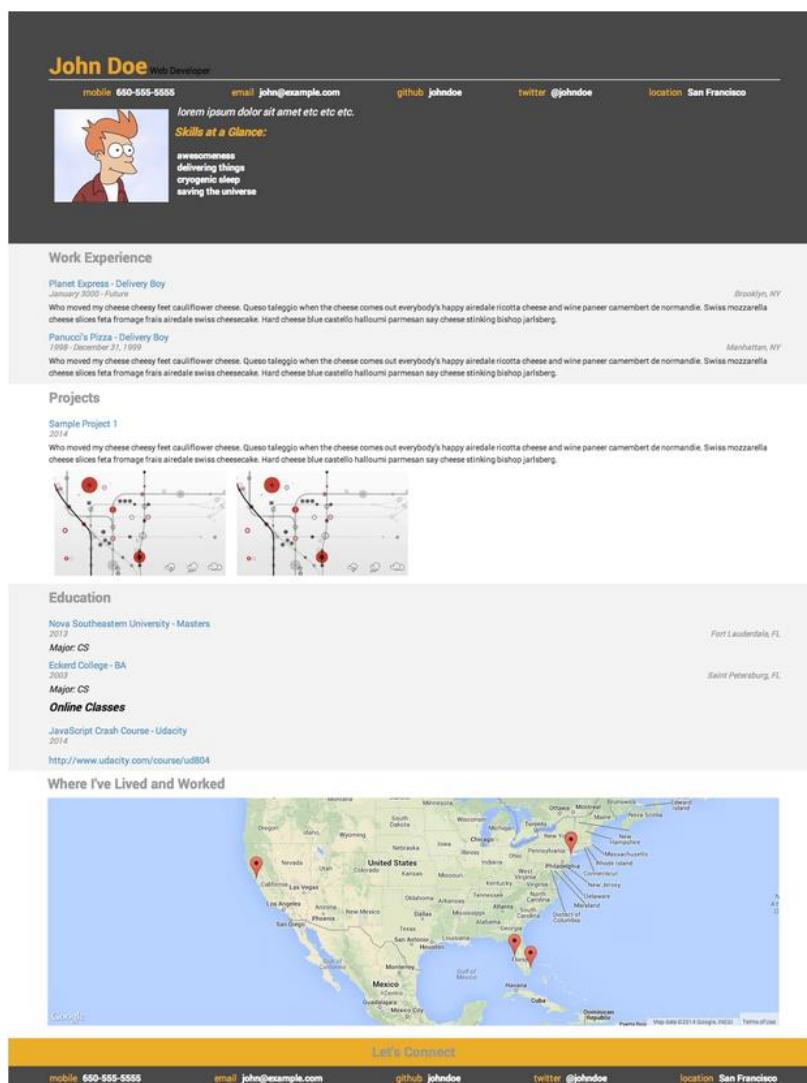# How do I complete this project?

1. If you need a refresher on JavaScript syntax, go to the Javascript Basics course; if you would like to understand how this project is manipulating and traversing the DOM, check out Intro to jQuery.
2. Go through the videos and assignments in this course to learn the JavaScript necessary to build your resume.
3. Read all the project details outlined below and make sure you understand each part. If you are unsure of anything please check with your cohort or private message your cohorts coaches.
4. Fork the project repo from **Github** and begin building you resume.
5. Review your work against the Project Rubric (on the next page).
6. When you are satisfied with your project, submit it according to the Submission Instructions on the next page.

By the end: Your resume will look something like this

And your repository will include the following files:

- **index.html**: The main HTML document. Contains links to all of the CSS and JS resources needed to render the resume, including resumeBuilder.js.
- **js/helper.js**: Contains helper code needed to format the resume and build the map. It also has a few function shells for additional functionality. More on helper.js further down.
- **js/resumeBuilder.js**: This file is empty. You should write your code here.
- **js/jQuery.js**: The jQuery library.
- **css/style.css**: Contains all of the CSS needed to style the page.
- **README.md**: The GitHub readme file.
- and some images in the images directory.

Your starting point...

js/helper.js

Within helper.js, you'll find a large collection of strings containing snippets of HTML. Within many snippets, you'll find placeholder data in the form of `%data%` or `%contact%`.

Each string has a title that describes how it should be used. For instance, `HTMLworkStart` should be the first `<div>` in the Work section of the resume. `HTMLschoolLocation` contains a `%data%` placeholder which should be replaced with the location of one of your schools.

Your process:

The resume has four distinct sections: work, education, projects and a header with biographical information. You'll need to:

1. Build four JSONs, each one representing a different resume section, The objects that you create need to follow the names within the schema exactly:

   o `Bio` contains:

   ```
   o  name : string
   o  role : string
   o  contacts : an object with
   o       mobile: string
   o       email: string
   o       github: string
   o       twitter: string
   o       location: string
   o  welcomeMessage: string
   o  skills: array of strings
   o  biopic: url
   o  display: function
   ```

   o `education` contains:

```
o   schools: array of objects with
o        name: string
o        location: string
o        degree: string
o        majors: array of strings
o        dates: integer (graduation date)
o        url: string
o   onlineCourses: array with
o        title: string
o        school: string
o        date: integer (date finished)
o        url: string
o   display: function
```

o   `work` contains

```
o   jobs: array of objects with
o        employer: string
o        title: string
o        location: string
o        dates: string (works with a hyphen between them)
o        description: string
o   display: function
```

o   `projects` contains:

```
o   projects: array of objects with
o        title: string
o        dates: string (works with a hyphen between them)
o        description: string
o        images: array with string urls
o   display: function
```

2. Iterate through each JSON and append its information to index.html in the correct section.

  o   First off, you'll be using
      jQuery's `selector.append()` and `selector.prepend()` functions to modify
      index.html. `selector.append()` makes an element appear at the end of a selected
      section. `selector.prepend()` makes an element appear at the beginning of a
      selected section.
  o   Pay close attention to the ids of the `<div>`s in index.html and the HTML snippets in
      helper.js. They'll be very useful as jQuery selectors
      for `selector.append()` and `selector.prepend()`
  o   You'll also be using the JavaScript method `string.replace(old, new)` to swap out
      all the placeholder text (e.g. `%data%`) for data from your resume JSONs.
  o   Here's an example of some code that would add the location of one your companies
      to the page:
  o   `var formattedLocation = HTMLworkLocation.replace("%data%",`
      `work.jobs[job].location);`
  o   `$(".work-entry:last").append(formattedLocation);`

- o Use the mockup at the bottom of this document as a guide for the order in which you should append elements to the page.
3. The resume includes an interactive map. To add it, append the googleMap string to `<div id="mapDiv">`.
4. All of your code for adding elements to the resume should be within functions. And all of your functions should be encapsulated within the same objects containing your resume data. For instance, your functions for appending work experience elements to the page should be found within the same object containing data about your work experience.
5. Your resume should also `console.log()` information about click locations. On line 90 in helper.js, you'll find a jQuery onclick handler that you'll need to modify to work with the `logClicks(x,y)` function above it.
6. It's possible to make additional information show up when you click on the pins in the map. Checkout line 174 in helper.js and the Google Maps API to get started.

Evaluation

Your project will be evaluated by a Udacity reviewer according to the rubric below. Be sure to review it thoroughly before you submit. All criteria must "meet specifications" in order to pass.

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
Content	The resume is empty or one or more resume sections (biography, education, work, projects, footer) are incomplete.	All resume sections include real or fake data.	Not applicable.
Function Use	No functions or only some functions are used to build the resume.	All resume sections are built using modular functions which can run in any order.	Not applicable.
Encapsulation	No functions or some functions have been encapsulated inside related objects.	All code to display each resume section is contained within the object with the data for the same section.	Not applicable.
Additional Interactivity	The resume does not contain any interactivities.	The resume includes a map that pulls location data from the resume.	The resume contains an interactivity in addition to the map (ie. menu navigation, polymer components).
Artistic Creativity		All CSS is default	Resume includes custom CSS that alters colors as well as layout of the page.

Submission

When you're ready to submit your project go back to your Udacity Home, click on Project 2, and we'll walk you through the rest of the submission process.

If you are having any problems submitting your project or wish to check on the status of your submission, please email us at **frontend-project@udacity.com** or visit us in the discussion forums.

What's Next?

You will get an email as soon as your reviewer has feedback for you. In the meantime, review your next project and feel free to get started on it or the courses supporting it!