

Harver Assignment

Problem Context

The matching engine is a small application that finds the best candidates for a vacancy. You are tasked with implementing this engine. It should receive an input text that represents a candidate's application with their module scores and return the best candidates per vacancy.

High level description

You are given an input file with two sections split by an "=" sign. The top section represents the vacancies and their hiring limits, while the bottom section will contain an applicant per line with its module scores and which vacancy they applied to.

The best candidates for a vacancy are the candidates with the highest overall score. The overall score is the rounded average of non "X" module scores. "X" means the module does not belong to the vacancy, thus should not be considered.

The order of the applicants in the file represents the order in which they submitted the application. If the overall score is the same, hiring priority should be given to who applied first.

See example:

Input file / expected output file:

Vacancy Id,Hiring Limit

5c0e2314a1c9e9714fe3b2f2,2

=

Vacancy Id,Candidate Id,Module1,Module2,Module3,Module4,Module5

5c0e2314a1c9e9714fe3b2f2,5bffe0fb3a8d783e648f8fdd,2,30,24.5,X,X

5c0e2314a1c9e9714fe3b2f2,5bffe0fb3a8d783e648f8fdc,20,30,50,X,X

5c0e2314a1c9e9714fe3b2f2,5c00fd874e843c3e6ab79911,40,40,60,X,X

5c0e2314a1c9e9714fe3b2f2,5c069d933a8d783e64907809,5,50,100,X,X

5c0e2314a1c9e9714fe3b2f2,5c00981e4e843c3e6ab784e9,20,60,22,X,X

VacancyId,CandidateId,Overall Score

5c0e2314a1c9e9714fe3b2f2,5c069d933a8d783e64907809,52

5c0e2314a1c9e9714fe3b2f2,5c00fd874e843c3e6ab79911,47

Requirements

Please note that your solution should meet the following requirements:

- The assignment should be done in JavaScript or Typescript
- Harver will test your application against multiple different input files. Please ensure its correctness
- Performance will not be the primary focus of the assessment. Emphasize simplicity, maintenance, and the development of production-ready code for both server and client systems.

- An initial template is provided together with this assignment. Feel free to change it in any way you want. You can add new dependencies, completely change its structure, or create a new one from scratch if you wish.
- Provide the solution in a zipped folder

Constraints

- CandidateId and VacancyId will be in ObjectId format
- Module Scores will range from 0 to 100, allowing a maximum of 2 decimal points.
- Each Vacancy can have different number of modules.
- Hiring Limit will be below 1,000,000.



Glossary

- **Candidate**

A person who applies for the job.

- **Vacancy**

A (generally) unoccupied position or job (eg: Cashier, Pizza Chef, Store Manager, etc ...). It involves defining the role (such as Cashier, Pizza Chef, Store Manager) and outlining the required qualifications, responsibilities, and sometimes specific limits on the number of candidates that can be hired.

- **Vacancy Hiring Limit**

It refers to the predefined or allocated number of individuals that a company plans to hire for a vacancy.



- **Module**

Defines an assessment step to measure different skills of a candidate. Each module represents a distinct area of skills or knowledge required for the job.

- **Module Score**

The score for each module is derived from the candidate's answers within that specific section.

- **Overall Score**

Average of all the module scores.

Input

The engine should function with a provided input file through the user interface. You can find a sample input file at **resources/input.txt**.

Upload the input file

Choose file

No file chosen

Upload

Result:

Output

Once a file is uploaded, the engine's result should be visible on the user interface. You can refer to a sample output file at [resources/output.txt](#).

Upload the input file

input.txt

Result:

VacancyId,CandidateId,Overall Score

5c0e2314a1c9e9714fe3b2f2,5c069d933a8d783e64907809,52

5c0e2314a1c9e9714fe3b2f2,5c00fd874e843c3e6ab79911,47

5bd1b930c430d917041836bd,5c00981e4e843c3e6ab784e8,55

5c0e231500411d7154949e1b,5c01471f3a8d783e648fc35b,61

5c0e231500411d7154949e1b,5c014da24e843c3e6ab7b538,49

5bd1b98ddd59b417229619a0,5c0157b33a8d783e648fc713,36