TEUTSCH PARTNERS
Real Estate Services

Although there is no hard deadline for this take-home task, we hope to receive your submission in at most 5 days. The completion of all the tasks should not take more than about 5 or so hours, but we are providing extra time to allow flexibility in how and when you complete the tasks.

Please do not stress too much about including every possible detail mentioned in the questions and limit answers to at most 3 paragraphs, this is not meant to be an exhaustive examination but merely an exercise which tells us more about your web development experience and coding style.

In order to submit, create a copy of this document where you have filled in your answers and email the link to both stefan.alex4@gmail.com and justin.teutsch@teutsch.com. Please ensure you have the permissions set up properly so that we can access the document!

1) Which web project (ideally React) that you worked on are you most proud of and why? Did you work alone or as part of a team, and if so, what were you responsible for? If possible, please include a link and any instructions about how to access it.

// TODO

2) Describe a time when you encountered a serious challenge when working on a coding problem and what steps you took to resolve the issue. How did you approach the problem, and what did you learn?

// TODO

3) Create a calculator web app using React and TypeScript (and no other third-party libraries). The calculator should be capable of performing addition, subtraction, multiplication, division and exponentiation. It should be usable via both keyboard and mouse. Once completed, paste the link to source code and the deployed project below (you can find some starter code here).

// TODO: paste repository link
// TODO: paste project link

4) Choose three design aspects that you focused on while building the calculator project and briefly explain what you did to that end and why. Here are some examples of aspects to help you out:

- UX design (and/or aesthetics)
- Maintainability (e.g. modular code, test cases, easily extensible etc.)
- Readability (e.g. clean code, JSDoc comments, good variable & function names, etc.)
- Robustness and error handling
- Accessibility (e.g. usable by differently abled users, works on mobile, etc.)

// TODO