You will create a screen recording video of yourself completing the challenge, then send me a link to the file via Google Drive. A few things to consider:

- We ask that you complete this challenge within the timeframe agreed on in our conversation.
- You can use screen recording software like Loom, QuickTime, or something similar, to create the video.
- The recording should be of the entire coding challenge, from the beginning to end, which is about 1 hour and 15 minutes.
- Please upload the video file to Google Drive and share an open link with us (we support .wmv, .avi, .webm, .mp4, .mov files smaller/with less than 4gb).
- As you complete the challenge, please explain what you are doing. Walk us through your thinking, explain your decisions, etc.
- Here is a short clip from a recent coding challenge, as an example of what your recording should look like: Example video

----

RoR Challenge

Directions:

- Detailed instructions for the challenge are included in the readme file in the repository.
- You have 75 minutes to complete the challenge from the time you begin.
- Tests are a mandatory part of the challenge, so please include them. Please add full test coverage to account for the features of the challenge, and make sure the full test suite passes.
- At the end of the challenge please verbally summarize and explain everything you have done and show your UI work.
- Once you have completed the challenge, please email me a link to the video.

Before You Start

Clone repository

Install Ruby version

```
$ rvm install 2.7.3
```

Install PostgreSQL >9.4 and start it

```
$ brew install postgresql
```

Run setup

```
$ bin/setup
```

Run specs

```
$ bin/rspec
```

Start the server

```
$ bin/rails server
```

Open your browser

```
http://localhost:3000
```

Summary

For this code challenge a Candidate will clone and setup an existing Rails application. The application will contain routes, migrations, models, and minimal views but with no actual functionality created. The candidate will show all her/his expertise building apps with the Ruby on Rails framework and problem solving skills.

Overview

HeyURL! is a service to create awesome friendly URLs to make it easier for people to remember. Our team developed some mock views but they lack our awesome functionality.

Requirements

1. Implement actions to create a short URL based on a given full URL
2. If URL is not valid, the application returns an error message to the user
3. We want to be able to provide basic click metrics to our users for each URL they generate.
   i. Every time that someone clicks a short URL, it should record that click
   ii. the record should include the user platform and browser detected from the user agent
4. We want to create a metrics panel for the user to view the stats for every short URL.
   i. The user should be able to see total clicks per day on the current month
   ii. An additional chart with a breakdown of browsers and platforms
5. If someone tries to visit a invalid short URL then it should return a 404 page
6. Controllers, endpoints and models should be fully tested with RSpec

Spec for generating short URLs

- It MUST have 5 characters in length e.g. NELNT
- It MUST generate only upper case letters
- It MUST NOT generate special characters
- It MUST NOT generate whitespace
- It MUST be unique
- `short_url` attribute should store only the generated code

Recommendations

1. Check routes defined in `config/routes.rb`
2. Check controller actions in `app/controllers/urls_controller.rb`
3. Check views in `app/views/urls/`
4. Check existing tests in the `spec` folder

5. Google Charts is already added to display charts but you can use any library
6. Use the `browser` gem already installed to extract information about each click tracked

Pages

The following pages/urls are already built into our app:

1. `GET /`: Contains the form and a list of the last 10 URL created with their click count
2. `GET /:url`: Redirects from a short URL to the original URL and should also track the click event
3. `GET /urls/:url`: Shows the metrics associated to the short URL

API - Optional Bonus Points

We would like to have a way to retrieve the last 10 URLs created using an API endpoint. It should be JSON-API complaint. Here is an example of a response from the API:

```
{
  "data": [
    {
      "type": "urls",
      "id": "1",
      "attributes": {
        "created-at": "2018-08-15T02:48:08.642Z",
        "original-url": "www.fullstacklabs.co/angular-developers",
        "url": "https://domain/fss1",
        "clicks": 2
      },
      "relationships": {
        "clicks": {
          "data": [
            {
              "id": 1,
              "type": "clicks"
            }
          ]
        }
      }
    }
  ],
  "included": [
    {
      "type": "clicks",
      "id": 1,
      "attributes": {
        ...
      }
    }
  ]
}
```

Accomplishment

- Completed functionality 65%
- Completed test 20%
- Completed bonus 15%