ENSEK REMOTE TECHNICAL EXERCISE

FOR SOFTWARE ENGINEERING

Thanks for agreeing to do our remote technical test
The brief, plus two data files should be included in your pack

We hope you enjoy it. Good Luck.

ENSEK REMOTE TECHNICAL EXERCISE

01 TASK

Ensuring the Acceptance Criteria are met, build a C# Web API that connects to an instance of a database and persists the contents of the Meter Reading CSV file.

We have provided you with:

- A list of test customers along with their respective Account IDs
- Please refer to Test_Accounts.csv
- Please seed the Test_Accounts.csv data into your chosen data storage technology and validate the meter read data against the accounts

02 USER STORY

As an Energy Company Account Manager, I want to be able to load a CSV file of Customer Meter Readings so that we can monitor their energy consumption and charge them accordingly

03 ACCEPTANCE CRITERIA

MUST HAVE

- Create the following endpoint:

  POST => /meter-reading-uploads

- The endpoint should be able to process a CSV of meter readings. An example CSV file has been provided (Meter_reading.csv)

- Each entry in the CSV should be validated and if valid, stored in a DB.

- After processing, the number of successful/failed readings should be returned.

- Validation:
  - You should not be able to load the same entry twice
  - A meter reading must be associated with an Account ID to be deemed valid
  - Reading values should be in the format NNNNN

NICE TO HAVE

- Create a client in the technology of your choosing to consume the API. You can use angular/react/whatever you like

- When an account has an existing read, ensure the new read isn't older than the existing read

ENSEK REMOTE TECHNICAL EXERCISE

04 ATTACHED MATERIALS

- Meter_reading.csv (test file for uploading meter readings)
- Test_accounts.csv (use this to seed your DB)

05 OUR TOP TIPS

We want you to be able to give of your best in this exercise so here are some pointers on what we look for when we mark it:

- Readable, self-explanatory code
- Some evidence of SOLID principles
- The creation of clearly testable code
- Evidence of thorough unit testing
- Easily maintainable code
- Having a user interface is a bonus

- Overall, we are looking for clarity of code, understanding of key and core modern coding principles and for you to put your mark on the exercise and enjoy it along the way.

06 NEXT STEPS AND HOW TO SHARE YOUR SOLUTION WITH US

- So that we can turn your application around as quickly as possible we'd suggest you aim to complete the exercise within 48 hours of receiving it, wherever possible.

- We strongly advise that you use a publicly accessible Git repository to commit and share your work with us.

- Please send your work to careers@ensek.co.uk. We'll pick it up there and aim to carry out an initial assessment of it.

- If the initial assessment of your solution looks good, then be invited to a final stage interview where you'll be asked to present the solution you've created and discuss it.

- The purpose of this discussion is to help us understand how you went about designing the solution, design choices, testing etc. Its your chance to tell us how it went and share your coding philosophy with us in a collaborative conversation.

- Please make sure you have an appropriate way of presenting your solution (usually this is done via a screenshare + IDE)