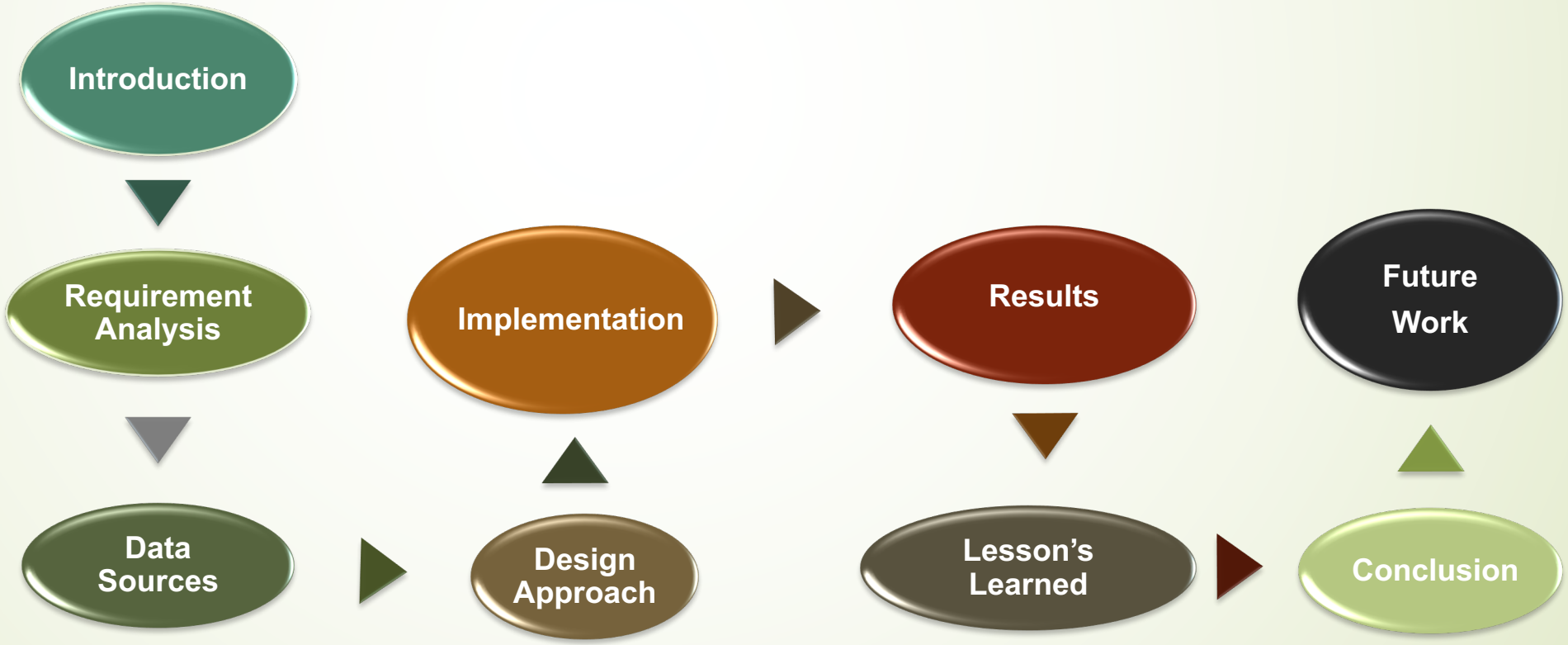
Invoice Matching Application

- Suresh CHINNA SHANMUGAM

Agenda

- Project Flow – SDLC Life Cycle => Requirements to deployment & Conclusion.
- Functional Flow - Load Input Data-> Invoice Processing -> Matching/Unmatching -> Output Rendering using Python API



Introduction

- Invoice Processing

Motivation

- The key focus area is to implementation of Invoice Matching(Processing) Application using Python

Context

- The Process logic, Design Approach, Implementation and test execution, deployment and results validation are key activities to meet the functional requirements.

Problem Statement

- Required to find the list of Invoices/payments and that are matched or Balanced between Debit and Credit amounts, and list the remaining Unmatched records.

Contribution

- Implementation of business process logic using python, database and Flask API

Requirement Analysis

Requirement Analysis

- High level Requirements – To create Matched Invoices with the payment (Types) data based on Debit and credit amounts – The Balanced amounts.
- Technological Specification: Pure Python, API Mode, One backend Table – Hence Opted for MySQL DB to store the required data into Database.
- Meta Data/ Schema for the tables

The Key Requirements from User(Operado)

- Create an instance to feed the table. The video does not show the addition of Invoice, but it will be necessary to envisage it for the continuation
- Filter results based on instance data
- The possibility of associating instances of payment and invoice type when the amount is matching
  - ex : Invoice in the amount of € 20 and a payment of € 2

Assumptions:

- NO Specific Data format given, Hence Created CSV file format as input from the Jpeg Sample Data provided by Operado for the current project implementation.
- No Technological Specification to use on Python API – Hence Opted for Python Flask to display the output data in the HTML page from the processed Json Files for all the matched and Invoices invoices
- No Specific requirement on Unmatched records, hence added additional Enhancements

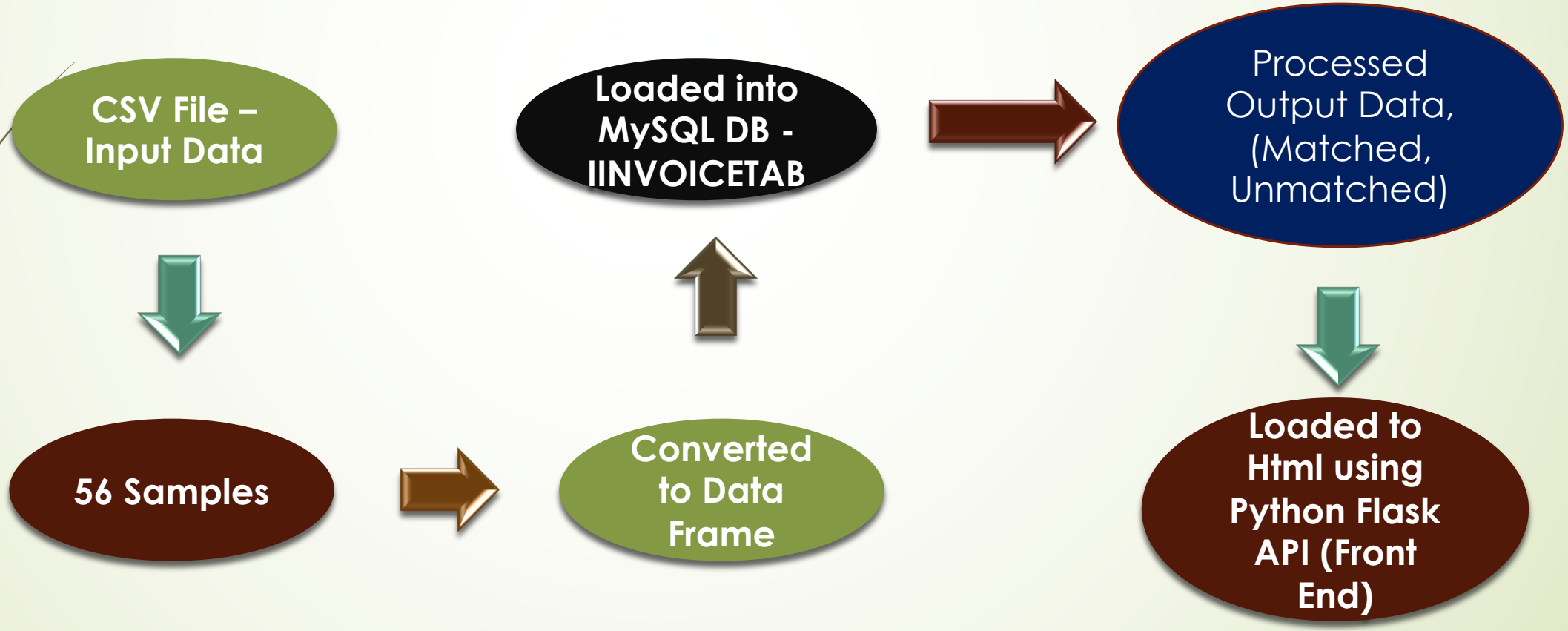
Additional Enhancements:

– Additional Implementation to display All Unmatched Invoices order by Recover (Due, Not Due and Blank)

- Implementation on to display the summary of Unmatched records based on Recovery (Debit and Credit Summary)

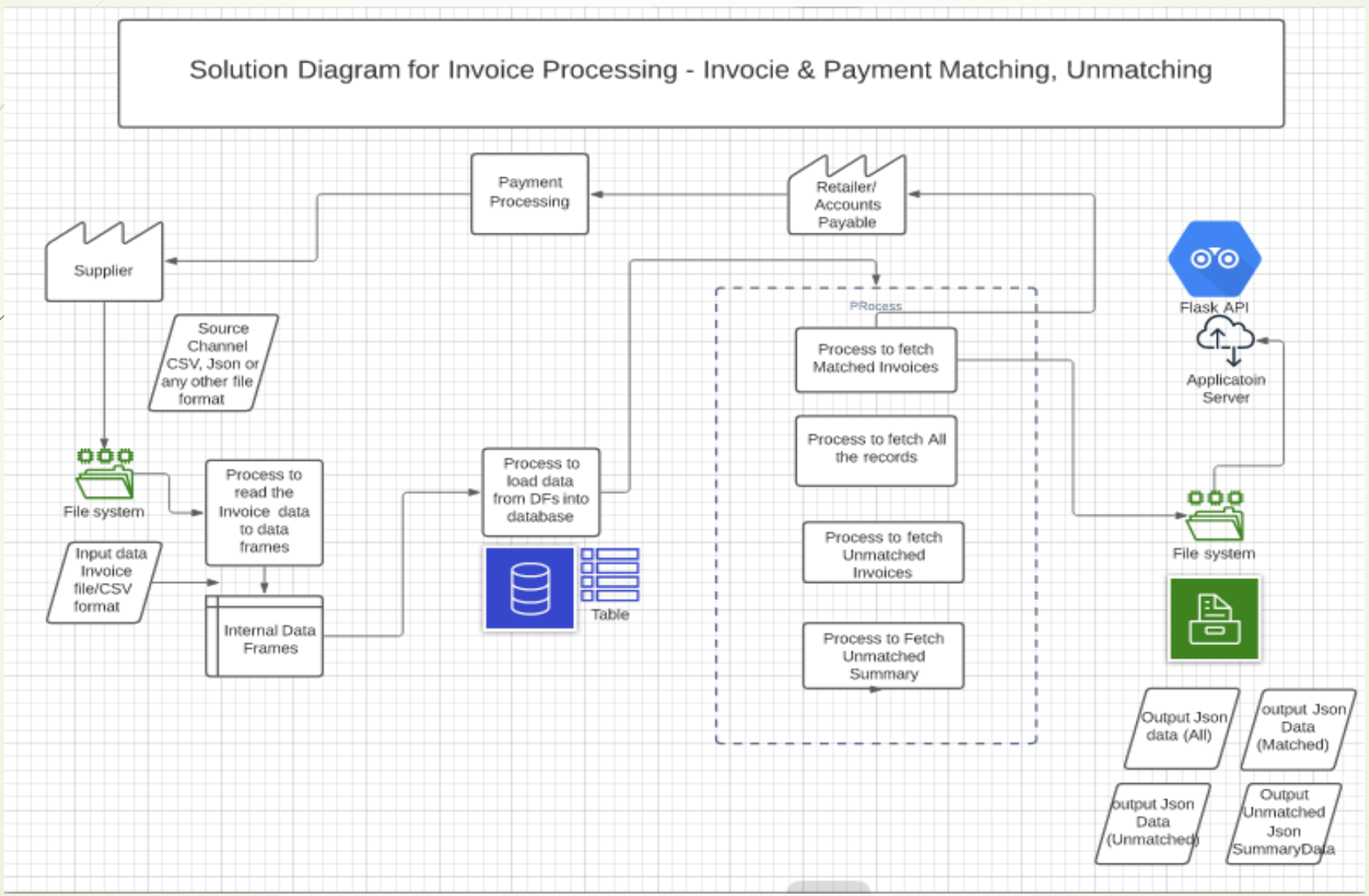
Data Source – Input and Output

- Data Source – From the Jpeg Image provided by Oparedo used as Input to Create CSV file format and created output matched and unmatched data into Json Files and loaded in HTML page using Python Flask API methods.



Design Approach

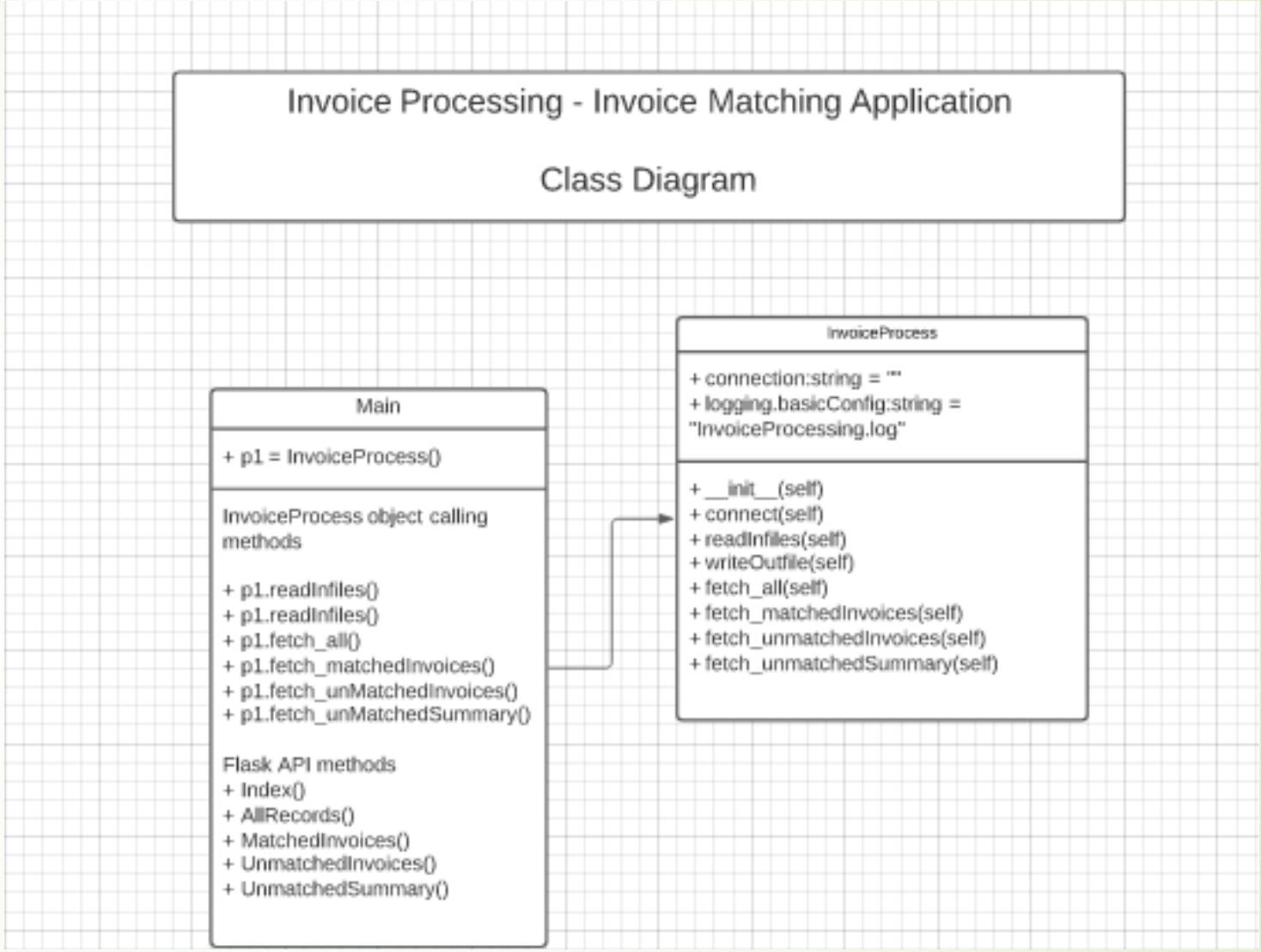
Solution Architecture Diagram – High level



Implementation

Implementation – Low Level Design

- **Git Hub Link - https://github.com/sureshcs0524/PythonProjects**
- 2 python scripts – main.py, InvoiceProcess.py
- Packages used:
  - pandas, pymysql, numpy, shutil, os, json, logging, Flask, jsonify, render_template, request, redirect
- Database: mysql
  - Database Name: InvoiceDB
  - Table name: INVOICETAB
- 1 Key class – InvoiceProcess with the methods. List of members and methods
- templates folder used for flask API to load json data into Html page
  - content.html – Home Page
  - Index.html – Index Page
  - AllRecords.html
  - MatchedInvoicesPayment.html
  - UnmatchedInvoices.Payment.html
  - UnmatchedSummary.html



Code & Data Paths – Input/output

- Project Path:
  - ~/PycharmProjects/InvoiceProcessing

  Files: main.py

  InvoiceProcess.py

  Templates:

  Html Files listed in previous page

  Log File:

  InvoiceProcessing.log
- Input:
  - ~/PycharmProjects/input
  - File: CSV File: InvoiceTestData.csv
- Output:
  - Path: ~/PycharmProjects/output
  - Temp Files: Json Files:
    - AllRecords.json
    - MatchedRecords.json
    - UnMatchedInvcRecords.json
    - UnMatchedSumRecords.json

```
(venv) (base) Sureshs-MacBook-Pro:PycharmProjects sureshcs$ pwd
/Users/sureshcs/PycharmProjects
```

```
(base) Sureshs-MacBook-Pro:InvoiceProcessing sureshcs$ pwd
/Users/sureshcs/PycharmProjects/InvoiceProcessing
```

```
(base) Sureshs-MacBook-Pro:InvoiceProcessing sureshcs$ ls -ltr
total 56
drwxr-xr-x  6 sureshcs  staff    192 May 29 14:42 venv
drwxr-xr-x  8 sureshcs  staff    256 May 30 23:50 templates
-rw-r--r--  1 sureshcs  staff  12710 May 31 17:00 InvoiceProcess.py
-rw-r--r--  1 sureshcs  staff   3650 May 31 17:00 main.py
drwxr-xr-x  3 sureshcs  staff     96 May 31 17:00 __pycache__
-rw-r--r--  1 sureshcs  staff   4214 May 31 18:03 InvoiceProcessing.log
```

```
(base) Sureshs-MacBook-Pro:templates sureshcs$ ls -atlr
total 48
-rw-r--r--  1 sureshcs  staff  397 May 30 23:49 AllRecords.html
-rw-r--r--  1 sureshcs  staff  446 May 30 23:49 MatchedInvoicesPayment.html
-rw-r--r--  1 sureshcs  staff  465 May 30 23:49 UnmatchedInvoices.html
-rw-r--r--  1 sureshcs  staff  457 May 30 23:49 UnmatchedSummary.html
-rw-r--r--  1 sureshcs  staff  359 May 30 23:49 content.html
-rw-r--r--  1 sureshcs  staff  818 May 30 23:49 index.html
```

```
(venv) (base) Sureshs-MacBook-Pro:output sureshcs$ pwd
/Users/sureshcs/PycharmProjects/output
```

```
(venv) (base) Sureshs-MacBook-Pro:output sureshcs$ ls -atlr
total 48
drwxr-xr-x  14 sureshcs  staff   448 May 31 15:59 ..
-rw-r--r--   1 sureshcs  staff  9341 May 31 17:00 AllRecords.json
-rw-r--r--   1 sureshcs  staff   257 May 31 17:00 MatchedRecords.json
-rw-r--r--   1 sureshcs  staff  1490 May 31 17:00 UnMatchedInvcRecords.json
-rw-r--r--   1 sureshcs  staff   117 May 31 17:00 UnMatchedSumRecords.json
```

Database

- Database Details
  - Database: MySQL DB
  - Python package: pymysql
  - Database name: InvoiceDB
  - Table name: INVOICETAB
  - Data Schema
    - Shown in the screen shot

```
mysql> desc invoicetab;
+--------------+----------------+------+-----+---------+----------------+
| Field        | Type           | Null | Key | Default | Extra          |
+--------------+----------------+------+-----+---------+----------------+
| ID           | int(11)        | NO   | PRI | NULL    | auto_increment |
| INVOICE_NO   | varchar(20)    | YES  |     | NULL    |                |
| DATE         | varchar(10)    | YES  |     | NULL    |                |
| TYPE         | varchar(50)    | YES  |     | NULL    |                |
| CONTRACT_NO  | varchar(20)    | YES  |     | NULL    |                |
| DEBIT        | decimal(10,2)  | YES  |     | NULL    |                |
| CREDIT       | decimal(10,2)  | YES  |     | NULL    |                |
| DUE_DATE     | varchar(10)    | YES  |     | NULL    |                |
| DOCUMENT     | blob           | YES  |     | NULL    |                |
| LETTERING    | varchar(50)    | YES  |     | NULL    |                |
| RECOVERY     | varchar(50)    | YES  |     | NULL    |                |
+--------------+----------------+------+-----+---------+----------------+
11 rows in set (0.02 sec)
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| InvoiceDB          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use InvoiceDB
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------------+
| Tables_in_invoicedb  |
+----------------------+
| email                |
| invoicetab           |
| users                |
+----------------------+
3 rows in set (0.01 sec)
```

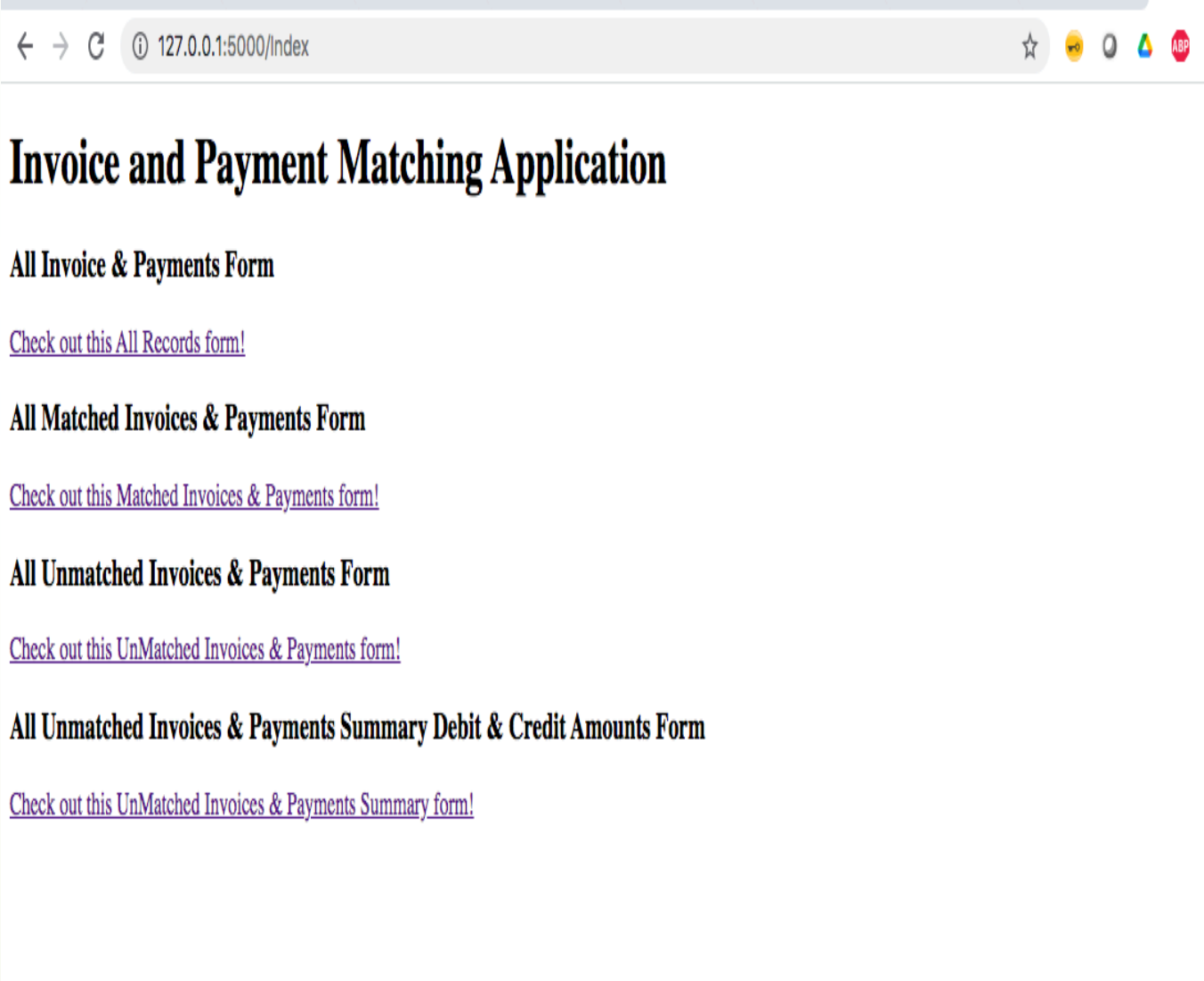
```
mysql> select * from invoicetab;
+------+------------+------------+-----------------------+-------------+---------+---------+------------+----------+-----------+---------+
| ID   | INVOICE_NO | DATE       | TYPE                  | CONTRACT_NO | DEBIT   | CREDIT  | DUE_DATE   | DOCUMENT | LETTERING | RECOVERY |
+------+------------+------------+-----------------------+-------------+---------+---------+------------+----------+-----------+---------+
| 6248 | nan        | 2020-02-27 | Miscellaneous operation |           |    0.00 |   16.00 | nan        | nan      | 20.AAAAV  | nan      |
| 6249 | nan        | 2020-02-27 | Payment               |             |    0.00 |  200.00 | nan        | nan      | 20.AAAAV  | nan      |
| 6250 | nan        | 2020-02-27 | Payment               |             |    0.00 |  200.00 | nan        | nan      | nan       | nan      |
| 6251 | nan        | 2020-02-27 | Payment               |             |    0.00 |  924.00 | nan        | nan      | nan       | nan      |
| 6252 | nan        | 2020-02-18 | Pay                   | 000001      |    0.10 |    0.00 | nan        | nan      | nan       | nan      |
| 6253 | nan        | 2020-02-18 | Miscellaneous operation | 000001    |    0.10 |    0.00 | nan        | nan      | nan       | nan      |
| 6254 | nan        | 2020-02-18 | Payment               | 000001      |    0.00 |    0.10 | nan        | nan      | nan       | nan      |
| 6255 | nan        | 2020-02-18 | Miscellaneous operation | 000001    |    2.00 |    0.00 | nan        | nan      | 20.AAAAU  | nan      |
| 6256 | nan        | 2020-02-18 | Payment               | 000001      |    0.00 |  170.00 | nan        | nan      | 20.AAAAU  | nan      |
| 6257 | nan        | 2020-02-18 | Payment               |             |    0.00 |  168.00 | nan        | nan      | nan       | nan      |
| 6258 | nan        | 2020-02-18 | Miscellaneous operation |           | 1292.00 |    0.00 | nan        | nan      | 20.AAAAT  | nan      |
| 6259 | nan        | 2020-02-18 | Miscellaneous operation |           |    0.00 |  620.00 | nan        | nan      | 20.AAAAT  | nan      |
| 6260 | nan        | 2020-02-18 | Payment               | 000001      |    0.00 | 2000.00 | nan        | nan      | 20.AAAAT  | nan      |
| 6261 | nan        | 2020-02-17 | Payment               | 000001      |    0.00 | 1000.00 | nan        | nan      | 20.AAAAT  | nan      |
| 6262 | 2020_00029 | 2020-02-05 | Invoice               | 000001      |   84.00 |    0.00 | 2020-04-05 | nan      | 20.AAAAT  | Balance  |
| 6263 | 2020_00028 | 2020-02-05 | Invoice               |             |  924.00 |    0.00 | 2020-03-06 | nan      | 20.AAAAT  | Balance  |
| 6264 | 2020_00027 | 2020-02-04 | Invoice               |             |  228.00 |    0.00 | 2020-03-05 | nan      | 20.AAAAT  | Balance  |
| 6265 | 2020_00026 | 2020-02-04 | Invoice               |             | 1008.00 |    0.00 | 2020-03-05 | nan      | 20.AAAAT  | Balance  |
| 6266 | 2020_00025 | 2020-02-04 | Invoice               |             |  216.00 |    0.00 | 2020-03-05 | nan      | 20.AAAAV  | Balance  |
| 6267 | 2020_00024 | 2020-02-04 | Invoice               | 000001      |   84.00 |    0.00 | 2020-04-04 | nan      | 20.AAAAT  | Balance  |
| 6268 | 2020_00023 | 2020-02-04 | Invoice               |             |  410.40 |    0.00 | 2020-03-05 | nan      | nan       | Not Due  |
| 6269 | 2020_00022 | 2020-02-04 | Invoice               |             |  252.00 |    0.00 | 2020-03-05 | nan      | nan       | Not Due  |
| 6270 | 2020_00021 | 2020-02-04 | Invoice               | 000001      | 19572.00 |    0.00 | 2020-04-04 | nan      | nan       | Not Due  |
| 6271 | 2020_00020 | 2020-02-04 | Invoice               |             |  708.00 |    0.00 | 2020-03-05 | nan      | nan       | Not Due  |
| 6272 | 2020_00019 | 2020-02-04 | Invoice               |             |  264.00 |    0.00 | 2020-03-05 | nan      | nan       | Not Due  |
| 6273 | 2020_00018 | 2020-02-04 | Invoice               |             |  345.60 |    0.00 | 2020-03-05 | nan      | nan       | Not Due  |
| 6274 | 2020_00017 | 2020-02-04 | Invoice               | 000001      |   84.00 |    0.00 | 2020-04-04 | nan      | nan       | Not Due  |
| 6275 | 2020_00016 | 2020-02-03 | Invoice               |             |  518.40 |    0.00 | 2020-03-04 | nan      | nan       | Not Due  |
| 6276 | 2020_00015 | 2020-02-03 | Invoice               |             |  168.00 |    0.00 | 2020-03-04 | nan      | nan       | Not Due  |
| 6277 | 2020_00014 | 2020-01-31 | Invoice               |             |  840.00 |    0.00 | 2020-03-01 | nan      | nan       | Not Due  |
| 6278 | 2020_00013 | 2020-01-31 | Invoice               | 000001      |   84.00 |    0.00 | 2020-03-31 | nan      | nan       | Not Due  |
| 6279 | 2020_00012 | 2020-01-31 | Invoice               | 000001      |  336.00 |    0.00 | 2020-03-31 | nan      | nan       | Not Due  |
| 6280 | 2020_00011 | 2020-01-31 | Invoice               | 000001      |  168.00 |    0.00 | 2020-03-31 | nan      | nan       | Not Due  |
| 6281 | 2020_00010 | 2020-01-31 | Invoice               | 000001      |   84.00 |    0.00 | 2020-03-31 | nan      | 20.AAAAU  | Balance  |
| 6282 | 2020_00008 | 2020-01-28 | Invoice               | 000001      |   84.00 |    0.00 | 2020-03-28 | nan      | 20.AAAAU  | Balance  |
| 6283 | 2020_00007 | 2020-01-22 | Invoice               | 000001      |   84.00 |    0.00 | 2020-03-22 | nan      | nan       | Not Due  |
| 6284 | 2020_00006 | 2020-01-22 | Invoice               | 000001      |  252.00 |    0.00 | 2020-03-22 | nan      | nan       | Not Due  |
```

Results

Index Page – using python API

- Results Displayed in Front End – Index Page for Invoice and Payment Matching Application: Rendered using Python Flask API - with POST, GET methods.
- URL:

  Index Page: http://127.0.0.1:5000/Index
- Index Page Includes List of Operations
  - All Invoices and Payment Form
  - All Matched Invoices & Payments Form
  - All Unmatched Invoices or Payments Form
  - All Unmatched Invoices & Payments Summary – Debit & Credit Amounts Form



All Records Page - using python API

- All Records Page – List of all the data records loaded to database from the Input CSV data – Rendered using Python Flask API - with POST, GET methods.
- URL: All records Page:
  - http://127.0.0.1:5000/All
- All Records - List of Operations
  - List of All the records from the database table – INVOICETAB
  - The data pulled from INVOICETAB, converted to Data frame -> Json file -> then rendered the json file to html

127.0.0.1:5000/All

Invoice and Payment Records - All

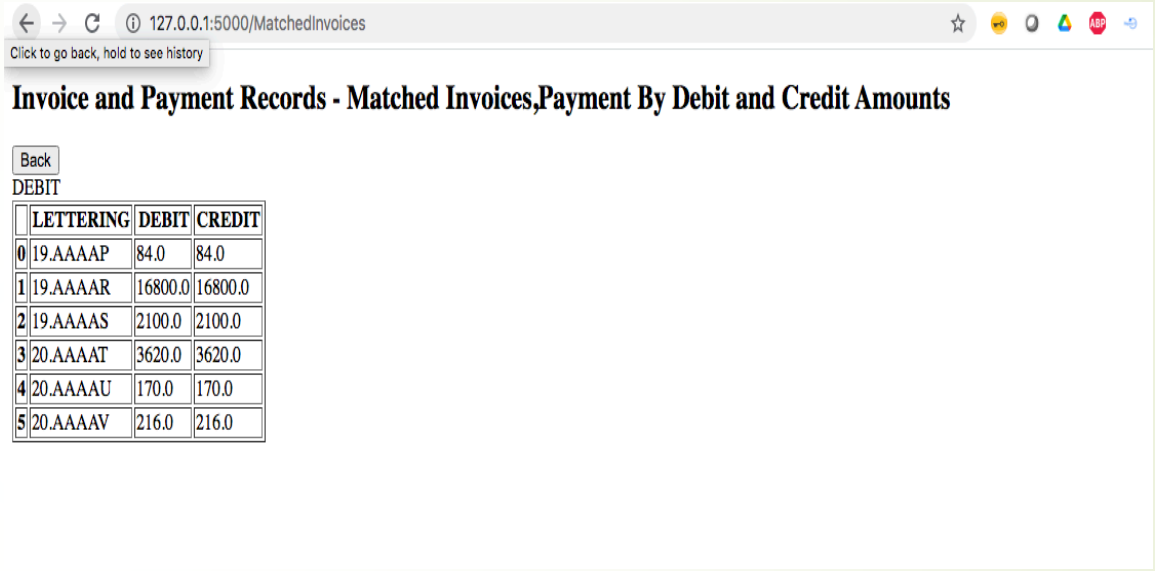
Back

INVOICE_NO

	ID	INVOICE_NO	DATE	TYPE	CONTRACT_NO	DEBIT	CREDIT	DUE_DATE	DOCUMENT	LETTERING	RECOVERY
0	6192	nan	2020-02-27	Miscellaneous operation		0.0	16.0	nan	nan	20.AAAAV	nan
1	6193	nan	2020-02-27	Payment		0.0	200.0	nan	nan	20.AAAAV	nan
2	6194	nan	2020-02-27	Payment		0.0	200.0	nan	nan	nan	nan
3	6195	nan	2020-02-27	Payment		0.0	924.0	nan	nan	nan	nan
4	6196	nan	2020-02-18	Pay	000001	0.1	0.0	nan	nan	nan	nan
5	6197	nan	2020-02-18	Miscellaneous operation	000001	0.1	0.0	nan	nan	nan	nan
6	6198	nan	2020-02-18	Payment	000001	0.0	0.1	nan	nan	nan	nan
7	6199	nan	2020-02-18	Miscellaneous operation	000001	2.0	0.0	nan	nan	20.AAAAU	nan
8	6200	nan	2020-02-18	Payment	000001	0.0	170.0	nan	nan	20.AAAAU	nan
9	6201	nan	2020-02-18	Payment		0.0	168.0	nan	nan	nan	nan
10	6202	nan	2020-02-18	Miscellaneous operation		1292.0	0.0	nan	nan	20.AAAAT	nan
11	6203	nan	2020-02-18	Miscellaneous operation		0.0	620.0	nan	nan	20.AAAAT	nan
12	6204	nan	2020-02-18	Payment	000001	0.0	2000.0	nan	nan	20.AAAAT	nan
13	6205	nan	2020-02-17	Payment	000001	0.0	1000.0	nan	nan	20.AAAAT	nan
14	6206	2020_00029	2020-02-05	Invoice	000001	84.0	0.0	2020-04-05	nan	20.AAAAT	Balance
15	6207	2020_00028	2020-02-05	Invoice		924.0	0.0	2020-03-06	nan	20.AAAAT	Balance
16	6208	2020_00027	2020-02-04	Invoice		228.0	0.0	2020-03-05	nan	20.AAAAT	Balance
17	6209	2020_00026	2020-02-04	Invoice		1008.0	0.0	2020-03-05	nan	20.AAAAT	Balance
18	6210	2020_00025	2020-02-04	Invoice		216.0	0.0	2020-03-05	nan	20.AAAAV	Balance
19	6211	2020_00024	2020-02-04	Invoice	000001	84.0	0.0	2020-04-04	nan	20.AAAAT	Balance
20	6212	2020_00023	2020-02-04	Invoice		410.4	0.0	2020-03-05	nan	nan	Not Due

Matched Invoices Page - using python API

- Matched Invoices Page – List of all the Matched Invoices associated based on LETTERING Ids– Rendered using Python Flask API - with POST, GET methods to display in html page.
- URL: Matched Invoices Page:
  - http://127.0.0.1:5000/MatchedInvoices
- Matched Invoices - List of Operations
  - List of All matched Invoices and Payments by Debit and Credit Amounts associated by LETTERING IDs processed from the data retrieved from the database table – INVOICETAB
  - The matched Invoice data pulled from INVOICETAB, converted to Data frame -> Json file -> then rendered the json file to html



	LETTERING	DEBIT	CREDIT
0	19.AAAAP	84.0	84.0
1	19.AAAAR	16800.0	16800.0
2	19.AAAAS	2100.0	2100.0
3	20.AAAAT	3620.0	3620.0
4	20.AAAAU	170.0	170.0
5	20.AAAAV	216.0	216.0

Unmatched Invoices Page – using python API

- Unmatched Invoices Page – List of all the Unmatched Invoices order by RECOVERY – Rendered using Python Flask API - with POST, GET methods to display in html page.
- URL: Unmatched Invoices Page:
  - http://127.0.0.1:5000/UnmatchedInvoices
- Unmatched Invoices- List of Operations
  - List of All Unmatched Invoices and Payments by Debit and Credit Amounts group by Invoice_no order by recovery (Due – Echu , Not Due – Non échu, none/blank), for the data retrieved from the database table – INVOICETAB
  - The data pulled from INVOICETAB, processed using unmatched invoices business logic, converted to Data frame -> Json file -> then rendered the json file to html

127.0.0.1:5000/UnmatchedInvoices

Invoice and Payment Records - Unmatched Invoices/Payment of DEBIT & Credit Amounts - Order by Recovery

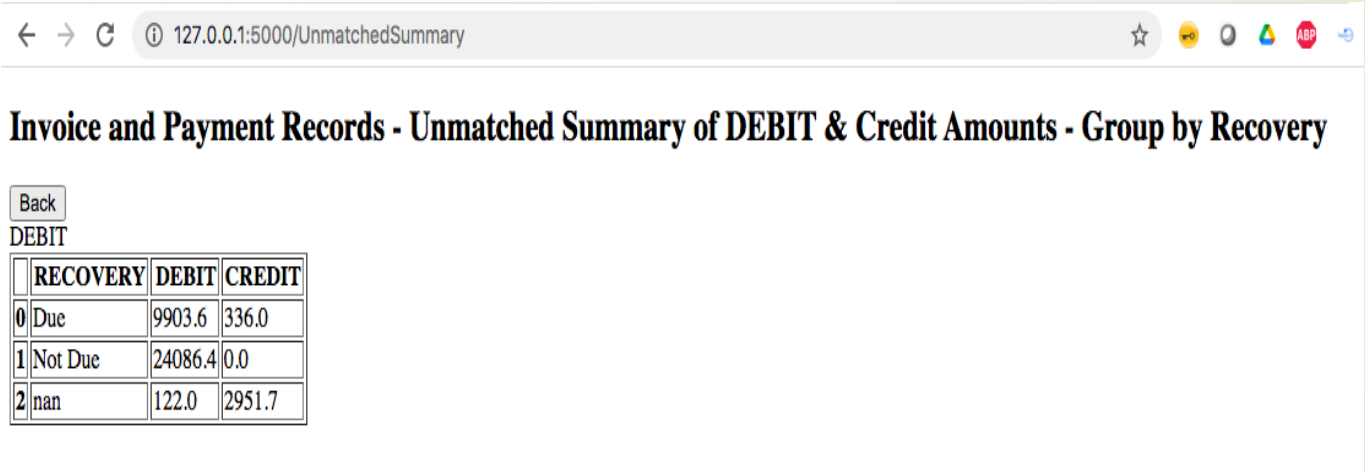
Back

RECOVERY

	INVOICE_NO	RECOVERY	DEBIT	CREDIT
0	2019_00046	Due	0.0	336.0
1	2019_00047	Due	504.0	0.0
2	2019_00082	Due	8400.0	0.0
3	2019_00084	Due	336.0	0.0
4	2019_00085	Due	84.0	0.0
5	2019_00086	Due	84.0	0.0
6	2019_00088	Due	37.8	0.0
7	2019_00089	Due	37.8	0.0
8	2019_00141	Due	420.0	0.0
9	2020_00006	Not Due	252.0	0.0
10	2020_00007	Not Due	84.0	0.0
11	2020_00011	Not Due	168.0	0.0
12	2020_00012	Not Due	336.0	0.0
13	2020_00013	Not Due	84.0	0.0
14	2020_00014	Not Due	840.0	0.0
15	2020_00015	Not Due	168.0	0.0
16	2020_00016	Not Due	518.4	0.0
17	2020_00017	Not Due	84.0	0.0
18	2020_00018	Not Due	345.6	0.0

Unmatched Summary – using python API

- Unmatched summary Page – List of all the Unmatched summary of records group by RECOVERY – Rendered using Python Flask API - with POST, GET methods to display in html page.
- URL: Unmatched Invoices Page:
  - http://127.0.0.1:5000/UnmatchedSummary
- Unmatched Summary - List of Operations
  - List of All Unmatched summary of Debit and Credit Amounts group by by recovery (Due – Echu , Not Due – Non échu, none/blank) for the data retrieved from the database table – INVOICETAB
  - The data pulled from INVOICETAB, processed using unmatched summary business logic, converted to Data frame -> Json file -> then rendered the json file to html



	RECOVERY	DEBIT	CREDIT
0	Due	9903.6	336.0
1	Not Due	24086.4	0.0
2	nan	122.0	2951.7

Lesson's Learned

Requirement Analysis

- Clearly addressed requirement gaps. – In/out of scope.
- Additional enhancement requirement questions – Data Security, User authentications, Functionality etc

Development Best practices

- Enhanced Standard Implementation practices with the efficient usage of python packages.
- Thorough research and study with the usage of efficient process logic in the implementation to transfer the json data directly into python flask API to get and post the messages to the front-end app.
- Deployment of code in the application server or cloud.

Future Enhancements work:

- Understand any Business process enhancement requirements from the users.

Future Scope

Within Scope

- Matched Invoices
  - Invoice: Debit type transaction
    Payment: Credit type transaction
    Transaction association (lettering) is only done between balanced credit and debit type transactions
    Example:
    Ex:
    - 1 invoice of 10€ and 1 payment of 10€.
    - 2 invoices of 10€ and 1 payment of 20€.
    - 2 invoices of 10€ and 2 payments of 10€.
    - 1 invoice of 10€ and 2 payments of 5€ each

Out of Scope:

- No 30way matching between Order, Invoice, and Payments - Requirement to match only between DEBIT and credit amounts.
- No Invoice detail(line) level matching (SKUs, Quantities etc)

Additional Enhancements Included:

- Unmatched Invoices/payments of Debit & Credit Amounts – Order by Recovery ((Due. - Echu, Not Due – Non-échu, Blank)
- Unmatched Summary of DEBIT & Credit Amounts – Group by Recovery. (Due. - Echu, Not Due – Non-échu, Blank)

Further Enhancements Scope:

This project can be enhanced for future requirements

- This can be enhanced for future requirements like - 3 or 4-Way Matching - If additional details are provided (Order Information, Shipment, Invoice and Payments)
- Matched and Unmatched Detail level Matching (SKUs, and Quantities)
- Multiple Format Input files & sources
- Data Security/User authentication requirements
- Payment processing – Accounts Payable process etc

“

Thank you

”

- Suresh CHINNA SHANMUGAM
- Email: suresh.cs0524@gmail.com
- Mobile: +33 758072207