### **Build a voting application**

### **Overview**

For this exercise, you will build a simple voting web application. Create a GitHub repo and submit your application as a series of pull requests or a single pull request. Explain your design, technical, and UX decisions in the pull request body or bodies. Assume the code reviewer is technically proficient but has zero context. Explain potential solutions to any requirements / extra credit not met.

In this folder, you’ll find:

* Two wireframes, Vote.png and Results.png
  + These wireframes are for a suggested but optional UI

### **Features**

* Your Rails application will accept an email address and zip code on a sign-in form.
* It will present the current list of candidates and the ability to add a candidate.
* Adding a candidate will automatically cast a vote for that candidate.
* It will show the current results tabulated per candidate on a separate, unauthenticated dashboard that updates once every 10 votes.

### **Requirements**

* A given email address can vote 1 time.
* There are no predetermined candidates, all votes will come in the form of write-ins. A write-in vote creates a new candidate on the form for the following voters to choose.
* There is a maximum of 10 unique candidates.
* Everyone gets exactly 5 minutes to vote after signing in before they need to sign in again.
* We use postgres, you are free to use SQLite as your database. Memcached and Redis can be used.
* The application is accessible to screen readers and keyboard navigation.
* The application can be built and run by issuing a single command. Alternatively, it works by issuing these commands:

bash

brew bundle # optional, if not using homebrew provide exact commands for dependencies

bundle install

bundle exec rails db:migrate

bundle exec rails s

### **Assumptions**

* You don’t need to build/integrate an authentication system. Just have a sign-in form that accepts an email address and pretend like the password supplied would always be correct. It's suggested that you store some sort of identifier in an encrypted cookie for simplicity/privacy/tamper resistance.
* Assume that votes are done in good faith, and Votey McVoterson will not be entered as a candidate.
* Assume that only one person is voting at any given time on the only voting machine shared by all in town not connected to the internet.
* Voting is mandatory for everyone.

### **Extra credit**

* Account for things like omitted middle initials/names, typos, or any other reason why a good faith but inaccurate write-in entry could lead to problems.
* Use react-query on the front end.
* Use GraphQL on the backend.
* Have the dashboard update “live” as new values are counted - i.e. without triggering a periodic refresh of the page.
* Add specs. We use RSpec.

### **Rough time estimate for this exercise**

Doing a technical exercise on top of having all the other stuff you have to do during a week can be a challenge, so first off, thank you for taking the time to give this a shot.

In general, we recommend spending about 2 hours on this exercise, the first chunk of which is likely just reading through this document and choosing an approach. If

you can't find that amount of time because of work or life or anything else, get through as much as you are able to (and please feel free to reach out to get additional time!). We'll take a look at your submission no matter how far long you've made it in the process.

If you complete the core feature and requirements criteria and want to get creative with the exercise, then you might consider 1 or 2 of the extra credit options or add any additional feature you think would be useful.

### **Questions**

If you have questions about this assignment, from the specific to the broad, please don't hesitate to reach out to the recruiter you’ve been working with or to the hiring manager. Good luck!

### **Submission**

To submit your solution, please upload a text file or README to Greenhouse that includes the following:

1. A link to your code that we can access, such as a Google Drive folder or a GitHub repo
2. A description of how to run your code
3. Anything else we should know

If you're having any trouble at all with Greenhouse, feel free to email us the file instead.