Why an Assignment?

We are asking for software engineers to do a coding assignment so that we can get an accurate picture on how you code. An assignment you can complete at your own pace is more realistic and less stressful than whiteboard interview questions.

The assignment

This assignment is designed to be completed on your own time. You may use any resources you need to complete it, including emailing questions to me. However, please be sure that you've googled and read docs thoroughly prior to asking questions.

Rules and Guidelines

- You are showcasing the readability and cleanliness of your code in addition to working code.
- Provide unit tests. The cleanliness and readability of tests is just as important as your production code.
- Think simple. Readability and modularity are better than being clever.
- Make a github or gitlab repository and commit your work in small cohesive chunks.

Task 1 - Build a system to parse and sort a set of records

Create a command line app that takes a list of input files in different formats, combines and sorts the records and writes the results to the screen.

Input A record consists of the following 5 fields:

- Last Name
- First Name
- Gender
- Favorite Color

- Date of Birth

There are 3 types of input files, each containing records stored in a different format. You may generate these files yourself, and you can make certain assumptions if it makes solving your problem easier.

- **Pipe-delimited**: LastName | FirstName | Gender | Favorite Color | Date Of Birth
- **Comma-delimited:** LastName, FirstName, Gender, Favorite Color, Date Of Birth
- **Space-delimited:** LastName FirstName Gender FavoriteColor DateOfBirth

You may assume that the delimiters (commas, pipes and spaces) do not appear anywhere in the data values themselves.

Write a program in Python or Javascript to read-in records from these files and combine them into a single set of records.

Output Create and display 3 different views of the data you read in: Output 1 – sorted by gender (females before males) then by last name ascending. Output 2 – sorted by birth date, ascending. Output 3 – sorted by last name, descending.

Display dates in the format M/D/YYYY.

Task 2 - Build a REST API to access your system

Tests for this section are required as well. Within the same code base, build a standalone REST API with the following endpoints:

- POST /records - accepts a record, or json array of records in any of the 3 formats from step 1
- GET /records/gender - returns all records in json format, sorted by gender
- GET /records/birthdate - returns all records in json format, sorted by birth date
- GET /records/name - returns all records in json format, sorted by name

These endpoints should return JSON and should return all records that may have been

added with multiple POSTs.

There does not need to be any UI.

Keep it simple and keep all records in memory, do not use a persistent datastore.