# Welcome to Paymentology!

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. In other words, the project should do its best to identify for users all non-perfectly matched records possible matches based on a reconciliation strategy you come up with , it might be you consider only the most important fields, or consider all fields but at different level of importance... Defining the important fields is up to you, the TransactionID for example might be the only field you consider or one important field out of several, or it may not. There is no right or wrong answer, please feel free to make logical assumptions in your reconciliation strategy, detail what they are and deliver that documentation alongside the project code.
- 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, in line with the job description, we expect you to code in Java (or at least a JVM based language such as Kotlin) where relevant. Obviously this wouldn't apply to the UI portion, which can be any language of your choice… but if it's something not typically considered mainstream (ReactJS, 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 hosted by you on a public web server (using for example a service such as Heroku or similar), as well as a Zip file for the code. Sharing private repositories creates too much admin work to ensure everybody on our side involved in reviewing projects has the access they need, and public repositories mean the code is publicly available and can potentially be copied by other candidates so we cannot accept submissions with a link to any respository.

## 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 the person who did your Technical Interview (they will be your point of contact during the project development phase) when you have 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. To confirm, it doesn't need to daily, just any time you get the chance to progress on the project.

Good luck!