LINK TO REPOSITORY: [Shyamalasridar/exa-data-eng-assessment (github.com)](https://github.com/Shyamalasridar/exa-data-eng-assessment)

1. **Objective:**
   * The objective of this task was to convert JSON data fetched from remote sources into a structured tabular format using Python, and then store it either as CSV files or directly into a SQL Server database.
2. **Tools and Libraries Used:**
   * **Python:** Programming language used for data processing.
   * **Jupyter Notebook:** Interactive computing environment for Python.
   * **Pandas:** For data manipulation and transformation, it provides a powerful DataFrame structure for tabular data representation.
   * **Requests:** Library for making HTTP requests.
   * **SQLAlchemy:** Utilized for database operations in Python, enabling interaction with SQL databases.
   * **SQL Server:** Chosen as the target database for storing the transformed data.
3. **Process Overview:**
   * **Data Retrieval:** Fetched JSON data from remote sources using the requests library, accessing URLs provided for each JSON file.
   * **Data Transformation:** Parsed the JSON data and organized it into structured tabular formats using Pandas DataFrame. Extracted relevant information from the JSON structure and mapped it to DataFrame columns.
   * **Storage Options:**
     + *Export to CSV:* Utilized the to\_csv() method provided by Pandas to export DataFrame data into CSV files. This approach ensures easy access and sharing of the transformed data in a standard format.
     + *Import into SQL Server:* Established a connection to the SQL Server database using SQLAlchemy and inserted DataFrame data into SQL Server tables using the to\_sql() method. This approach ensures data storage in a relational database, enabling efficient querying and manipulation.
4. **Execution:**
   * **Environment Setup:** Ensured that all necessary dependencies, including Python, Pandas, Requests, and SQLAlchemy, were installed.
   * **Configuration:** Specified the URLs of the JSON files to be processed and configured the SQL Server credentials and database settings as required.
   * **Execution:** Ran the code in a Jupyter Notebook or Python environment, validating the output to ensure correct transformation and storage of the data.

**Code:**

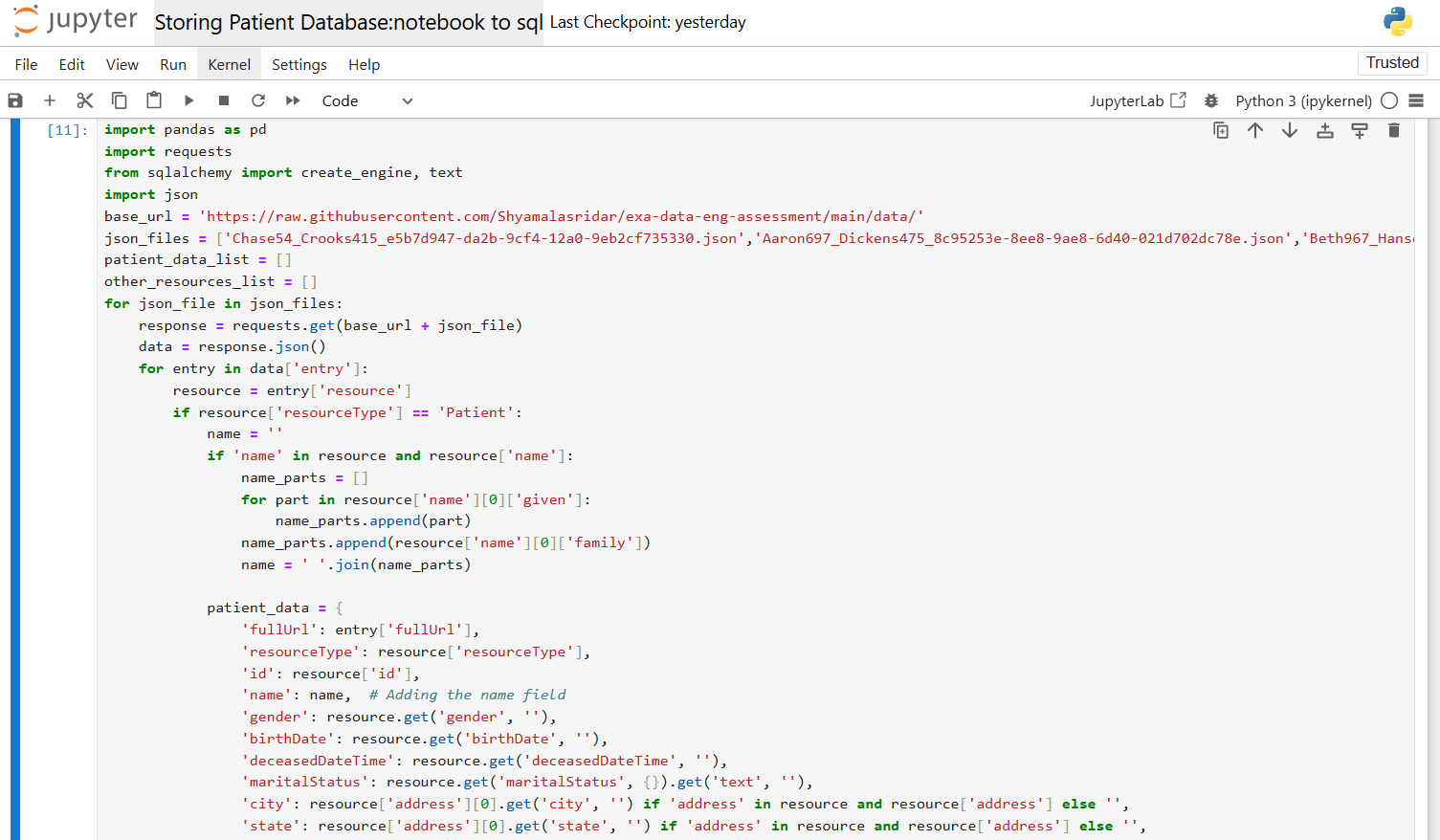
* + Included the Python code snippets used for fetching JSON data, transforming it into tabular formats, and storing it in CSV files or SQL Server database tables.

[C:\Users\AMSAS\Downloads\EMIS\Storing Patient Database\_notebook to sql.ipynb](file:///C:\Users\AMSAS\Downloads\EMIS\Storing%20Patient%20Database_notebook%20to%20sql.ipynb)

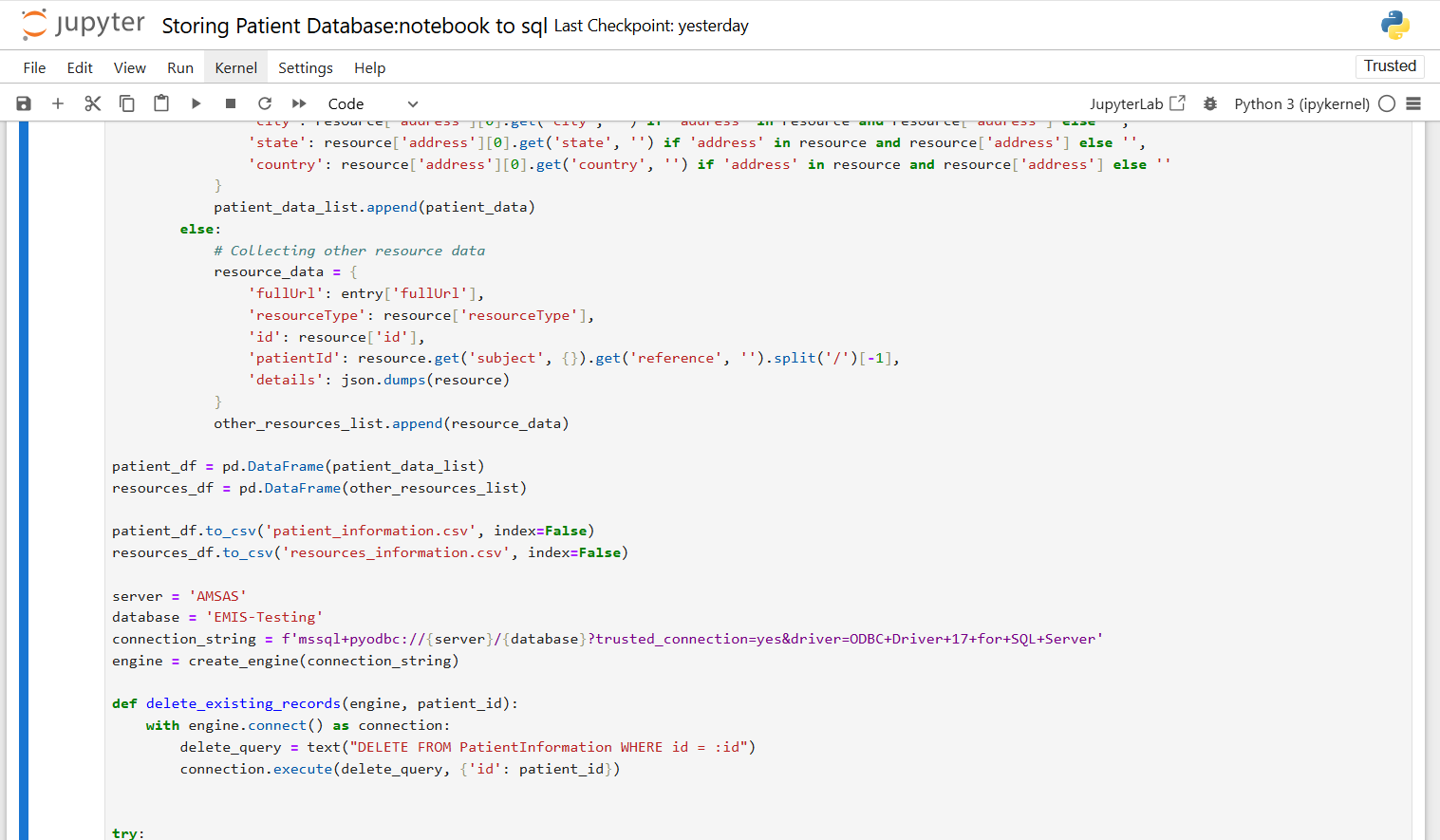
**Link tot Python notebook**

[Storing Patient Database:notebook to sql](http://localhost:8888/notebooks/Storing%20Patient%20Database:notebook%20to%20sql.ipynb)

* + Below Python code imports the necessary library required for the transformation and connect to the URL.
  + It creates an empty list to fetch and process the json file once the response is successful.
  + The script iterates over each entry in the JSON data.
  + If the resourceType is Patient, it extracts relevant patient details (like name, gender, birth date, etc.) and appends them to patient\_data\_list.
  + If the resourceType is not Patient, it collects the resource details and appends them to other\_resources\_list.



* + Create DataFrames from the Collected Data:
  + Save DataFrames to CSV Files:
  + Define the SQL Connection String



* + In a try block, the script iterates over the DataFrame rows.
  + For each patient record, it first deletes any existing record with the same ID and then inserts the new record.
  + It then inserts other resource records into a separate table.
  + If any exception occurs, it prints an error message.
  + Finally, it closes the database connection.



**DATABASE:**

Now, connect to SQL Server Management Studio (SSMS) and verify the results in your tables

* + **Server name**: AMSAS (Server which I have created)
  + **Authentication**: Choose the appropriate authentication method (I have chosen Windows authentication)
  + **DB**: EMIS-Testing

Once connected, expand the Database to see the list of tables created. Below is the transformed data I have fetched from provided json file.

