Python Flask APIs

Web APIs using Flask in Python.
On startup, the application reads transactions data from `transactions.csv` file and product reference data from `productReference.csv` file.
This is an in-memory application, so persistent storage is not used.

Available APIs

`http://localhost:5000/assignment/transaction/<transactionId>`

Provides transaction details of a given Transaction ID.
Input needs to be a valid Transaction ID `<transactionId>` (an Integer) else it will show error.
e.g. `http://localhost:5000/assignment/transaction/1`
Output JSON format:

```
{
  "productId": 80,
  "transactionAmount": 3500.0,
  "transactionDatetime": "Thu, 19 Mar 2020 23:02:53 GMT",
  "transactionId": 1
}
```

`http://localhost:5000/assignment/transactionSummaryByProducts/<lastNDays>`

Provides Summary by Products for the transactions during the last N days.
Input needs to be a valid number of days `<lastNDays>` (an Integer) else it will show error.
e.g. `http://localhost:5000/assignment/transactionSummaryByProducts/4`
Output JSON format:

```
{
  "summary": [
    {
      "productName": "P6",
      "totalAmount": 6000.0
    },
    {
      "productName": "P5",
      "totalAmount": 5000.0
    },
    {
      "productName": "P4",
      "totalAmount": 15500.0
    },
    {
      "productName": "P3",
```

```
      "totalAmount": 6000.0
    },
    {
      "productName": "P2",
      "totalAmount": 8000.0
    },
    {
      "productName": "P9",
      "totalAmount": 12000.0
    },
    {
      "productName": "P1",
      "totalAmount": 1000.0
    }
  ]
}
```

`http://localhost:5000/assignment/transactionSummaryByManufacturingCity/<lastNDays>`

Provides Summary by Manufacturing City for the transactions during the last N days.
Input needs to be a valid number of days `<lastNDays>` (an Integer) else it will show error.
e.g. `http://localhost:5000/assignment/transactionSummaryByManufacturingCity/4`
Output JSON format:

```
{
  "summary": [
    {
      "cityName": "C3",
      "totalAmount": 21500.0
    },
    {
      "cityName": "C2",
      "totalAmount": 11000.0
    },
    {
      "cityName": "C1",
      "totalAmount": 9000.0
    },
    {
      "cityName": "C6",
      "totalAmount": 12000.0
    }
  ]
}
```

Prerequisites

- Python 3 should be installed on your machine.

Steps to run the server on Windows

Create an environment

```
py -3 -m venv env
```

Activate the environment

```
env\Scripts\activate
```

Install Flask

```
pip install Flask
```

Run the app

```
python app.py
```

The app would now start running on Port 5000. You can press CTRL+C to quit.

Deactivate the environment

```
deactivate
```