



## Skill Assessment: Full-Stack Engineer

---

### Background

👏 A sincere congratulations on progressing your candidacy for the Full-Stack Engineer role at Cascade Debt! This role is essential for building and integrating how users interact with their own data artifacts. Your duties will largely cover design, implementation, and development of Cascade's UI and API. As such, we'd love for you to show off your front-end and dev-ops skillsets to us! 😎

For this final assessment, we ask you to please build and containerise a very simple React app addressing the problem below. Ensure that: (a) you have it cloned to a git-based sharable public repository, (b) you have a README.md file in the repository's root discussing your technical design and choices, and (c) you have a working Dockerfile to container and host your app. You may use your preference for the app's framework (e.g. Express, Nest, Meteor, Koa), however it should be in TypeScript.

---

### Problem: Word Memory



A popular children's game is "Memory" - in which the player needs to find matching visual cards. Your task today is an homage to this game - instead of matching pictures, you'll match words!

The French government has recently contracted your firm to build a language training app for their "visiting American friends".

1. Name your app - give it a nice little splash page with a "GO!" button.
2. After pressing "GO!", change the state of the button to "GRADE" (see step #4) below.
3. Also, after pressing "GO!", your app must scramble the following key-value dictionary:

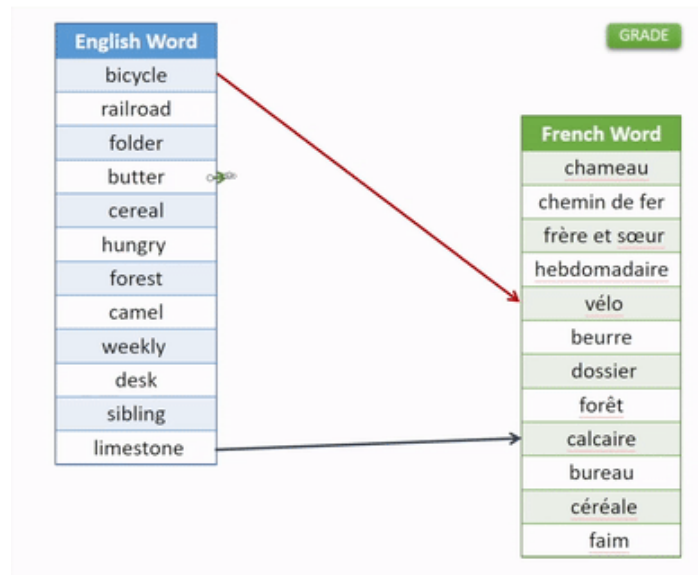
English Word	French Word
forest	forêt
sibling	frère et sœur
cereal	céréale
desk	bureau

camel	chameau
butter	beurre
bicycle	vélo
railroad	chemin de fer
folder	dossier
weekly	hebdomadaire
hungry	faim
limestone	calcaire

After randomising, use React components to present the key-value scramble into two separated columns - something like this:

English Word	French Word
bicycle	chameau
railroad	chemin de fer
folder	frère et sœur
butter	hebdomadaire
cereal	vélo
hungry	beurre
forest	dossier
camel	forêt
weekly	calcaire
desk	bureau
sibling	céréale
limestone	faim

- The goal of this app is then for the user to match the English words to the French words. Build and implement a user-interaction component that does so: e.g. a drag line or similar. As an example, something like this:



4. Upon user click on the "GRADE" button: (a) display the total percentage correct, (b) changes "GRADE" to "GO!" - e.g. reset back to Step #2.
5. [Dockerise your build so all your package dependencies are addressed](#) - please include the [Dockerfile](#) in your submission, as well as your [docker run](#) command to launch your app.

---

## Submission

- **Ensure that your project runs fully BEFORE submission!**
- Verify that you have dockerised your app, and you are confident with your deploy.
- Push and merge into your git repository's main branch.
- Submit your project's git repository URL to either the Rippling comms channel or via email to [joshua@cascadedebt.com](mailto:joshua@cascadedebt.com).