ActiveFence Jr. Fullstack Candidate Assignment

The **Challenge** is to design and build a simple Spider solution (aka Crawler).

Requirements:

- The spider solution should receive and manage crawling requests.
- The solution should receive a request and extract links that it finds underneath, with relation to its hosting web site.
- The solution should have UI and API for crawling requests.
- A crawling request should include:
  - Start url - the entry point where the crawling starts.
  - Max number of links to return - stop condition.
  - Crawl depth - how deep the crawler should enter (nested links).

Deliverables:

- Design document - Solution guidelines, architecture, system level.
- Implement **one** of the suggested components\services.
  The implementation should be using at least of the following technologies: node.JS, Vue or React.
- All deliverables should be shared over a private repo on **bitbucket.org** with activefence.rnd@gmail.com**.**

Bonus points:

- The solution should support heavy requests: pages with many links, deep crawling (e.g. level 100)
- The solution should have persistence for all requests and their results.
- The solution support scale: High number of requests (1M per day, unknown stress)
- The crawling strategy should be BFS.
- The solution should support real time updates on the crawling request, that should be sent to the client (UI or another system) who requested it.
- Using TypeScript.
- Automated unit testing.

Feel free to reach out to us with any questions.

Good Luck!