divide up duties, and you can share skills and knowledge with other developers and lean on their strengths. This can also be challenging if you're used to working alone. Constant communication and time management are just two of the skills you'll need to practice to make sure everyone in your group works together to complete the project. We'll provide some tips on how to achieve this in this module's Roadmap.

Projects won't provide you with a user story or acceptance criteria, because you and your group will create them once you decide which real-world problem your application will solve. This lack of constraints can be freeing in a way, because you have room to build what you want, but it also means that you have to decide what those constraints are before you can start working.

Finally, you haven’t had to present your Challenge assignments. A project requires a presentation, because you’re trying to convince an audience that it serves a purpose. Your instructional staff and fellow students are investors, and you’re pitching your creation to them—an experience that developers are required to do frequently. Your presentation is just as important as the actual project, so take it just as seriously.

Why Projects Are Important

Projects are extremely important on your journey to becoming a full-stack web developer. As you get further along in your boot camp, you’ll find opportunities to apply for web developer roles. One of the first requirements you’ll need to even get an interview are high-quality deployed examples of your work featured in your portfolio.

This project will be a centerpiece for your portfolio and your first opportunity to show employers your collaborative skills and coding abilities. Employers want to see what you can do, but they also want to see how you work with other developers. The more examples of deployed collaborative work you have in your portfolio, the more likely you are to get an interview and a job. One of the requirements for this project is that you add it to your portfolio.

Near the end of the boot camp, you’ll have the opportunity to present your work to industry professionals at a Demo Day. As you work with your group, think about the types of questions you may get from potential employers during a presentation or an interview and practice articulating your answers. Why did you choose a certain technology? What was your collaborative process? How did you decide who would work on what? How did you communicate with your group members? How did you contribute specifically to the project? Pay special attention to this last one, as it’s a very common interview question.

Project Examples

It’s completely up to you and your group members to decide what you will build for your project. This is your chance to be creative, so collaborate closely with your group members and your instructional staff to decide what you build. They’ll help you determine what an appropriate project would look like, but just to give you an idea of what other students have built in the past, here are a few examples:

- Tourism app: allow users to plan a trip by using APIs to combine weather data for cities and airline booking data with upcoming events in that area.

- Recipe app: allow users to search for custom recipes using APIs to comb through a library of recipes using specific food search terms, then find the ingredients at the nearest grocery store.

Project Requirements

You and your group will use everything you’ve learned over the past six weeks to create a real-world front-end application that you’ll be able to showcase to potential employers. The user story and acceptance criteria will depend on the project that you create, but there are some requirements that your project must fulfill.

The requirements for this project ask you to use the skills and technologies you’ve learned over the past six weeks (deployment, interactivity, client-side storage, responsive design, polished UI).

They also ask you to use some of the less obvious skills you’ve practiced in the lessons and in your class. You’ve learned that there are many frameworks used in web development. You’ve already experienced what it’s like digging through Bootstrap’s documentation—now use the skills you’ve developed to dive into the documentation for an alternative framework.

The lessons and Challenge assignments have reinforced the importance of quality coding standards and best practices for repositories and READMEs. Use these skills to create a professional and impressive application that you could present with pride to any potential employer.

Finally, these requirements ask you to place your work in the context of your future career by adding your project to your portfolio. Each module's Career Connection has prepared you for life after this boot camp. Getting your first developer role requires you to have applications in your portfolio so that employers can see your work.

Your project should fulfill the following requirements:

- Use a CSS framework other than Bootstrap.
- Be deployed to GitHub Pages.
- Be interactive (i.e., accept and respond to user input).
- Use at least two server-side APIs.
- Does not use alerts, confirms, or prompts (use modals).
- Use client-side storage to store persistent data.
- Be responsive.
- Have a polished UI.
- Have a clean repository that meets quality coding standards (file structure, naming conventions, follows best practices for class/id naming conventions, indentation, quality comments, etc.).
- Have a quality README (with unique name, description, technologies used, screenshot, and link to deployed application).

Finally, you must add your project to the portfolio that you created in Module 2.

Presentation Requirements

You and 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. Use this **Project Presentation Template (https://docs.google.com/presentation/d/1_u8TKy5zW5UlrVQVnyDEZ0unGI2tjQPDEpA0FNuBKAw/edit)** to address the following:

- Elevator pitch: A one-minute description of your application.
- Concept: What is your user story? What was your motivation for development?
- Process: What were the technologies used? How were tasks and roles broken down and assigned? What challenges did you encounter? What were your successes?
- Demo: Show your stuff!
- Directions for future development.
- Links to the deployed application and the GitHub repository.

Grading Requirements

This project is graded based on the following criteria:

Technical Acceptance Criteria: 25%

- Satisfies the following code requirements:
    - Application uses at least two server-side APIs.
    - Application uses client-side storage to store persistent data.

  - Application doesn't use JS alerts, prompts, or confirms (uses modals instead).
  - Application uses a CSS framework other than Bootstrap.
  - Application is interactive (accepts and responds to user input)

Concept 10%

- Application should be a unique and novel idea.
- Your group should clearly and concisely articulate your project idea.

Deployment: 20%

- Application deployed at live URL and loads with no errors.
- Application GitHub URL submitted.
- Portfolio at live URL submitted, featuring project.

Repository Quality: 10%

- Repository has a unique name.
- Repository follows best practices for file structure and naming conventions.
- Repository follows best practices for class/id naming conventions, indentation, quality comments, etc.
- Repository contains multiple descriptive commit messages.
- Repository contains a quality README file with description, screenshot, and link to deployed application.

Application Quality: 15%

- Application user experience is intuitive and easy to navigate.

- Application user interface style is clean and polished.
- Application is responsive.

Presentation 10%

- Your group should present using Powerpoint or a similar presentation software.
- Every group member should speak during the presentation.
- Your presentation should follow the **Project Presentation Template (https://docs.google.com/presentation/d/1_u8TKy5zW5UlrVQVnyDEZ0unGI2tjQPDEpA0FNuBKAw/edit)** .

Collaboration 10%

- There are no major disparities in the number of GitHub contributions between group members.

Due Date

You and your group members will work on this project over the course of the next two weeks.

Project presentations will take place during the Virtual Class (Required) on the week of the next module, Module 8.

The project itself is due on the **Sunday following group presentations**. **Each member of your group** is required to submit the following for review:

- The URL of the deployed application.
- The URL of the GitHub repository, with a unique name and a README describing the project.
- The URL of your portfolio, with your project added to it.

You’ll be able to submit your project assignment once you reach Module 8 next week.