Solved by: Ojaswini Nimbalkar

Dare

(and stack overflow)

**Problem 2**

Data processing

Generate a csv file containing first\_name, last\_name, address, date\_of\_birth

Process the csv file to anonymise the data

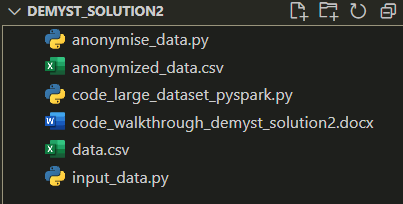
Columns to anonymise are first\_name, last\_name and address

You might be thinking that is silly

Now make this work on 2GB csv file (should be doable on a laptop)

Demonstrate that the same can work on bigger dataset

Hint - You would need some distributed computing platform



**Solution Walkthrough:**

**1] Generating the CSV File:**

First, we'll generate a sample CSV file containing `first\_name`, `last\_name`, `address`, and `date\_of\_birth` columns. This step involves creating a script **(`input\_data.py`)** to generate random data and write it to a CSV file **(`data.csv`).**

**>>run input\_data.py**

**2] Anonymizing the CSV File:**

Next, we'll create a script **(`anonymise\_data.py`)** to read the `data.csv` file, anonymize the `first\_name`, `last\_name`, and `address` columns, and write the anonymized data to a new CSV file **(`anonymized\_data.csv`).**

**>>run anonymise\_data.py**

**3] Handling Large Datasets:**

For larger datasets beyond what a single machine can handle efficiently (e.g., multi-GB or TB scale), consider using a distributed computing platform like Apache Spark.

(I installed spark locally for this as my laptop GPU was able to support the load)

**>>run code\_large\_dataset\_pyspark.py**

The above using Spark can work on larger datasets as well, we can use cloud environments(AWS, Azure, GCP) to host this service and store data there.