**Assignment :***Create an API using fastAPI framework in Python(*<https://fastapi.tiangolo.com/>)*which will  decide if a customer would be approved for a loan based on below mentioned logic*

**1.**       **Customers who pass through the following condition would be Approved, else Rejected**

a.       Age between 21 to 55

b.       Income GT 20000

c.       bureauScore GE 600

d.       applicationScore GE 600

e.       maxDelL12M <=30

f.        Current Address should match with bureau address with score GT 80% ( you have to write an address matching algorithm)

**2.**       **Line assignment logic:**

a.       allowedFoir : 60%

b.       calculate headroom as headroom = Income \* allowed FOIR – existingEMI

        (e.g, Income = 20000, existingEMI = 4000 and allowedFOIR = 60% then headroom would be 8000 )

c.       loanAmount = headroom \* loanTenure, floor it to nearest hundred i.e. 15540  would be 15500

d.       loanAmount should be between 10K to 5 Lakhs, so loan amount more than 5L would be capped at 5 Lakhs and less than 10K would be rejected

**3.**       **API output would be all Input Variables along with the following**

a.       approvalStatus – approved/Rejected

b.       if rejected – provide rejection reason code (e.g. Age LT 21 or GT 55)

c.       addressMatchingScore

d.       if Approved – loanAmount, loanTenure

**Sample Input JSON**

{

                "customerID" : 100000,

                "dob" : "01/01/1977"      ,

                "income" : 25000              ,

                "bureauScore " : 700       ,

                "applicationScore " : 750,

                "maxDelL12M " : 0,

                "allowedFoir " : 60           ,

                "existingEMI" : 2000        ,

                "loanTenure" : 24             ,

                "currentAddress" : "15 2nd cross vagdevi layout Marathahalli Bangalore Karnataka 560037",

                "bureauAddress" : "HOUSE NO 15 2ND CROSS VASANT LAYOUT MARATHALLI NEAR VAGDEVI SCHOOL BANGALORE 560037 KA"}

* **Estimate Average response time of the API:** Once you create the API, can you please capture time taken by this API and average time by calling for say 2000 cases
* **Estimate break event point of concurrent calls:** Can you also estimate breakeven point of concurrent number of calls which increase API response time by 2 times of Average response time measured earlier?