⅄ sf-wdi-48 / **project-01**
forked from sf-wdi-46/project-01

No description, website, or topics provided.

⊙ **18** commits	⅃ **1** branch	◇ **0** releases	⧉ **3** contributors

Branch: **master** ▾   New pull request   Create new file   Upload files   Find file   Clone or download ▾

This branch is 4 commits ahead of sf-wdi-46:master.   ⅄ Pull request   ⊡ Compare

👤 **kbbushman** Update README.md		Latest commit 9a09a0c 18 hours ago
📁 readme-assets	added assets	5 months ago
📄 README.md	Update README.md	18 hours ago
📄 example-readme.md	took out racing game reference	3 years ago
📄 feedback.md	project 1 yo	3 years ago
📄 past-project-examples.md	project 1 yo	3 years ago
📄 planning.md	Update planning.md	a day ago

📖 **README.md**

🅖🅐 Project One

DESCRIPTION

It's time to put everything that you've learned in the past month together! For your first project you will build a full stack web application using jQuery and AJAX, Express, Node.js, Mongoose, and MongoDB.

The objective of this project is to:

- Apply the skills you've learned by building a full-stack web application from the ground up.
- Demonstrate mastery of topics covered during this course so far.
- Build a polished, published website you can share in your portfolio.

CORE REQUIREMENTS

Make sure to do all of the following with your app.

- **Express API** Build an Express Application that has both HTML and JSON endpoints.
- **RESTful Routes** Design your CRUD routes using the REST convention.
- **AJAX** Leverage AJAX to fetch JSON data from the backend.
- **jQuery** Use jQuery to add interactivity and render data on the client-side.
- **Templating** Render the JSON data on the frontend using template strings.
- **MongoDB** Persist at least two models to a Mongo Database. Use at least one one-to-many or many-to-many relationship between models. You can choose to reference or embed your data.
- **External API** Use an external API to integrate rich data into your app.
- **Git** 50+ commits. Commit early, commit often. Tell a story with your commits. Each message should give a clear idea what you changed. (And don't expose any secret keys/tokens on GitHub!)
- **Code Style** Write professional-looking code. Follow the Airbnb Javascript Styleguide.
- **Visual Design** Use Flexbox, CSS Grid, Bootstrap, Materialize, Foundation, Skeleton, or another CSS framework to

make your front-end snazzy. First impressions matter!

- **Heroku** Deploy your app to Heroku.
- **Documentation** Write a README.md that would make an employer excited to hire you. Screenshots are encouraged. See the example readme for a suggested structure.

PLANNING DELIVERABLES

See the planning deliverables document for more information on the planning steps you should take.

- A clearly defined **Minimum Viable Product (MVP) Scope**. What can you reasonably accomplish in a week?
- **Wireframes** for *every* page. These don't have to be pretty; just sketch what the page will include.
- **User Stories** (divided into sprints) - we recommend Trello for project tracking.
- **Database Models and ERD** Make plans for each resource. List the attributes you'll include in your schemas and what type of data each attribute will be. Draw an Entity Relationship Diagram to illustrate the relationship(s) between models, and note whether you plan to reference or embed related data.
- A **Feasibility Check** for any bonus feature you'd like to complete.

Once your project has been approved, create a **GitHub Repo** for your project and add your teaching team as collaborators.

Examples of External APIs

- Petfinder API- https://www.petfinder.com/developers/api-docs
- Reddit- https://www.reddit.com/dev/api/
- Omdb- http://www.omdbapi.com/
- Nasa- https://api.nasa.gov/#getting-started
- Zomato- https://developers.zomato.com/api

BONUS IDEAS

If you want to push yourself and learn something new, optionally consider doing some of the following with your app, but *please talk to an instructor* beforehand:

- **Front-End Data Validation** Validate data on the front-end by handling incorrect form inputs during create/update. For example, when a form is submitted, check that a field has some text in it, or that its value is a number.
- **Back-End Data Validation** Validate data on the back-end using mongoose's built-in validations, or make your own custom validation.
- **More Models or Relationships** Add another model to your project or create a new relationship.
- **Model Methods** Level up your models by adding a method to one of your schemas. For example, a person schema with `firstName` and `lastName` can have a `fullName` method (see mongoose docs on instance methods and static methods).
- **Authentication** Enable users to store account information and signup, login, and logout.
- **Sass** Use a CSS pre-compiler to write more imperative CSS.
- **Whatever else inspires you!**

TIMELINE

> DO NOT START CODING UNTIL YOUR PROJECT IS APPROVED.

- **Monday, October 8th by 4:00pm** - Present your app plan to your instructors. Before you begin work on your project, the planning deliverables must be checked off with an instructor!
- **Weekdays, October 9th to 12th** - Standups @ 10:00am
- **Saturday, October 13th @ 3pm** - Feature Freeze! Deploy your code to Heroku, and start polishing existing features. Finalize your README.md and prepare for presentations.
- **Monday, October 15th @ 9:15am** - Project due and presentations!

WHAT WE ARE LOOKING FOR

Code must be...

- Clean
  - no unused or commented-out code
  - proper spacing and indentation
- Modular and well organized
  - use the module pattern with `module.exports`
  - separate large tasks into shorter functions
- Appropriately commented
  - use comments to plan, but remove the unnecessary comments for your professional portfolio
  - prefer a well-named variable or function to a comment

See the feedback document for the criteria on which your project will be evaluated.

ACCESS TO INSTRUCTORS

We will hold 1:1s throughout the week. We will also do mini lessons on certain topics if we notice that several people are running into the same issues.

FINAL DELIVERABLES

- Completion of the **core requirements**
- A link to your website **hosted on Heroku**
- A link to your **source code on GitHub**
- A `README.md` file that serves as your **project documentation** (this is important!)
- A **10 minute presentation**, in the company of friends, illustrating:
  - Triumphs
  - Challenges
  - Words of Wisdom
  - And *3* lines(!) of code you want to share with your classmates, copied into a seperate document.

Presentation Tips

- Please zoom your browser window to 150%.
- Please set your SublimeText font to at least 20.0 pixels.
- And speak up!

▸

Project Teams