### Pre-employment checklist

- [] Add the deployed URL to repo settings

- [] Add a readme with:

- [] An embedded screenshot of the app

- [] Explanations of the \*\*technologies\*\* used

- [] \*\*Installation instructions\*\* for any dependencies

- [] Link to your \*\*user stories\*\* – who are your users, what do they want, and why?

- [] Link to your \*\*wireframes\*\* – sketches of major views / interfaces in your application

- [] Descriptions of any \*\*unsolved problems\*\* or \*\*major hurdles\*\* you had to overcome

User Stories

## Weigh-In

User will be able to create a weigh-in {OBJECT}

User will be able to add a new participant [CREATE a record]

## Participants

User will be able to conduct a weigh-in [adding participants’ weight using WEIGH-IN OBJECT]

User will be able to record the weight of a program participant [CREATE new weight]

User will be able to view a list of all program participants [Participants’ INDEX]

User will be able to add a new participant [CREATE new participant]

User will be able to show the weigh-in history of a program participant [SHOW method]

## Statistics

User will be able to conduct computations on weigh-in data

### Hints and Suggestions

\* \*\*Don’t get too caught up in too many awesome features\*\* – simple is always better. Build something impressive that does one thing well.

\* \*\*Design first.\*\* Planning with user stories & wireframes before writing code means you won't get distracted changing your mind – you'll know what to build, and you can spend your time wisely by just building it.

\* \*\*Don’t hesitate to write throwaway code\*\* to solve short term problems.

\* \*\*Read the docs for whatever technologies / frameworks / API’s you use\*\*.

\* \*\*Write your code DRY\*\* and \*\*build your APIs RESTful\*\*.

\* \*\*Commit early, commit often.\*\* Don’t be afraid to break something because you can always go back in time to a previous version.

\* \*\*Keep user stories small and well-defined\*\*, and remember – user stories focus on what a user needs, not what development tasks need accomplishing.

\* \*\*Write code another developer wouldn't have to ask you about\*\*. Do your naming conventions make sense? Would another developer be able to look at your app and understand what everything is?

\* \*\*Make it all well-formatted.\*\* Are you indenting, consistently? Can we find the start and end of every div, curly brace, etc?

\* \*\*Comment your code.\*\* Will someone understand what is going on in each block or function? Even if it's obvious, explaining the what & why means someone else can pick it up and get it.

\* \*\*Write pseudocode before you write actual code.\*\* Thinking through the logic of something helps.

---

### Potential Project Ideas

##### Q&A App

Think of how helpful sites like Quora & StackOverflow are. Maybe there's some other niche, or some surprising twist you can add to the question-and-answer game.

##### Car Repair Log

Keep up with your vehicle as you commute every day to your final weeks of WDI save things like mileage, last inspect, oil changes, and maybe even receipts & important document you need to have backed up.

##### Email-like Messages

Imagine a realtime messaging client, but with private messages, an inbox, unread messages, and who knows what else. This is your chance to re-invent email as we know it.

Here is the GA gallery of exceptionally awesome WDI projects from all over the world! - https://gallery.generalassemb.ly/WDI

---

### Resources

\* \*\*[HackDesign](https://hackdesign.org/lessons)\*\* \_(beginner's reference for thinking like a designer)\_

\* \*\*[Visual Design Hacking](https://generalassemb.ly/online/videos/visual-design-hacking)\*\* \_(a great tips-and-tricks focused video from Front Row)\_

\* \*\*[Web Design For Non-designers](https://generalassemb.ly/online/videos/web-design-for-non-designers)\*\* \_(another great design-related course for all the nerds out there)\_

---

### Project Feedback + Evaluation

Instructors will not be grading this project. We want you to use this final project as your magnum opus as you apply for jobs.

### Support + Issues

Instructors will meet with students during scheduled one-on-ones. Please \*\*submit technical issues to the `project4` repository as Git issues\*\*.

---

### Presentations

#### 2:00 - 3:50

Presentation Schedule updates to follow!

#### 4:00 - 4:30

Scrum

#### 4:30 - 4:40

Life after WDI (Outcomes)

#### 4:40 - 5:00

Life after WDI

#### 5:00 - 5:15

Surveys n Surprises