## **Project 1: Interactive Front-End Application**

At the conclusion of each phase of this course, you’ll work with a group of your fellow students to create a **project**—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 from the Challenge assignments you’ve worked on so far. 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 rely on their strengths. Yet 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.

You won't receive a user story or acceptance criteria for your projects, 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, 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.

### **Project Requirements**

With your team, you'll conceive and execute a design that solves a real-world problem by integrating data received from multiple server-side API requests. You'll also learn about agile development methodologies to help you work collaboratively. You'll implement feature and bug fixes using Git branch workflow and pull requests.

You'll write your own user stories and acceptance criteria in GitHub Issues to help your team stay on track with the project. Using GitHub Project to track the status of your project tasks will help you understand the benefits of Kanban boards.

You and your group will use everything you’ve learned over the past six modules 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 your project must fulfill the following requirements:

* Use a CSS framework other than Bootstrap.
* Be deployed to GitHub Pages.
* Be interactive (in other words, accept and respond to user input).
* Use at least two [server-side APIs](https://coding-boot-camp.github.io/full-stack/apis/api-resources)
* [Links to an external site.](https://coding-boot-camp.github.io/full-stack/apis/api-resources)
* .
* Use modals instead of alerts, confirms, or prompts.
* 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, best practices for class/id naming conventions, indentation, quality comments, and so on).
* Have a quality README (including a unique name, description, technologies used, screenshot, and link to the deployed application).

### **Presentation Requirements**

Use this [Project Presentation Template](https://docs.google.com/presentation/d/10QaO9KH8HtUXj__81ve0SZcpO5DbMbqqQr4iPpbwKks/edit?usp=sharing)

[Links to an external site.](https://docs.google.com/presentation/d/10QaO9KH8HtUXj__81ve0SZcpO5DbMbqqQr4iPpbwKks/edit?usp=sharing)

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:

**NOTE**

If a project submission is marked as “0”, it is considered incomplete and will not count towards your graduation requirements. Examples of incomplete submissions include the following:

* A repository that has no code
* A repository that includes a unique name but nothing else
* A repository that includes only a README file but nothing else
* A repository that only includes starter code

#### **Technical Acceptance Criteria: 25%**

* Satisfies the following code requirements:
  + Application uses at least two [server-side APIs](https://coding-boot-camp.github.io/full-stack/apis/api-resources)
  + [Links to an external site.](https://coding-boot-camp.github.io/full-stack/apis/api-resources)
  + .
  + 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.

#### **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, and so on.
* 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/10QaO9KH8HtUXj__81ve0SZcpO5DbMbqqQr4iPpbwKks/edit?usp=sharing)
* [Links to an external site.](https://docs.google.com/presentation/d/10QaO9KH8HtUXj__81ve0SZcpO5DbMbqqQr4iPpbwKks/edit?usp=sharing)
* .

#### **Collaboration 10%**

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

### **How to Submit Your Interactive Front-End Project**

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.