# Project 2: Interactive Full-Stack Project

Projects play a key role in your journey to becoming a full-stack web developer. As you enter the last phase of the boot camp, you’ll begin to apply for development jobs. If you want to land interviews, your portfolio must feature high-quality deployed examples of your work—and you can use your finished projects for that very purpose.

As your first opportunity to show employers your collaborative skills and coding abilities, this particular project will be a focal point of your portfolio. Employers want to know what you can do, but they also want to know 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.

## Project Requirements

With your team, you’ll conceive and execute a design that solves a real-world problem. In creating your first collaborative full-stack application, you’ll combine a robust back end—with servers, databases, advanced APIs, and user authentication—to an intuitive front end. You’ll continue to build on the agile development methodologies you’ve used throughout this course, like storing your project code in GitHub, managing your work with a project management tool, and implementing feature and bug fixes using the Git branch workflow and pull requests.

Instead of using the front end you developed for your first project, you should start from scratch. Doing so will help you reinforce your existing front-end abilities and solidify your understanding of the dynamic between the front end and back end. Also, you’ve likely improved your skills since that first project, so with a stronger foundation now you can build a better application, using all of the additional practice you’ve had in the last several weeks.

You and your group will use everything you’ve learned over the past six weeks to create a full-stack application that works with real-world data to solve a real-world challenge. Even more than your first group project, this project will provide you with an opportunity to demonstrate your problem-solving skills, which employers will want to see when you begin to interview for development roles. Once again, the user story and acceptance criteria will depend on the project that you create, but your project must fulfill the following requirements:

Use Node.js and Express.js to create a RESTful API.

Use Handlebars.js as the template engine.

Use MySQL and the Sequelize ORM for the database.

Have both GET and POST routes for retrieving and adding new data.

Use at least one new library, package, or technology that we haven’t discussed.

Have a folder structure that meets the MVC paradigm.

Include authentication (express-session and cookies).

Protect API keys and sensitive information with environment variables.

Be deployed using Heroku (with data).

Have a polished UI.

Be responsive.

Be interactive (in other words, accept and respond to user input).

Meet good-quality coding standards (file structure, naming conventions, follows best practices for class/id naming conventions, indentation, quality comments, and so on).

Have a professional README (with unique name, description, technologies used, screenshot, and link to deployed application).

## Presentation Requirements

Use this Project Presentation TemplateLinks to an external site. 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. Use the Guide to Deploy with Heroku and MySQLLinks to an external site. on The Full-Stack Blog if you need a reminder on how to deploy to Heroku.

## 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 a Node.js and Express.js back end and uses both GET and POST routes for retrieving and adding new data.

Application has a folder structure that meets the MVC paradigm and uses Handlebars.js as the template engine.

Application is backed by a MySQL database with a Sequelize ORM and protects API keys and sensitive information with environment variables.

Application includes user authentication (express-session and cookies).

Application uses at least one new library, package, or technology not covered in class.

### 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 on Heroku 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, 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 TemplateLinks to an external site..

### Collaboration 10%

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

## How to Submit Your Interactive Full-Stack 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.

## Notes:

Lesson 1 Submitted a project proposal (ideas, outlines, APIs) for feedback.

Created a project repo that is set up with linting and add group members as collaborators.