./

GENESIS- Advanced Python Programming Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1.0 | 22-03-2020 | Abhishek Guria |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc67351121)

[PROBLEM STATEMENT:- 4](#_Toc67351122)

[DEFINING THE IDEA FOR PROJECT : 4](#_Toc67351123)

[DETAILED REQUIREMENTS:- 5](#_Toc67351125)

[SWOAT ANALYSIS:- 6](#_Toc67351128)

[4 WH1’S :- 6](#_Toc67351129)

[DESIGN:- 7](#_Toc67351130)

[STRUCTURAL DIAGRAM 8](#_Toc67351131)

BEHAVIORAL DIAGRAM [8](#_Toc67351132)

[SOURCE CODE:- 9](#_Toc67351133)

[CODE QUALITY:- 12](#_Toc67351134)

[LEARNING CERTIFICATES:- 13](#_Toc67351135)

[REFRENCES:- 14](#_Toc67351136)

# PROBLEM STATEMENT:-

To write program in python to extract data from sheets by giving one or more input and validate them and then to put the whole data of all the sheets of the same in Master sheet. Which will store them.

# DEFINING THE IDEA FOR PROJECT :

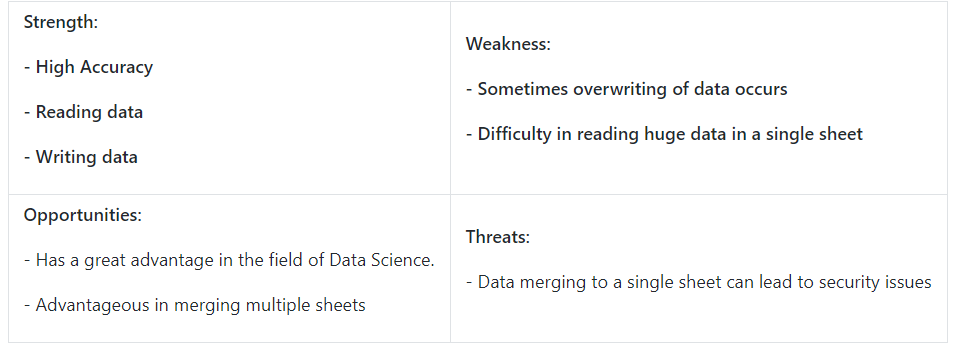
# Customized code is used to read and write data. An excel sheet made by hand containing 5 sheets and one master sheet. Here we search for the details of the person associated with the specific name and registration number on all 5 sheets. Once the data has been downloaded from the all sheets it will be printed on the master sheet. All its usage is used to read better search and write file. The code makes studying easier in the field of data science where there is a lot of data that needs to be extracted.

# DETAILED REQUIREMENTS:-

# 

# 

# SWOAT ANALYSIS:-



# 4 WH1’S :-

**Who** **:** Basically used in research field and data science fields where large data needs manipulation and extraction.

**What :** Xlsx file or CSV file with python code to read and write data

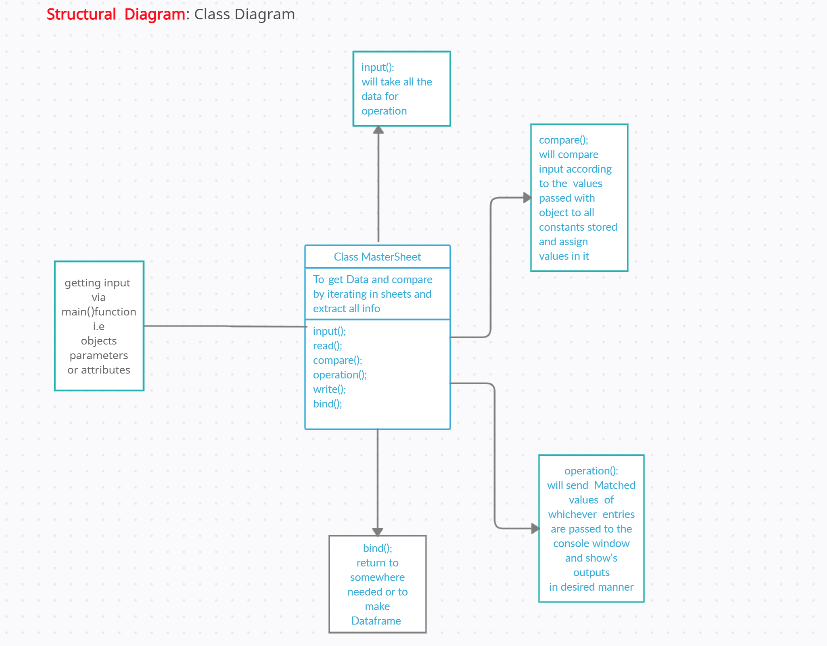
**When :** Used when large data needs to manipulated and extracted.

**Where :** In Research and technical fields

**How :** User friendly and easily accessible.

# DESIGN:-

**STRUCTRAL DIAGRAM**



# BEHAVIORAL DIAGRAM

# 

# SOURCE CODE:-

import pandas as pd

import openpyxl as op

from openpyxl import Workbook

from openpyxl.utils.dataframe import dataframe\_to\_rows

def read\_path\_of\_files():

    numberof\_sheets = int(input

                          ("Enter Number of sheets To "

                           "read And extract data: "))

    print("1. Enter the Paths of ", numberof\_sheets, " Excel sheets")

    for i in range(1, numberof\_sheets + 1):

        print('Path of sheet', i, ':')

        filepath\_ofSheet.append(input())

def read\_files():

    for fpath in (filepath\_ofSheet):

        workbok1 = op.load\_workbook(fpath)

        dff.append(pd.read\_excel(fpath))

        namesheet = workbok1.sheetnames

        print(namesheet)

        allsheetnames.append(namesheet[0])

        wb\_read.append(op.load\_workbook(fpath))

def write\_files\_in\_one():

    with pd.ExcelWriter('auto5sheets1.xlsx') as writer:

        for (df, a) in zip(dff, allsheetnames):

            df.to\_excel(writer, sheet\_name=a, index=False)

def header\_in\_new\_sheets():

    MasterSheet = Workbook()

    wb\_read = op.load\_workbook("auto5sheets1.xlsx")

    wsheet = MasterSheet.active

    wsheet.title = 'output'

    # sheets\_read = wb\_read.sheetnames

    for sheet in wb\_read.sheetnames:

        rs = wb\_read[sheet]

        Maxrow = rs.max\_row

        Maxcol = rs.max\_column

        if sheet == allsheetnames[0]:

            for j in range(1, Maxcol + 1):

                value = rs.cell(row=1, column=j).value

                wsheet.cell(row=1, column=j).value = value

        else:

            print("You are in now else block")

            maxColmaster = wsheet.max\_column

            for j in range(7, Maxcol + 1):

                value = rs.cell(row=1, column=j).value

                print(value)

                maxColmaster = maxColmaster + 1

                wsheet.cell(row=1, column=maxColmaster).value = value

    MasterSheet.save('Openpyxl.xlsx')

    wb\_read = op.load\_workbook("auto5sheets1.xlsx")

    count = int(input("Enter how Many Data you want to read"))

    for i in range(1, count + 1):

        print('Enter Details for the ', i, ' Data')

        FirstName = str(input('Enter First name '))

        Email = str(input('Enter email Id'))

        PS\_No = int(input('Enter PS Number '))

        Datatoload = []

        for sheet in wb\_read.sheetnames:

            rs = wb\_read[sheet]

            Maxrow = rs.max\_row

            Maxcol = rs.max\_column

            for i in range(2, Maxrow + 1):

                if rs.cell(row=i, column=1).value == \

                        PS\_No and rs.cell(row=i, column=2).value == \

                        FirstName and rs.cell(

                        row=i, column=6).value == Email:

                    if sheet == allsheetnames[0]:

                        for j in range(1, Maxcol + 1):

                            Datatoload.append(rs.cell(row=i, column=j).value)

                    else:

                        for j in range(7, Maxcol + 1):

                            Datatoload.append(rs.cell(row=i, column=j).value)

                    df = pd.DataFrame(Datatoload)

                    df = df.T

        for r in dataframe\_to\_rows(df, index=False, header=False):

            wsheet.append(r)

        MasterSheet.save('Openpyxl.xlsx')

if \_\_name\_\_ == '\_\_main\_\_':

    filepath\_ofSheet = []

    allsheetnames = []

    dff = []

    wb\_read = []

    read\_path\_of\_files()

    read\_files()

    write\_files\_in\_one()

    header\_in\_new\_sheets()

# D:\Python\python\_project\a\quiz\_1\_grades.xlsx

# D:\Python\python\_project\b\quiz\_2\_grades.xlsx

# D:\Python\python\_project\c\quiz\_3\_grades.xlsx

# D:\Python\python\_project\d\quiz\_4\_grades.xlsx

# D:\Python\python\_project\e\quiz\_5\_grades.xlsx

# 99003700  Richard Bennett Male    22  [richard.bennett@univ.edu](mailto:richard.bennett@univ.edu)

# 99003701  Timothy Parker  Female  25  [timothy.parker@univ.edu](mailto:timothy.parker@univ.edu)

# 99003704  Michael Taylor  Male    20  [michael.taylor@univ.edu](mailto:michael.taylor@univ.edu)

# CODE QUALITY:-

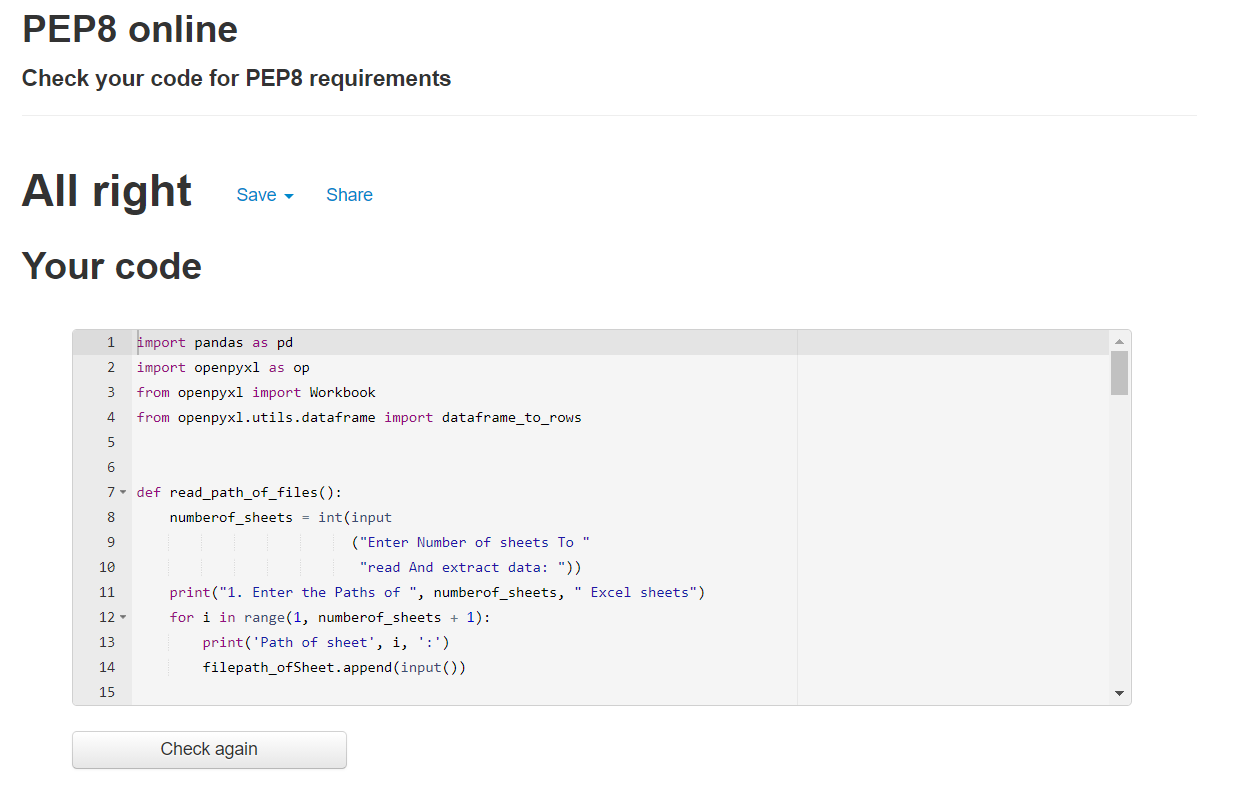


Figure 1 Code Quality

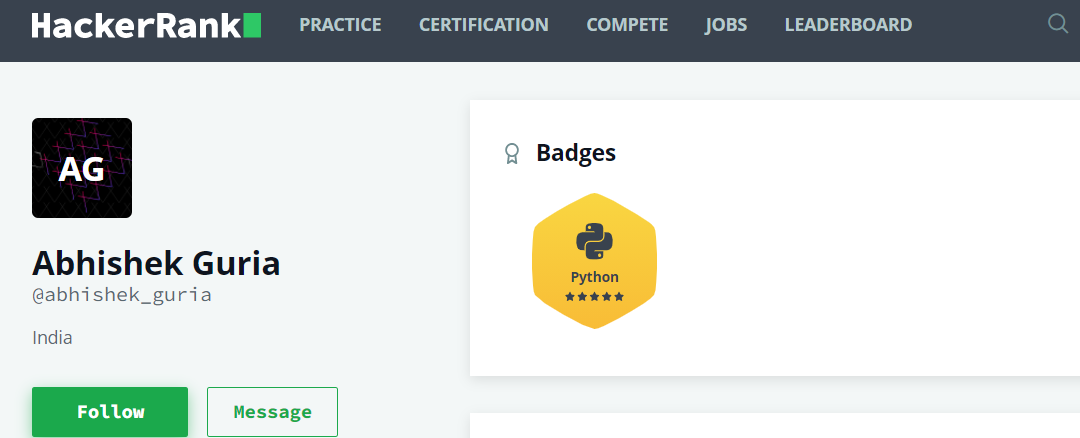
GITHUB LINK: [https://github.com/99003726/ Miniproject\_Python.git](https://github.com/99003726/%20Miniproject_Python.git%20t)

# LEARNING CERTIFICATES:-

1. SOLOLEARN PYTHON CORE COURSE



1. HACKERANK GOLD BADGE



# REFRENCES:-

* JAVATPOINT
* GEEKS FOR GEEKS
* PANDAS DOCUMENTATION
* OPENPYXL DOCUMENTATION
* WIKIPEDIA
* HACKERANK
* SOLOLEARN