**Explanation Document:**

**Problem Statement:**

Provided CSV (<https://opendata.fcc.gov/api/views/cw62-ya68/rows.csv?accessType=DOWNLOAD&sorting=true>) is data of customer feedback and complaints for a US based telecom firm. Given these data you need to come up with following:

1. High level API flow architecture diagram for your proposed solution.
2. Python Code to:
   1. Set up an in-memory database of your choice – and load the given CSV data into the in-memory DB
   2. REST APIs communicating using JSON to:

                                                               i.      Filter and fetch records provided one or all fields. At least one field should be compulsory (rest optional), and the API provides only relevant records back based on fields provided in JSON format.

                                                             ii.      Bulk Insert records into in-memory database. Add data validation based on your assumptions (at least 2 validation compulsory)

1. Clearly explain the assumptions you have taken to develop this architecture and code base.
2. Provide screenshots of API testing using app of your choice covering as many scenarios as possible. Limit the screenshots only to 10 png/jpg images.
3. Explain how would you make this architecture of yours scalable, reusable and manageable to incorporate more such APIs within your ecosystem.
4. API DOCUMENTATION IS SHARED BELOW.
5. PYTHON Code :
   1. Database\_operations – to set ,create collections, to add, retrieve data from database
   2. Bulk\_insert\_trigger – inserting bulk data through csv link
   3. Data\_management – API to get data, to insert data one by one, or to insert csv link

Why link is inserted and processing done non blocking way because if data is too large then processing and data validation may take significant amount of time that’s why Bulk\_insert\_trigger timer trigger is created which run every 20 seconds which process the csv data and insert into data base.

**Assumptions: azure cloud as api host , GIT as versioning tool for code, Mongodb (nosql) database**

**Scalability :**

1. **This approach is much scalable since timer trigger independent of use action and response is non blocking.**
2. **We can add as many apis as possible in a azure function app.**
3. **Also Csv files can be saved in azure storage account as Binary Large object file (blob)**

**DATA Validations :**

1) objectId – must not get into data base (since db hase own way of creating it automatically)

2) TicketId – since ticke id is planed to be primary key every documenbt must have ticket id other wise that document is discarded.

3) Location data validations – latitude ,latitude must be in valid given range.otherwise data is stored with corresponding error message.

**API TESTING :**

About api testing there are many way of doing that.

We can use pytest in python

1. Time complexity- time required for operations must be low.
2. Memory consumption – should not be any memory issue on large values
3. Scalability – large data also should be accepted
4. Basic API testing based on status codes – if status code 200 then it is passed.
5. Response data validations – key ,values of output data can be checked randomly (since keys are fixed and values ranges and types are fixed)

**API DOCUMENTATION ;**

**I)Api name : data\_management**

**Type : HTTP trigger**

**Methods supported : GET, POST**

**1)GET :**

**Params : feedback=true**

**Response : all the data present in data base**

**2) POST :**

**i)Params : feedback= true,**

**get\_results=true**

**sample\_request payload= {“filters”: {'City': 'Jacksonville'},”projections”:** **{“\_id”:0,'State': 1}}**

**Response : data based on filters and projections in database**

**ii)params : feedback =true**

**bulk\_insert=true**

**csv\_link=<** **csv\_link>}**

**response= <intimation about bulk insert will get started shortly>**

**iii) params : feedback =true**

**bulk\_insert=true**

**sample\_request payload= <data to be inserted>**

**(if data to be inserted is less then can be inserted by this api)**

**iv)** **params : feedback =true**

**insert\_one =true**

**sample\_request payload= <data to be inserted>**

**(if data to be inserted is one row then can be inserted by this api)**

**II)Api name :** bulk\_insert\_trigger

Type : timer tigger

Runs every 20 seconds (can be configured)

Fetches data from csv file and after data validations inserts into database non blocking way to user.