International Journal of Global Research Innovations & Technology (IJGRIT)
ISSN : 2583-8717, Impact Factor: 6.972, Volume 03, No. 01(II), January-March, 2025, pp 85-90

PYTHON DATA ANALYTICS TO CHECK YEARLY SUBSCRIPTION CEILING LIMIT AND YEARLY CUMULATIVE BALANCE FOR SUBSCRIBER OF GENERAL PROVIDENT FUND

Sumana Chatterjee[*]

ABSTRACT

As per office memorandum vide F. No. 3/13/2022-P& PW(F) (8353) Government of India, Ministry of Personnel, PG & Pensions Department of Pension & Pensioners' Welfare dated 11.10.2022,it was directed for strict compliance in order to maintain the yearly total accumulation of subscriptions in GPF (General Provident Fund) in such a way that the total yearly subscription amount along with all arear subscriptions in a financial year ,should not exceed the limit of rupees five lakh ,at present which is the threshold limit. So, since this time, the necessity of checking the yearly sum of total subscription for each GPF subscriber is urgently needed, before the end of each financial year, so that the accumulation of total subscription may not exceed ceiling limit. The request of subscription change is incorporated in MS-Excel file for each applicant, maintained throughout the whole year ,as per applications received and the csv file mode of which, later-on is analysed by python-AI to check ceiling limit on that platform .This is one part of analysis with the help of python code on the google collaborator platform to sort and adjust the amount of subscription accordingly so that the total amount of subscription in every financial year can be adjusted to satisfy the ceiling limit restriction. Apart from this, i.e. the compliance of ceiling restriction, this paper is also based on another application in MS Excel software to check and compare yearly cumulative balance in General Provident Fund (GPF) after each withdrawal in any financial year for any GPF subscriber when applies for withdrawal from GPF account. Application in MS-Excel, to prepare approval format of claim with the help of MS-Excel formula along with checking and comparison of GPF balance in a financial year using MS-Excel formula as well as using python collaborator platform to check GPF ceiling limit, GPF total withdrawal and total balance for a financial year, all of these applications help to run smoothly, the procedural work for General Provident Fund, with minimum effort to get output with maximum accuracy.

Keywords: *GPF, Python-AI, Google Collaborator, MS-Excel, Subscription, Ceiling.*

Introduction

Since the year, the procedural work of GPF was allotted, the necessity of processing on urgent basis as requested by the subscribers was felt inevitably, due to the reason of their sudden and urgent needs, sometimes for self-treatment or treatment of family members, sometimes for some other urgent need like children education etc. So on account of this purpose, the process of GPF claim was to be made faster with the help of MS-EXCEL format using input value and formula based output. The format of approval order is also made in such a way that, using input in some cells, the printout of the approval order can be taken with the desired output of that particular claim, instead of typing the approval in MS-

[*] Ph.D. Scholar (Computer Science), Nirwan University, Jaipur, Rajasthan, India.

WORD. So in this way using cell formula, the approval can be prepared in lesser time with accurate output. Moreover the record of withdrawal is to be maintained at the time of each withdrawal to maintain the updated balance, which is required to calculate the eligible amount in each financial year for the onward claims of the subscriber .This progressive balance at the time of each withdrawal for each subscriber of GPF is also maintained by Excel formula to maintain the updated balance .Similar as records of withdrawal, records of change of subscription amount is maintained in Excel file also and both of these two files, i.e. files of withdrawal as well as change of subscription, then, these files are subjected to analysis by Python software on google collaborator platform to check ceiling constraint and status of accumulation in GPF fund of each subscriber. Calculation of yearly interest also can be checked using Python code.

Literature Review

The study of calculation of GPF accumulation, relationship between opening balance, subscriptions and recoveries of advance (if any) for all months of a financial year, to establish the formula in EXCEL for subsequent calculation of total withdrawal, sum of deposit and recovery, updated accumulation, balance after each withdrawal in GPF fund for any subscriber. Similarly, to study and establish relationship between cells of MS-EXCEL to get the output of total yearly subscription. Finally with the help of Python and python AI, creation of code to check the updated withdrawal status and updated accumulated fund with or without interest, for any subscriber for the purpose of latest claim on any time in any financial year and also with the help of Python, to check the ceiling constraint for any subscriber.

Research Gap

Initially the processing of claims of subscribers of GPF was used to be done manually, using various registers to track the record of previous claims, monthly subscriptions as well as recoveries for advance withdrawal if applicable there, but no such practice of using software or AI was there .So applying day to day work experience with the pros and cons and also for the purpose of accurate and much faster process of preparation of claim approval including the process of sorting of name with GPF account number, taking the help from software and Python AI as well. This was definitely a new technique which was not used in the past. Without application of MS-Excel software and without programming with Python as well without the help from Python generative AI in google collaborator platform to check the constraint of rule for final withdrawal or advance withdrawal, for the purpose of checking of limitation according as subscriber’s application, the whole process, would not be possible in such lesser time by accurate suitable means.

Research Questions /Hypothesis

In this research, basically wanted to determine whether a subscriber has sufficient accumulation to satisfy the amount of the claim of withdrawal, sometimes 75% and sometimes 90% depending on the purpose of claim and obviously according to the rules. For the purpose of claim related with advance withdrawal, the eligible amount in relation with basic and three fourth of accumulation, whichever is less, the number of months of recovery etc. all these have to be validated as per withdrawal rule for advance. So, this is one part of checking, to check that constraint as applicable for that case. The another part is to check the ceiling constraint and picking up the names of subscribers, sorted month wise by their unique GPF account number so as to adjust the amount of subscription for the rest of that financial year. Sorting of names of subscribers in a financial year has a vital role for checking ceiling constraint as without this technique it is not easily possible to check the name with their unique identity number, i.e. GPF account number to verify ceiling limit of all subscribers one by one.

Methods

At the first step ,there was to open two MS-Excel files to keep records (1) file with records of withdrawals of subscribers when they apply for their claims .Here records of any subscriber are entered accurately whenever the subscriber applies for withdrawal .The relevant features are the unique identity number, here which is GPF account number, based on which the conditional formula of ‘sum if’ of Excel is used to calculate ‘financial yearly total withdrawal’ for that particular subscriber. This is a type of relational database, relation based on input in independent primary cell and cell formula for other dependent values. The most relevant feature columns as maintained there are GPF account number, withdrawal type, opening balance for the financial year, recommended withdrawal, subscription for months, recovery for months, balance after current withdrawal, balance till date. Among these, GPF account number, opening balance, recommended withdrawal, subscriptions and recoveries are mandatory fields with independent data and other fields are dependent variables with relational cell values in relation with these primary field values. There are some other fields also which are necessary to maintain the adequate reference, relevant with the subscriber and remarks for withdrawal. Besides keeping these records other fields for this EXCEL file as maintained here are the formula cells to track total number for each type of withdrawal, total number of file correspondences, total number of bills for any month. All these records are maintained sheet wise for every financial year. (2) Another file is to keep records for variation along with subscription for each month in a financial year. When any subscriber applies for change of subscription, the changed subscription starting from that month as well other months subscription also are entered perfectly with the help of excel and cell formula. When a subscriber applies for change of subscription, this record is maintained month wise to sum up the total yearly as well as till date subscription balance to check ceiling constraint whenever necessary at some interval in year. For both the files, Entry in EXCEL files is the first part of application of software. The second part is dealt with Python analysis of the two EXCEL files, uploaded in csv format, executed in google collaborator platform. For ease of analysis there has been created three program files of Python, opened in google collaborator platform, two of them are dealt with the csv format of EXCEL file with record of GPF withdrawal and updated balance, one Python file is dealt with withdrawal records while the second one is dealt with interest calculation. The third Python file is associated with ceiling check. In this first file, importing packages for Python pandas, uploaded the csv file with database for withdrawal records of subscribers, records with real time data, then executed code for data info and columns for analysis. Then gave user input for GPF account number to check the withdrawal status whether if any withdrawal is there or how many withdrawals are there for that particular subscriber, what is the total amount of withdrawal in that financial year, eligible amount for final withdrawal whether the constraint of rule of withdrawal satisfies with the claim amount and if not satisfies the eligibility, then there causes the case to change the recommended amount accordingly. During the time of using codes of Python, in this Python file, support of generative AI was taken but the logic of calculation to get desired output was to be judged and ultimately after several instructions with trial and error, succeeded to convey the actual logic for total algorithm to get exact result by the generative AI. Really it was a very interactive session with generative AI, nice experience also as well. After successful uploading of csv file, the file with GPF withdrawal records, input the unique number i.e. GPF account number to analyse the eligible amount for final withdrawal. As per rule whether the satisfactory percentage is fulfilled or not, that is checked and necessary change of claim amount is to be changed ,as mentioned earlier along with balance till date can also be determined for that subscriber also. Then the eligibility criteria for advance withdrawal of that particular subscriber is verified also, with the input of previous loan amount, outstanding loan amount till date, amount of new advance, basic pay, current month recovery amount, number of months of recovery as desired by the subscriber, to check the eligibility factor for advance withdrawal claim. With the input of desired number of months, if the recovery per month does not come in whole number, then there is the scope of change of input in programming file, for the number of months for recovery, to avoid the chance of being the monthly recovery amount as fractional value for that supposed would be advance withdrawal. So this was the brief description of analysis for the purpose of checking eligibility criteria for withdrawal from fund, whether in one time withdrawal mode or with monthly recovery mode, for both the cases the analysis comes into very much help to take decision to approve the claim amount .Secondly with the analysis of EXCEL file for withdrawal from fund, there is a scope to check the yearly balance with accrued interest ,calculated for each month’s monthly balance .Each month’s monthly balance is the amount with equation of balance equated as -monthly balance=monthly subscription+monthly recovery-monthly withdrawal. The rate of interest for each month is to be given as input along with input of monthly values of subscription, recovery and withdrawal. With the help of generative AI, here also progressive balance after each month and the final balance with accrued interest for the financial year for the particular subscriber is calculated. The initialization of balance before starting of Python loop is done by the opening balance as input .Values for each month subscription, each month recovery (if applicable), each month withdrawal along with each month interest rate are to provide by using Python loop. Value in the column for interest, for each month is based on each month balance multiplied by rate of interest divided by 1200, 12 for all months in year and 100 for percentage value. Summation of all months’ subscription, summation of all months recoveries, summation of all months withdrawal and summation of all months interest value, sum of all these terms with addition of opening balance for that financial year, gives the final balance with accrued interest at the end of financial year. To calculate the whole progressive balance, support of Python AI, here also was taken and after several communications by human brain and instructions for necessary addition, alteration by suitable fresh commands were inevitable to make it understand properly to build the exact logic for algorithm to get the correct output. Judgement of code as created by AI was very necessary in this process by which ultimately it was possible to reach the goal. However, the whole process was a very nice experience to have such interactive session with Python AI in google collaborator platform. So, this was the brief description of analysis in second file by Python in google collaborator platform with the same csv file as used for first analysis, with records of GPF withdrawal. The third analysis is done also with some other file in csv format of that EXCEL file with records of variation of subscription when any subscriber submits application. This csv file is uploaded in content folder of third file in google collaborator platform, analysis is done with Python code there but here no support of generative AI was taken. Date time variable had to be converted into actual date format, deleted duplicate records. Now formation of new columns were introduced one by one with progressive addition of monthly subscription total as required to check for ceiling constraint, to check whether and when ceiling of subscription total for a particular subscriber has reached or crossed. The list of names with unique account number can be sorted out by filtering process, filtering done for each month, one by one. Along with filtering, the adjustment amount of subscription amount is also obtained in the code accordingly to avoid hazard of crossing the yearly limit for subscription total. In this analysis the factor of date of retirement is considered also because GPF subscription deduction is stopped for a subscriber before three months of superannuation .So ceiling constraint is to be checked, considering this factor also and depending on retirement date, these names are to be eliminated from the filtering dataset as the subscription is not to be deducted for last three months and so the ceiling constraint factor would not matter for the last three months in these cases. So, this was the brief description for analysis with csv file with records of variation of subscription.

Significance of the Study

From the work experience, the necessity of application of software was felt urgently to meet up the process of frequent claims and preparation of all necessary proceedings .So somehow ,some adequate means to process in lesser time but also with full accuracy was felt as very much relevant .Moreover updating of records for withdrawal in writing mode was a cause of headache, as otherwise if missed ,then factor of missing records in ledger register would create reason for wrong balance for any subscriber in future at the time of next claims. From this point of view, the interest of developing some automatic calculation of sum of withdrawals, was evoked in mind which leaded to develop the EXCEL files, full of records of all withdrawals of all types of claim, all data entered with care at the time of submission of claim to be credited. In this file as by EXCEL conditional '@sum if' formula, values of withdrawals for any subscriber are automatically added, so the processing for claim submission followed by entry of records of submission with sequential record number helped to avoid the factor of missing record unlike the case for writing mode, in ledger register. Moreover, the scope of having automatic updated amount of total withdrawals as well as updated balance after the latest claim of any subscriber is very much helpful, flawless as calculation of derived fields is done by cell formula of EXCEL. Similarly, the case of another EXCEL file, which is maintained with records of variation of subscription for any subscriber. At the time of keeping writing records for request of variations, simultaneously records are updated as the soft copy, in the EXCEL file where also, cell formula is used for derived parameters to avoid unnecessary entries for subscriptions for all months. The relational database is made in such way that the data for preceding month is carried over for following month unless and until the deviation for any month takes place when the request from the subscriber for variation is received. So, in brief this was the description of framework for two files, which later on, with the help of Python programming are subjected to analysis to get the quick result from execution of code with uploading these EXCEL files in csv format. Filtering of data for a particular subscriber giving input with that particular GPF account, associated with the name and unique details of that subscriber can give valuable insights regarding the eligibility of claim whether and how permissible according as withdrawal rule. Sorting of names with unique account number can also give valuable month wise insights regarding the list with subscriber reaching to ceiling or crossing limit for oncoming months in a financial year.

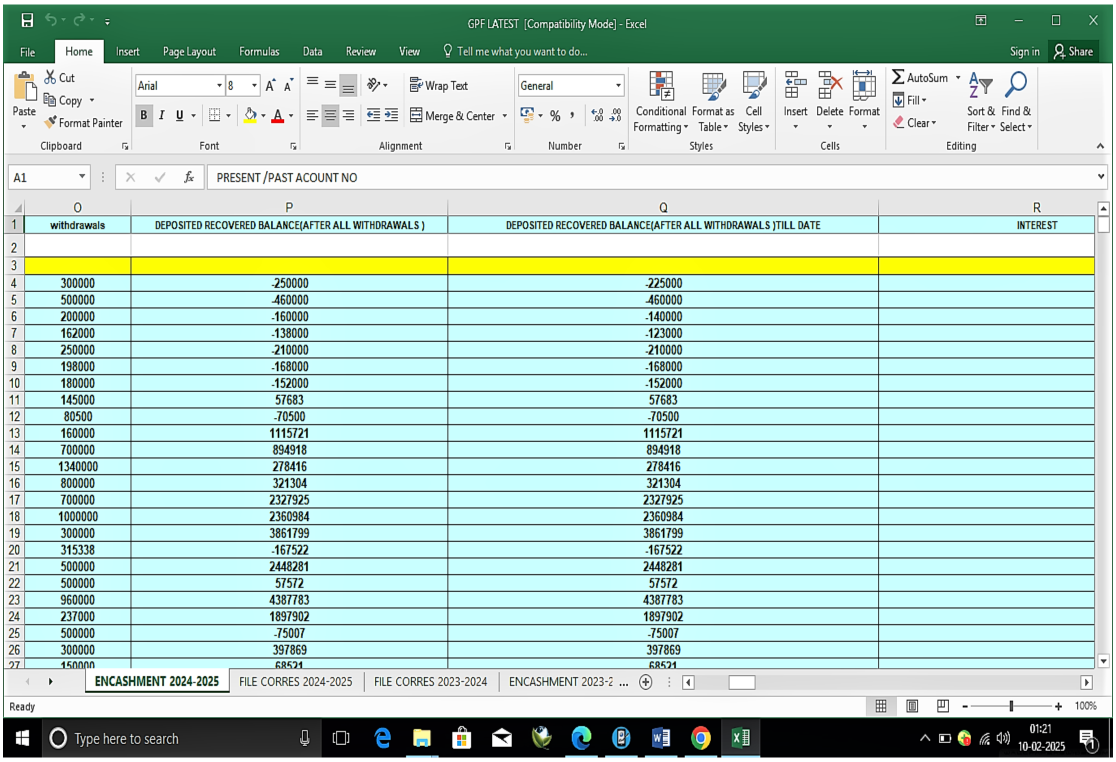
Timeline

The experience for several years, to process the claims of subscribers, when received applications, as well as practice of updating records day to day, throughout all the financial years, made it possible to find pros and cons and accordingly the modifications were done, to make it much more compatible and convenient. From the software, the past records of withdrawal, as well as updated balance along with all necessary information, can be obtained easily, as asked by the applicant also.

Conclusion and Future Work

For the ease of claim processing, application of software is used. In future if necessary to append or alter, that also can be modified according to the requirement in future at that time.

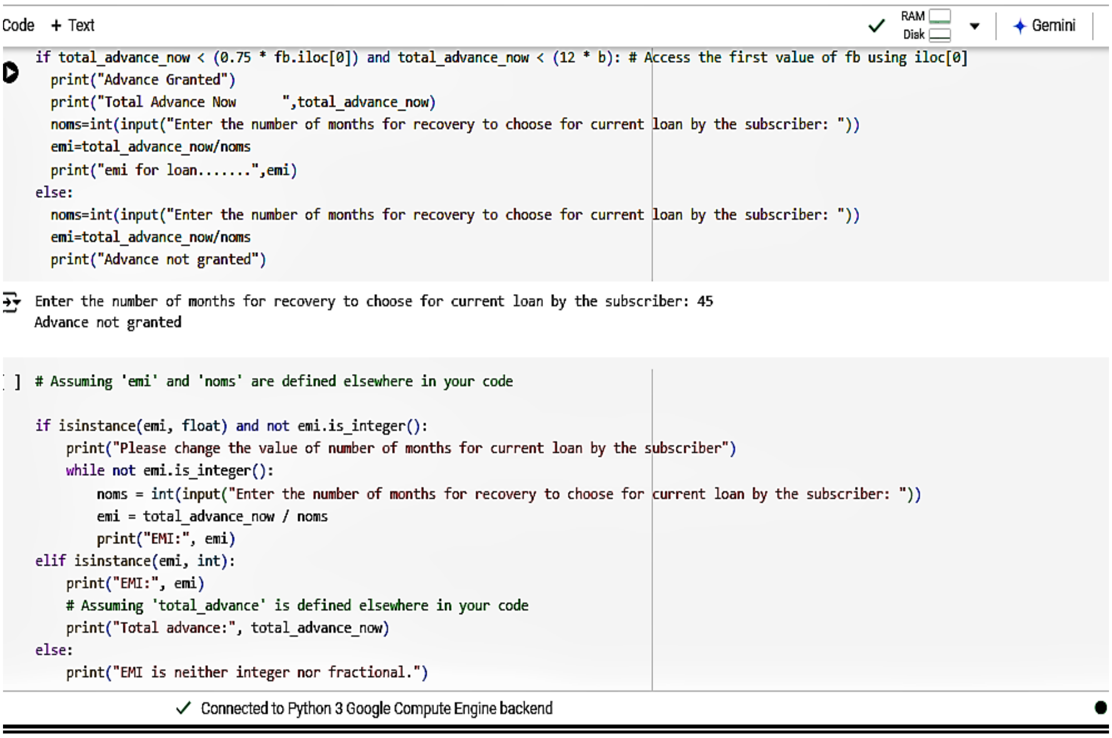
Some Screenshots of Software, as Used



PRESENT /PAST ACOUNT NO

	O	P	Q	R
1	withdrawals	DEPOSITED RECOVERED BALANCE(AFTER ALL WITHDRAWALS)	DEPOSITED RECOVERED BALANCE(AFTER ALL WITHDRAWALS)TILL DATE	INTEREST
2				
3				
4	300000	-250000	-225000	
5	500000	-460000	-460000	
6	200000	-160000	-140000	
7	162000	-138000	-123000	
8	250000	-210000	-210000	
9	198000	-168000	-168000	
10	180000	-152000	-152000	
11	145000	57683	57683	
12	80500	-70500	-70500	
13	160000	1115721	1115721	
14	700000	894918	894918	
15	1340000	278416	278416	
16	800000	321304	321304	
17	700000	2327925	2327925	
18	1000000	2360984	2360984	
19	300000	3861799	3861799	
20	315338	-167522	-167522	
21	500000	2448281	2448281	
22	500000	57572	57572	
23	960000	4387783	4387783	
24	237000	1897902	1897902	
25	500000	-75007	-75007	
26	300000	397869	397869	
27	150000	68521	68521	

Figure 1: Image of Excel file with Records of Withdrawal



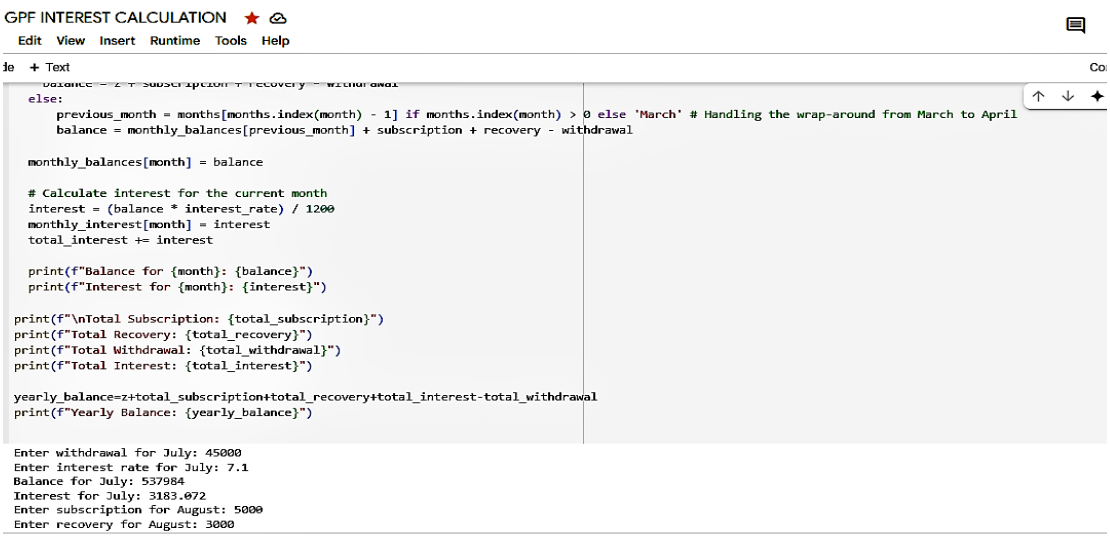
```
if total_advance_now < (0.75 * fb.iloc[0]) and total_advance_now < (12 * b): # Access the first value of fb using iloc[0]
  print("Advance Granted")
  print("Total Advance Now      ",total_advance_now)
  noms=int(input("Enter the number of months for recovery to choose for current loan by the subscriber: "))
  emi=total_advance_now/noms
  print("emi for loan.......",emi)
else:
  noms=int(input("Enter the number of months for recovery to choose for current loan by the subscriber: "))
  emi=total_advance_now/noms
  print("Advance not granted")
```

```
Enter the number of months for recovery to choose for current loan by the subscriber: 45
Advance not granted
```

```
# Assuming 'emi' and 'noms' are defined elsewhere in your code

if isinstance(emi, float) and not emi.is_integer():
    print("Please change the value of number of months for current loan by the subscriber")
    while not emi.is_integer():
        noms = int(input("Enter the number of months for recovery to choose for current loan by the subscriber: "))
        emi = total_advance_now / noms
        print("EMI:", emi)
elif isinstance(emi, int):
    print("EMI:", emi)
    # Assuming 'total_advance' is defined elsewhere in your code
    print("Total advance:", total_advance_now)
else:
    print("EMI is neither integer nor fractional.")
```

✓ Connected to Python 3 Google Compute Engine backend

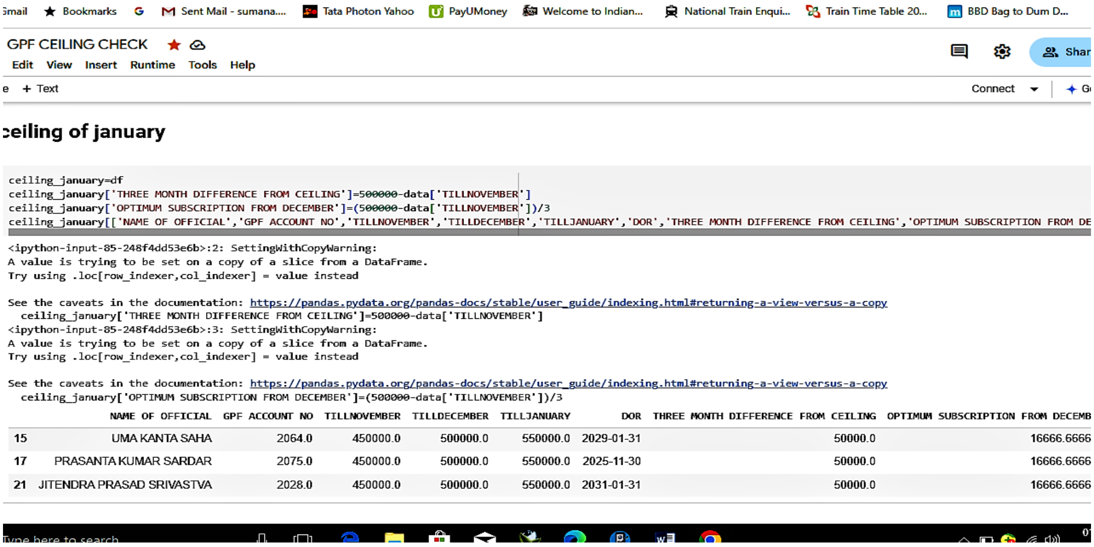
Figure 2: Image of Python File for Checking of Withdrawal and Eligibility Criteria



```
    balance = z + subscription + recovery - withdrawal
  else:
      previous_month = months[months.index(month) - 1] if months.index(month) > 0 else 'March' # Handling the wrap-around from March to April
      balance = monthly_balances[previous_month] + subscription + recovery - withdrawal

  monthly_balances[month] = balance

  # Calculate interest for the current month
  interest = (balance * interest_rate) / 1200
  monthly_interest[month] = interest
  total_interest += interest

  print(f"Balance for {month}: {balance}")
  print(f"Interest for {month}: {interest}")

print(f"\nTotal Subscription: {total_subscription}")
print(f"Total Recovery: {total_recovery}")
print(f"Total Withdrawal: {total_withdrawal}")
print(f"Total Interest: {total_interest}")

yearly_balance=z+total_subscription+total_recovery+total_interest-total_withdrawal
print(f"Yearly Balance: {yearly_balance}")
```

```
Enter withdrawal for July: 45000
Enter interest rate for July: 7.1
Balance for July: 537984
Interest for July: 3183.072
Enter subscription for August: 5000
Enter recovery for August: 3000
```

Figure 3: Image of Python File for Checking of GPF Interest



ceiling of january

```
ceiling_january=df
ceiling_january['THREE MONTH DIFFERENCE FROM CEILING']=500000-data['TILLNOVEMBER']
ceiling_january['OPTIMUM SUBSCRIPTION FROM DECEMBER']=(500000-data['TILLNOVEMBER'])/3
ceiling_january[['NAME OF OFFICIAL','GPF ACCOUNT NO','TILLNOVEMBER','TILLDECEMBER','TILLJANUARY','DOR','THREE MONTH DIFFERENCE FROM CEILING','OPTIMUM SUBSCRIPTION FROM DE
```

```
<ipython-input-85-248f4dd53e6b>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  ceiling_january['THREE MONTH DIFFERENCE FROM CEILING']=500000-data['TILLNOVEMBER']
<ipython-input-85-248f4dd53e6b>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  ceiling_january['OPTIMUM SUBSCRIPTION FROM DECEMBER']=(500000-data['TILLNOVEMBER'])/3
```

	NAME OF OFFICIAL	GPF ACCOUNT NO	TILLNOVEMBER	TILLDECEMBER	TILLJANUARY	DOR	THREE MONTH DIFFERENCE FROM CEILING	OPTIMUM SUBSCRIPTION FROM DECEMB
15	UMA KANTA SAHA	2064.0	450000.0	500000.0	550000.0	2029-01-31	50000.0	16666.6666
17	PRASANTA KUMAR SARDAR	2075.0	450000.0	500000.0	550000.0	2025-11-30	50000.0	16666.6666
21	JITENDRA PRASAD SRIVASTVA	2028.0	450000.0	500000.0	550000.0	2031-01-31	50000.0	16666.6666

Figure 4: Image of Python File for Checking of Subscription Ceiling

References

1. Ngpal, A. (2019). Python for Data Analytics, Scientific and Technical Applications. *IEEE Xplore*, DOI: 10.1109/AICAI.2019.8701341.

2. Patterson, E. (2017). Dataflow representation of data analyses: Toward a platform for collaborative data science. *IEEE Xplore*, 9.

3. Sukhdeve, D. s. (2023). Google Cloud Platform for Data Science. *https://link.springer.com/chapter/10.1007/978-1-4842-9688-2_2*, 11-34.

4. Vashisht, P. (2015). Big data analytics techniques: A survey. *IEEE Xplore*, DOI: 10.1109/ICGCIoT.2015.7380470.

5. Challa, N. (2023). Data analytics and its impact on future. *Research Gate*, https://www.researchgate.net/publication/371665415_DATA_ANALYTICS_AND_ITS_IMPACT_ON_FUTURE.