Mapping Process Documentation

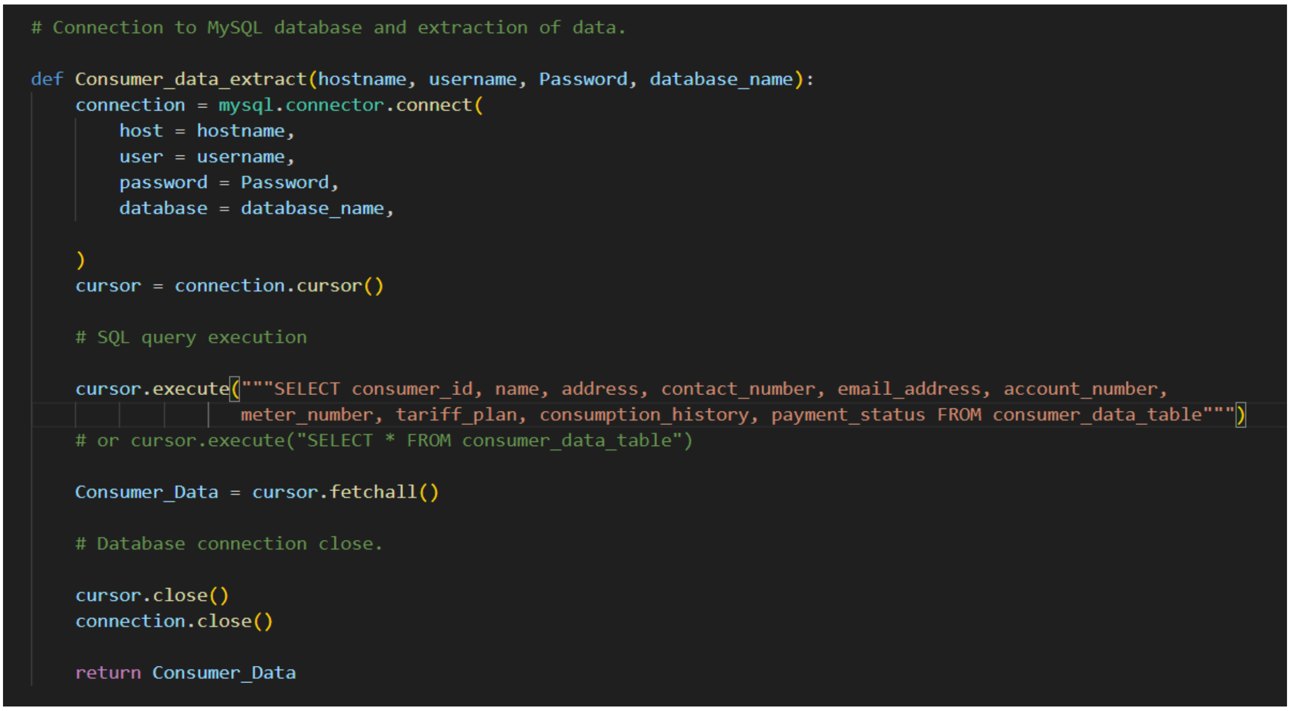
Objective:

The objective of this document is to provide a detailed overview of the process involved in mapping consumer data extracted from a MySQL database into the SMART360 format.

Process Overview:

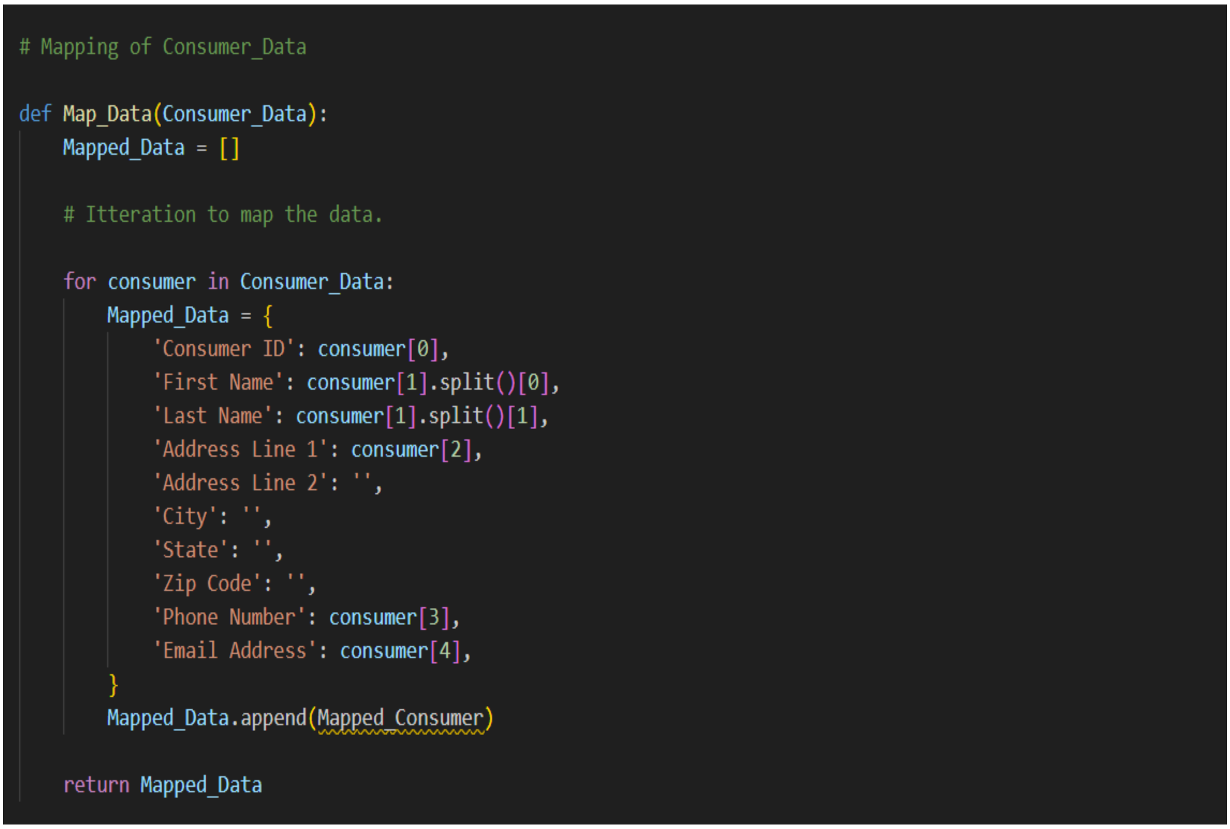
The mapping process consists of transforming raw consumer data into a standardized format compatible with the SMART360 platform. It involves extracting relevant fields from the source data, mapping them to corresponding fields in the SMART360 format, and preparing the data for ingestion into the platform.

1. **Data Extraction:-**
   - Connect to the MySQL Database containing the data.
   - Execute an SQL query to extract consumer data from the specific table "consumer_data_table".
   - Fetch the following fields:
     Consumer ID
     Name
     Address
     Contact Number
     Email Address
     (Optionally) Account Number
     (Optionally) Meter Number
     (Optionally) Tariff Plan
     (Optionally) Consumption History
     (Optionally) Payment Status



```python
# Connection to MySQL database and extraction of data.

def Consumer_data_extract(hostname, username, Password, database_name):
    connection = mysql.connector.connect(
        host = hostname,
        user = username,
        password = Password,
        database = database_name,

    )
    cursor = connection.cursor()

    # SQL query execution

    cursor.execute("""SELECT consumer_id, name, address, contact_number, email_address, account_number,
                   meter_number, tariff_plan, consumption_history, payment_status FROM consumer_data_table""")
    # or cursor.execute("SELECT * FROM consumer_data_table")

    Consumer_Data = cursor.fetchall()

    # Database connection close.

    cursor.close()
    connection.close()

    return Consumer_Data
```

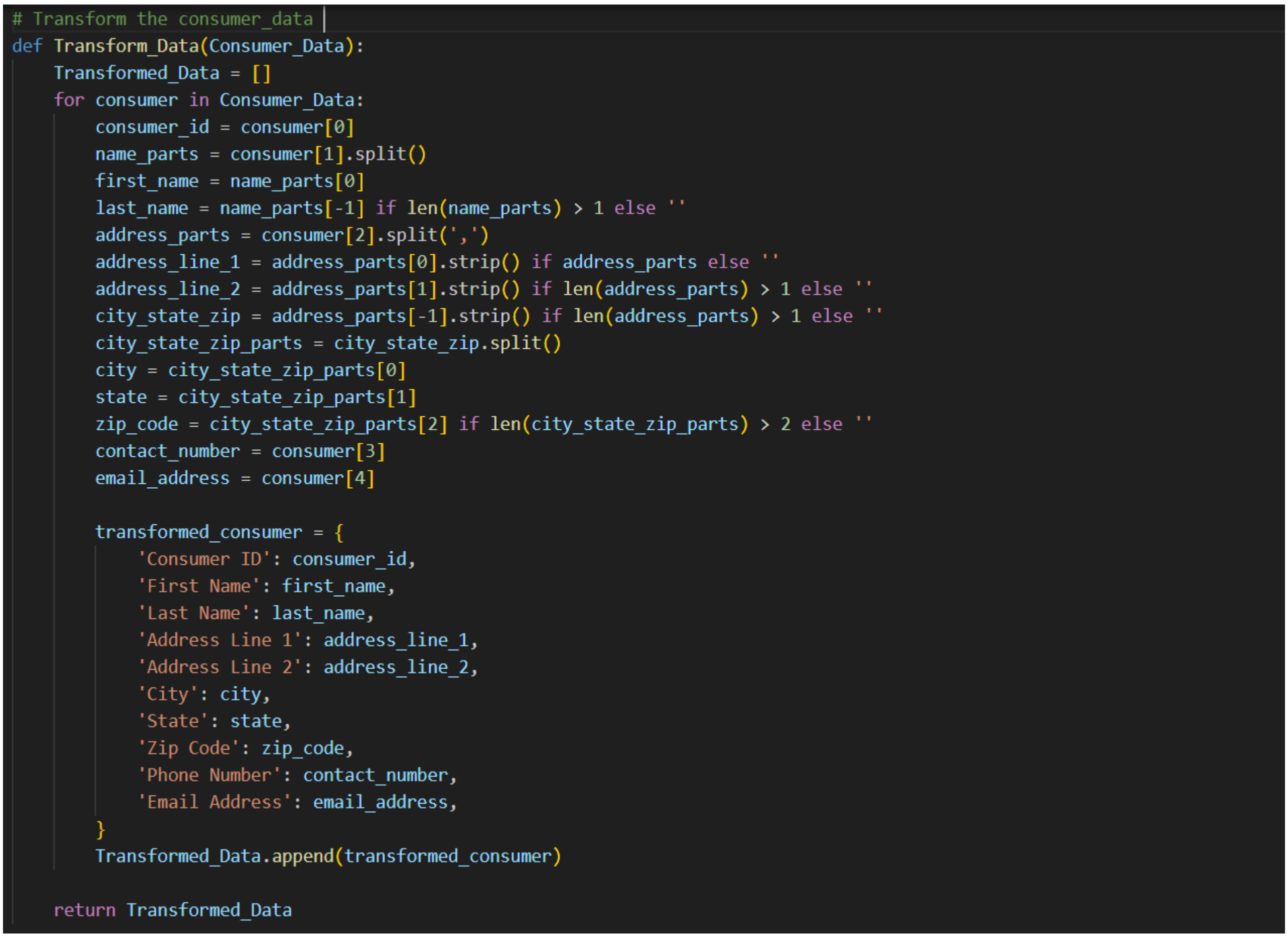
2. **Data Mapping:**
    - Split the 'Name' field into 'First Name' and 'Last Name'.
    - Extract individual components (e.g., street, city, state, zip code) from the 'Address' field if necessary.
    - Map the extracted fields to the corresponding fields in the SMART360 consumer table:
      Consumer ID
      First Name
      Last Name
      Address Line 1
      Address Line 2
      City
      State
      Zip Code
      Phone Number
      Email Address



```python
# Mapping of Consumer_Data

def Map_Data(Consumer_Data):
    Mapped_Data = []

    # Itteration to map the data.

    for consumer in Consumer_Data:
        Mapped_Data = {
            'Consumer ID': consumer[0],
            'First Name': consumer[1].split()[0],
            'Last Name': consumer[1].split()[1],
            'Address Line 1': consumer[2],
            'Address Line 2': '',
            'City': '',
            'State': '',
            'Zip Code': '',
            'Phone Number': consumer[3],
            'Email Address': consumer[4],
        }
        Mapped_Data.append(Mapped_Consumer)

    return Mapped_Data
```

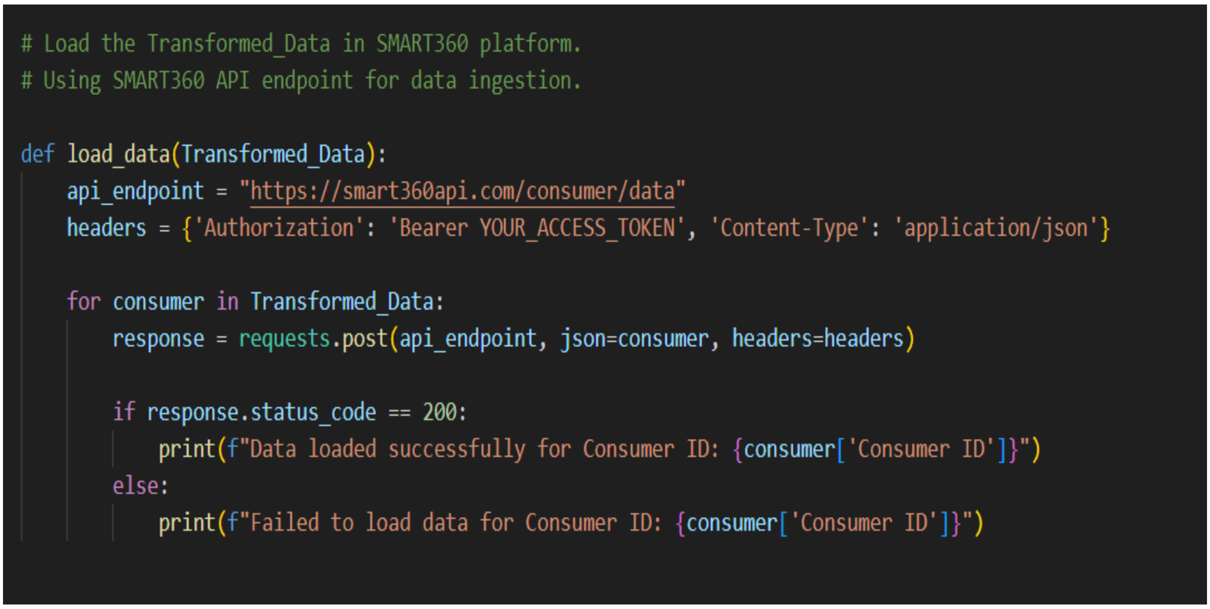
3. Transformation and Loading:

- Transform the extracted and mapped data into the format required by the SMART360 platform.
- Prepare the transformed data for loading into the SMART360 platform.

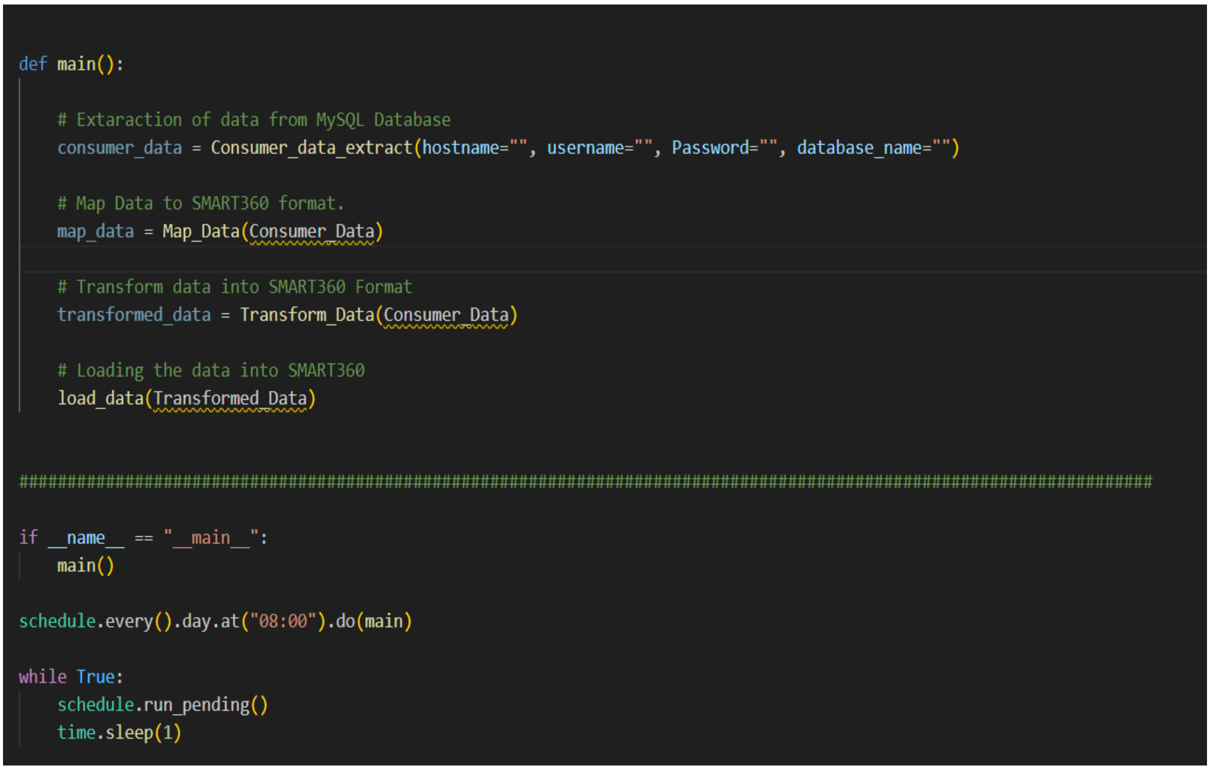


```python
# Transform the consumer_data
def Transform_Data(Consumer_Data):
    Transformed_Data = []
    for consumer in Consumer_Data:
        consumer_id = consumer[0]
        name_parts = consumer[1].split()
        first_name = name_parts[0]
        last_name = name_parts[-1] if len(name_parts) > 1 else ''
        address_parts = consumer[2].split(',')
        address_line_1 = address_parts[0].strip() if address_parts else ''
        address_line_2 = address_parts[1].strip() if len(address_parts) > 1 else ''
        city_state_zip = address_parts[-1].strip() if len(address_parts) > 1 else ''
        city_state_zip_parts = city_state_zip.split()
        city = city_state_zip_parts[0]
        state = city_state_zip_parts[1]
        zip_code = city_state_zip_parts[2] if len(city_state_zip_parts) > 2 else ''
        contact_number = consumer[3]
        email_address = consumer[4]

        transformed_consumer = {
            'Consumer ID': consumer_id,
            'First Name': first_name,
            'Last Name': last_name,
            'Address Line 1': address_line_1,
            'Address Line 2': address_line_2,
            'City': city,
            'State': state,
            'Zip Code': zip_code,
            'Phone Number': contact_number,
            'Email Address': email_address,
        }
        Transformed_Data.append(transformed_consumer)

    return Transformed_Data
```

4. **Data Loading:**
    - Use the SMART360 API endpoint for data ingestion.
    - Send a POST request with the transformed consumer data as a JSON payload.
    - Ensure proper authentication using an access token provided by SMART360.
    - Handle responses from the API to verify successful data loading.



```python
# Load the Transformed_Data in SMART360 platform.
# Using SMART360 API endpoint for data ingestion.

def load_data(Transformed_Data):
    api_endpoint = "https://smart360api.com/consumer/data"
    headers = {'Authorization': 'Bearer YOUR_ACCESS_TOKEN', 'Content-Type': 'application/json'}

    for consumer in Transformed_Data:
        response = requests.post(api_endpoint, json=consumer, headers=headers)

        if response.status_code == 200:
            print(f"Data loaded successfully for Consumer ID: {consumer['Consumer ID']}")
        else:
            print(f"Failed to load data for Consumer ID: {consumer['Consumer ID']}")
```



```python
def main():

    # Extaraction of data from MySQL Database
    consumer_data = Consumer_data_extract(hostname="", username="", Password="", database_name="")

    # Map Data to SMART360 format.
    map_data = Map_Data(Consumer_Data)

    # Transform data into SMART360 Format
    transformed_data = Transform_Data(Consumer_Data)

    # Loading the data into SMART360
    load_data(Transformed_Data)


########################################################################################################################

if __name__ == "__main__":
    main()

schedule.every().day.at("08:00").do(main)

while True:
    schedule.run_pending()
    time.sleep(1)
```

It will run every day at 08:00 am

Conclusion:

The mapping process is a critical step in integrating consumer data into the SMART360 platform. By following a systematic approach to extract, map, and transform the data, we ensure that it aligns with the structure and requirements of the platform. Documenting the mapping process helps maintain transparency and clarity, facilitating future updates and modifications as needed.