

tretton37

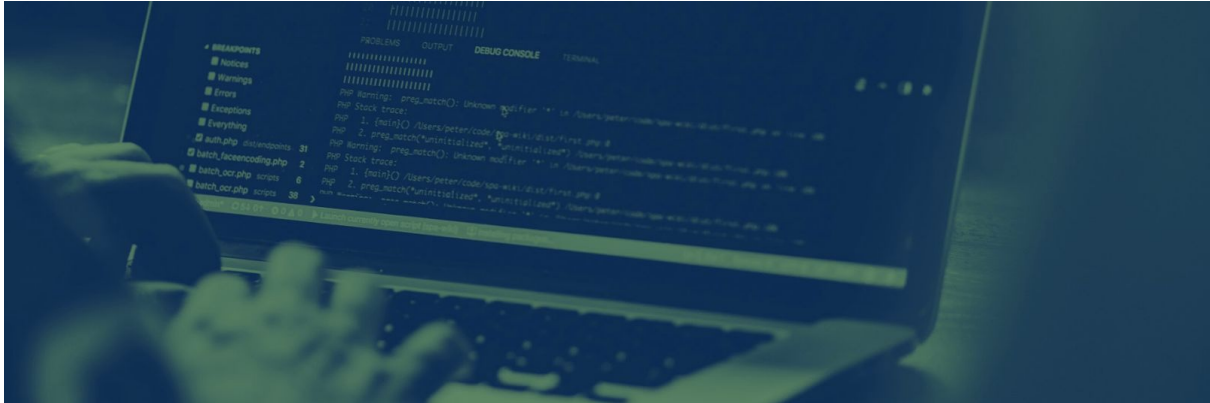
_JS Code Assignment

The fellowship of the tretton37

· Juni 2020 ·

_the fellowship of the tretton37

JS Developer Code Assignment



_background

This code assignment is designed to find out more about your knowledge in code quality, coding best practices, design patterns and written communication skills in the context of Javascript development. The output of this assignment will be the base for evaluation.

We ask you to approach this task as if you were already a member of the **tretton37** family and a part of a customer project.

Please read this description text carefully before you proceed and if anything is unclear, do not hesitate to contact us. We are here for you!

Before you start, we would like to thank you for your interest in **tretton37** and for the time you will spend doing this assignment. We sincerely hope to see you as part of our growing family.

Good Luck!

_delivery details

Submit your solution to a public version control repository (such as git) with a reasonable commit history that gives us some insight into how you work.

Enclose a README with a short description of your product, the thoughts behind the design of your code, a list of the stories you selected and why you chose the features that you did. It should also contain instructions on how to install, build and run the application.

We don't expect (nor want) you to spend more than 3-6 h on this assignment.

Once you are done, send us an email containing any notes/comments along with the link to your repo.

_your assignment

On our tretton37-website we have a page showing all of our colleagues ninja-profiles. You can find it here: <http://tretton37.com/meet>

Your assignment is to create a new and improved site to display all of our tretton37-colleagues. We have a whole bunch of feature requests but in this first iteration **we want you to handle a minimum of 8 points, and no more than 12 points** of the available user stories. We want you to finish the ones you chose as good as possible, rather than maximizing your feature count.

We want at least 2 points to be implemented from each of the requirements areas in the matrix further below in this document, and the rest is up to you! Make sure the points still create a somewhat user friendly page though (all QA and no functionality might make for a dull experience).

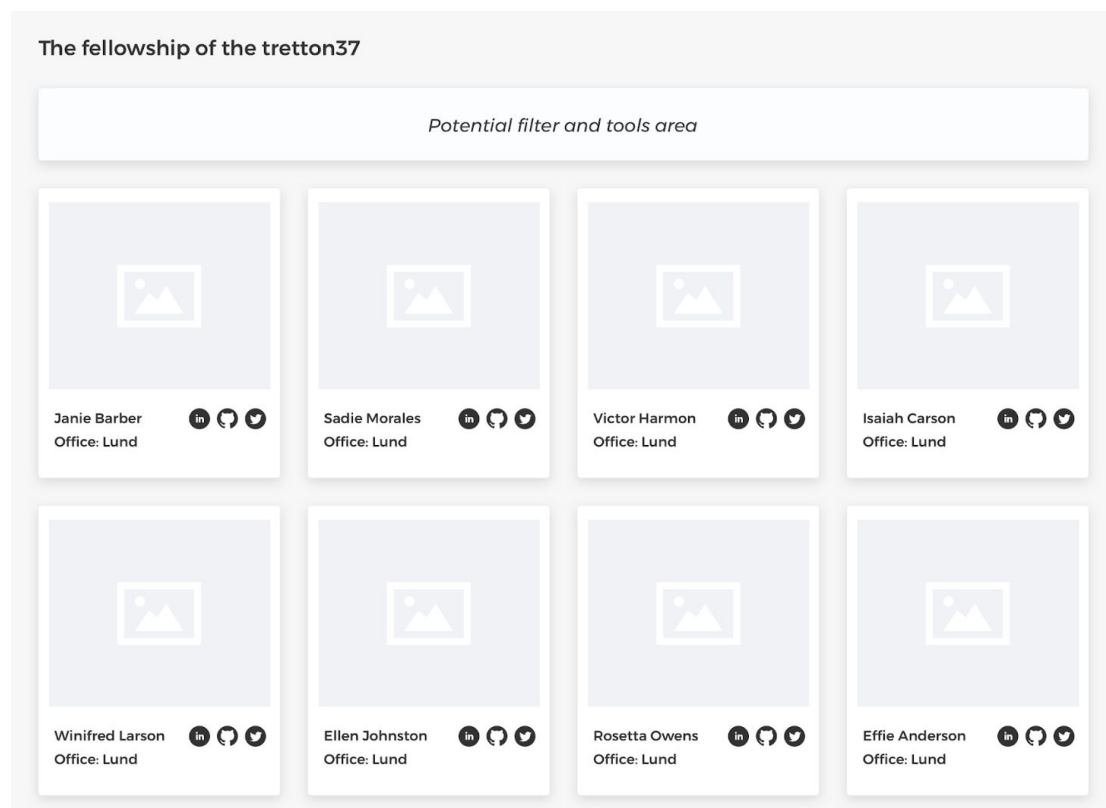
Notice that there is an API available to use below. However, if you are comfortable with NodeJS you could choose to scrape the old site and build an API yourself! :)

Please focus on clean, reusable, understandable & robust code. Use your good judgment to include the right amount of design patterns in your code. It should be structured but not over-engineered. We value deep knowledge of languages & concepts more than knowledge in a specific framework or library.

You can either do a vanilla solution or use a framework, in either case we expect a thought through structure. Design it as if you were to hand it over to another developer for future development.

→ The basic necessities are:

- It somewhat matches the wireframe below
- It works in at least one browser



_features and functional requirements

_design/accessibility	_functionality	_testing/QA
Fancy animations (1 pt)	Sort by name and office (1 pt)	Works in Chrome, Firefox, Edge (1 pt)
Keyboard only function (1 pt)	Filter by name and office (1 pt)	Works in IE11 (1 pt)
A modern design (1 pt)	Filter by contact links (1 pt)	End-to-end testing (with an existing framework) (2 pt)
No UI framework used (such as Bootstrap, Ant) (1 pt)	Enable switch between a grid and a different view (1 pt)	Unit tests for existing functionality (reasonable coverage) (2 pt)
Screen reader function (1 pt)	Available on a free public url (such as Azure, Heroku) (1 pt)	
Responsive design, works on mobile and tablets (2 pt)	CI/CD pipeline from your repo (e.g. Travis, GitLab, Azure) (1 pt)	
	Only render a set of profiles using either infinity scroll , pagination or a load more button (2 pt)	
	Don't use our API, build you own API by scraping the current page and setting up a local backend server (in this case set all offices to Lund as it isn't rendered on the page) (4 pt)	

_resources

Tretton37 ninja API: <https://api.tretton37.com/ninjas>

_questions and help

We expect you to have questions. Have in mind that we are very open to how the product may end up. We are interested to see who you are and what skills you bring to the table.

Many of the feature requests are vague and subjective by design (like: “fancy animations” and “a modern design”) and we expect you to, if you choose a feature you find vague, interpret it as you find best for the end result. I.e. it's as much about how you interpret the requests as how you implement them.

In the “second iteration”, the technical interview, we will talk about potential other features and how they could be implemented.

If there are things, however, that are unclear don't hesitate to contact us and we will get back to you as soon as possible! (emilie.nilsson@tretton37.com or tina.nielsen@tretton37.com)



Good luck from everyone at tretton37!
we've got this, you've got this too!