Welcome to Tutuka!

We're looking forward to the opportunity of working with you!

We've put together a small standardised project that will give you the chance to experience a simple sample of the kind of work you'll be doing, the technical environment we work in, and "*programmatically*" introduce yourself to our development team.

The Project

- The concept behind the project is to perform a financial reconciliation between two different sets of data
- You'll have received these sets of data as two sample CSV files
- All you need to do is compare the two files, and report on how many transactions match perfectly, versus transactions which cannot be matched
- And those transactions which cannot be matched will need to be reported on, so that a third party could refer to the report and investigate the exceptions
- If a transaction cannot be matched perfectly, you should attempt to look for any close matches and suggest them as possibilities
- Note that this is *not* a file comparison project, this is a transaction matching/reconciliation project
- You do not need to store the results, or provide any further functionality once you have listed the exceptions (and potential matches where possible)
- We've also provided you with Balsamiq screen mockups to give you an example of the flow of the process

Requirements

- For the most part, you are welcome to code in the language of your choice… but if it's something not typically considered mainstream (Java, Ruby, PHP, Python, etc.) please check with us first – we'll need to confirm we have developers that can suitably review the code in that language
- You'll need to provide a working version that can be run as an application, or hosted by you on a public web server, as well as a Zip file or repo for the code

Hints

- The Balsamiq mockups that have been provided are a guide, and the final design can be modified how you see fit
- But other than that, the project has purposefully been left "open-ended" and even quite vague, more so than would ever happen in a real-world task
- The intention is to see what choices you make when given freedom to do so
- So there are no instructions in terms of how you complete the task, but you'll be evaluated against the following criteria:
    - Layout and design
    - TDD/BDD
    - Exception handling
    - Algorithm
    - Speed of recon
    - Modularity of code
    - Neatness of code
    - Self-documenting code

    - Extra effort
    - Overall impression
- You're obviously encouraged to put your best foot forward and use the opportunity to impress the team, who will be your colleagues if you end up joining us

Communication

- For the trial project, please send an email to mathieu@tutuka.com after you have had a chance to work on the project, describing what development you have completed since your last email, what development you plan to complete during that session, and whether there are any questions or issues blocking your progress
- We use the Scrum methodology of Agile, which includes daily Scrum meetings, and this is in essence a "virtual scrum", letting us evaluate your communication skills, confirm you're making progress, or allows us to assist you if you are stuck

Source Code Submission Please submit the project source files for review in a single ZIP or TAR (gzipped) file. If available - include source code repository in the archive.We advise against using public repositories for storing the project. If the resulting archive is too large to be emailed, then cloud storage services (such as Dropbox, Google Drive, and similar) may be used and a link sent instead.

Good luck!
Mathieu