**[](https://ga-dash.s3.amazonaws.com/production/assets/logo-9f88ae6c9c3871690e33280fcf557f33.png) Project #4: React + Rails Application**

**Overview**

This fourth project combines all we've learned about Rails and React. You'll be **building a (optional ) React + Rails app,** which means you'll be using React for all of your front-end needs (templates), and Rails as your back-end API. Or if you wish to explore Rails further you can build an expanded API with more Models and implement more complex SQL based relationships

**Dates**

* Length: 1 week
* Start: 7/9/2020
* Present: 14/9/2020

**Technical Requirements**

Your app must:

* **Have at least one database model** (more if they make sense!) – the core functionality of your app should rely on a persistent database storage model
* **Include wireframes** that you designed during the planning process.
* Have **semantically clean HTML and CSS**
* **Be deployed online** and accessible to the public
* (we're going to teach you how to deploy on heroku)

**Necessary Deliverables**

* A **working full-stack or expanded API application, built by you, that has full CRUD functionality**
* Hosted on Heroku or somewhere on the internet
* A **link to your hosted working app** in the URL section of your Github repo
* A **git repository hosted on Github**, with a link to your hosted project, and frequent commits dating back to the **very beginning** of the project. Commit early, commit often.
* **A README.md file** with explanations of the technologies used, what problem your app solves/why you made it, the approach taken, installation instructions/requirements, contributor information, unsolved problems, forthcoming features, etc. The README.md should also include
* (links to) **Wireframes of your app**, hosted somewhere & linked in your readme
* (a link to) **User stories you created**

**Suggested Ways to Get Started**

* **Begin with the end in mind.** Know where you want to go by planning with wireframes & user stories & data, so you don't waste time building things you don't need. Keep it lean and keep it elegant.
* **It can be very helpful to write throwaway code to solve short term problems**
* **Use branches on git**
* **Read the docs for whatever technologies you use.** Most of the time, there is a tutorial that you can follow, but not always, and learning to read documentation is crucial to your success as a developer
* **Commit early, commit often.** Don’t be afraid to break something because you can always go back in time to a previous version.
* **User stories define what a specific type of user wants to accomplish with your application**. It's tempting to just make them *todo lists* for what needs to get done, but if you keep them small & focused on what a user cares about from their perspective, it'll help you know what to build.
* **Write pseudocode before you write actual code.** Thinking through the logic of something before you code it is critical.