Paymentology Transaction Reconciliation

Web Application Documentation

Catalog

1. Assumption:

As specified in the Welcome to Paymentology project brief, in order to complete the given project, certain assumptions were made. First of all, I assumed the two files were modeled from a relational database and as such surmised that using the transactionID column may not give me the most accurate comparison result and as such chose instead to use the WalletReference, TransactionAmount and TransactionDate as comparison keys.

Furthermore, I did not forgo the possibility of having a missing column in record and as well accounted for this in my assumptions, therefore, when getting the unmatched records, I only compared both files using the WalletReference. After this, I compared the compiled unmatched records on both files to find possible matches using WalletReference, TransactionAmount and TransactionDate as comparison keys.

1.1 Technologies Used:

The technologies and dependencies used for this project are:
FrontEnd: ReactJs, Tailwindcss, Axios, ReactDom, React Loader Spinner and React Toastify
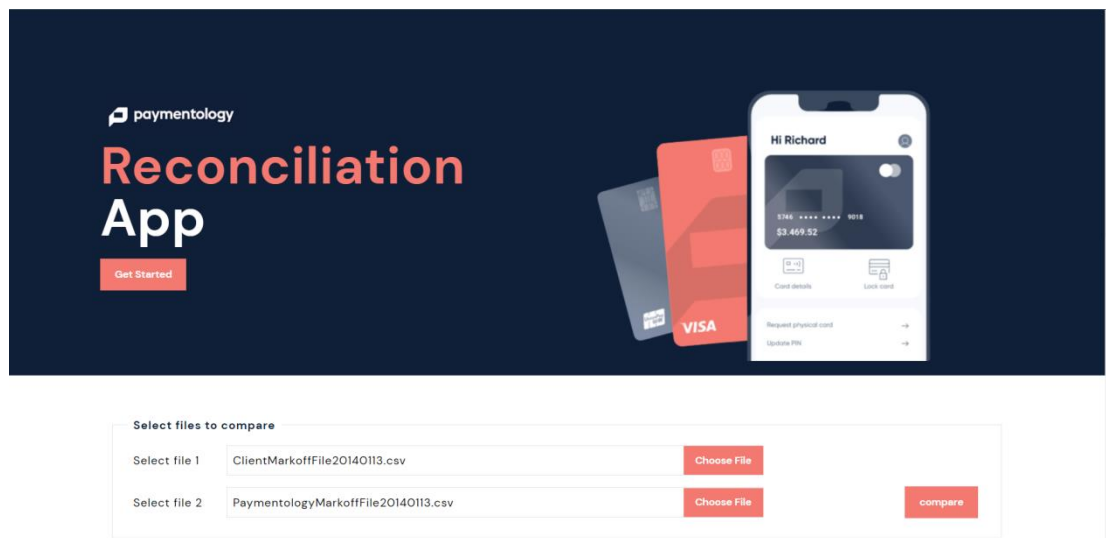**BackEnd:** Java, Springboot, Univocity and jUnit

2. Interface:

Figure 1: Landing page of web app

The landing page of the web app takes its design elements from the paymentology official website, sporting the same color profiles and subtleties

Figure 2: Result Component

The result component follows the balsamic mockup given in the devtrial zip file with additions of course.

As can be seen in the image, each record which could not be matched is stated with a badge which says unmatched record and beside it would be the closest possible matching record. If there are no close matches, the identifier N/A is written in its place signifying that there are no close matches currently available.