**A PROJECT DOCUMENT ON THE TOPIC**

**“SUBSTITUTION TABLE GENERATOR”**



Submitted by

**E. MANOJ**

**XII – A**

Under the guidance of

**Dr. Magesh Kumar**

**PGT (Computer Science)**

Department of Computer Science

Kendriya Vidyalaya CRPF Avadi

Chennai – 600 065**Kendriya Vidyalaya CRPF Avadi**

**Chennai – 600 065**

**Department of Computer Science**

**-----------------------------------------------------------------------------**

**CERTIFICATE**

This is to certify that **E. MANOJ** of class **XII – A** has prepared a **PYTHON PROJECT** on the topic **“SUBSTITUTION TABLE GENERATOR”**. The document is the result of efforts and endeavors. The document is found worthy of acceptance as final project for the subject **COMPUTER SCIENCE** of class XII. He has prepared the report under my guidance.

CBSE ROLL NUMBER:

**Signature of Examiner Dr. Magesh Kumar**

**PGT (Computer Science)**

Department of Computer Science

Kendriya Vidyalaya CRPF

DATE: Avadi Chennai – 600 065

**Kendriya Vidyalaya CRPF Avadi**

**Chennai – 600 065**

**Department of Computer Science**

**------------------------------------------------------------------------------**

**DECLARATION**

I hereby declare that the project work entitled **“SUBSTITUTION TABLE GENERATOR”,** submitted to Department of Computer Science, Kendriya Vidyalaya CRPF Avadi Chennai – 600 065 is prepared by me. All the works are the result of my personal effort.

**E. MANOJ**

**XII – A**

**ACKNOWLEDGEMENT**

I would like to express a deep sense of thanks & gratitude to my project guide Dr. MAGESH KUMAR, PGT Computer Science for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & constant motivation have been responsible for the completion of this project.

My sincere thanks to Smt. N. VALARMATHI, our principal madam, for her coordination in extending every possible support for the completion of this project.

I also thank my parents for their motivation & support. I must also thank my classmates for their timely help & support for the compilation of this project.

Last but not the least, I would like to thank all those who had helped me directly and indirectly towards the completion of this project.

**E. MANOJ**

**XII - A**

**CONTENTS**

1. Introduction .…………………………. . 6

2. Project Objective ……………………... 7

3. Requirements …………………………. 8

4. Source Code ………………………….. 9

5. Flow chart …………………………… 18

6. Advantage and Disadvantages ………. 18

7. Precautions ………………………….. 19

8. References …………………………... 19

**INTRODUCTION**

The project is based on our schools. As a student we take leave, likewise a teacher may also take leave. If he or she is not present then an alternate teacher will be provided arrangement for the class of the absent teacher. This work of giving substitution is taken care by the group of timetable incharge in the schools. They perform every permutations and combinations to get the perfect teacher for arrangement. Their work is a bit troublesome like duplication of data’s, overlapping, mismatching and time consuming.

To overcome this problems and to make it simple and efficient, the following project was coded, it has a simple mechanism, it uses the database of the teacher schedule bring out the match. The output is quick; the overlapping chaos and data duplication is removed. The program performs every permutation for the match.

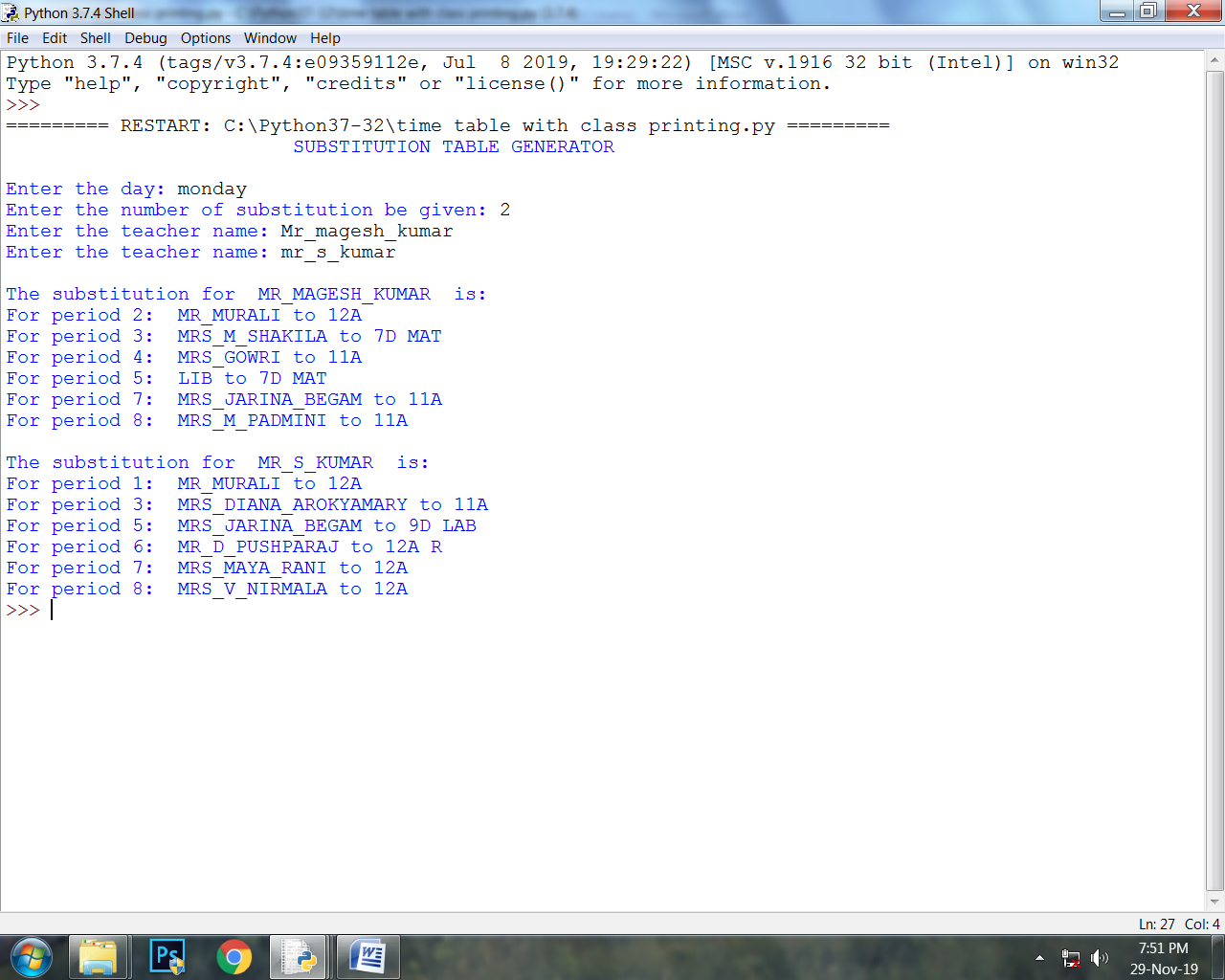
The program uses the data from a excel sheet, in a preprogrammed manner, and it displays the substitution details in the screen.

**PROJECT OBJECTIVE**

The project title **“SUBSTITUTION TABLE GENERATOR”** generates the Substitution Time Table for the absent teachers using the time schedule of the respective teacher.

The .py application created takes the required inputs from the end user and process that. A random arrangement for every absent teacher is given with respect to the availability of the teacher who doesn’t have a class at that particular period.

The output screen is application:

****

**REQUIREMENTS**

The basic requirements are: the source code, Python IDLE or other python software and the whole database in a single directory.

**System Properties:**

* Ram size >1GB
* Hard disk space >2 GB
* Operating System: Windows 7 or higher versions, macOS
* Python Idle version above 3.6

And basic computer knowledge.

**SOURCE CODE**

import csv

import random

print("\t\t\t SUBSTITUTION TABLE GENERATOR")

print()

masterlist=[]

sublist=[]

mast\_list=[]

P1=[]

P2=[]

P3=[]

P4=[]

P5=[]

P6=[]

P7=[]

P8=[]

day\_chk=True

while day\_chk:

day=str(input("Enter the day: "))

if day.lower()=='monday':

file\_to\_open= r'G:\Time Table\monday.csv'

day\_chk=False

elif day.lower()=='tuesday':

file\_to\_open= r'G:\Time Table\tuesday.csv'

day\_chk=False

elif day.lower()=='wednesday':

file\_to\_open= r'G:\Time Table\wednesday.csv'

day\_chk=False

elif day.lower()=='thursday':

file\_to\_open= r'G:\Time Table\thursday.csv'

day\_chk=False

elif day.lower()=='friday':

file\_to\_open= r'G:\Time Table\friday.csv'

day\_chk=False

elif day.lower()=='saturday':

file\_to\_open= r'G:\Time Table\saturday.csv'

day\_chk=False

else:

print("Check the day entered !!")

print()

no=int(input("Enter the number of substitution be given: "))

named=[]

for zd in range(0,no):

name2=str(input("Enter the teacher name: "))

name1=name2.lower()

if name1 not in named:

named.append(name1)

for name in named:

t=open (file\_to\_open,'r')

f=csv.reader(t)

fla=0

for row in f:

if name == row[0].lower():

bp=[]

for j in range(0,len(row)):

if row[j]=='1':

bp.append(j)

fla=1

t.close()

if fla==1:

u=open(file\_to\_open,'r')

g=csv.reader(u)

sub\_tr={}

for arow in g:

for k in bp:

if arow[k]=='0':

sub\_tr[k]=arow[0]

if sub\_tr != {}:

subl=sub\_tr.keys()

sublist=list(subl)

mast\_list.append(arow[0])

mast\_list.append(sublist)

masterlist.append(mast\_list)

mast\_list=[]

sub\_tr={}

p1=[]

p2=[]

p3=[]

p4=[]

p5=[]

p6=[]

p7=[]

p8=[]

for tr in range(0,len(masterlist)):

for bpp in bp:

if bpp in masterlist[tr][1]:

if bpp==1:

p1.append(masterlist[tr][0].lower())

elif bpp==2:

p2.append(masterlist[tr][0].lower())

elif bpp==3:

p3.append(masterlist[tr][0].lower())

elif bpp==4:

p4.append(masterlist[tr][0].lower())

elif bpp==5:

p5.append(masterlist[tr][0].lower())

elif bpp==6:

p6.append(masterlist[tr][0].lower())

elif bpp==7:

p7.append(masterlist[tr][0].lower())

else:

p8.append(masterlist[tr][0].lower())

final\_list=['-','-','-','-','-','-','-','-']

def tr\_selector(fap,alreadytr,abse):

flag=True

con=0

infi\_chk=0

xa=len(fap)-1

while flag:

P=random.randint(0,xa)

ya=fap[P]

if ya.lower() not in alreadytr:

con+=1

if ya.lower() not in abse:

con+=1

if con ==2:

alreadytr.append(ya)

return ya

flag=False

else:

con=0

flag=True

else:

infi\_chk+=1

if infi\_chk>100:

return None

else:

flag=True

if p1 != []:

tchr1=tr\_selector(p1,P1,named)

if tchr1 != None:

final\_list[0]=tchr1

else:

final\_list[0]="self study"

if p2 != []:

tchr2=tr\_selector(p2,P2,named)

if tchr2 != None:

final\_list[1]=tchr2

else:

final\_list[1]="self study"

if p3 != []:

tchr3=tr\_selector(p3,P3,named)

if tchr3 != None:

final\_list[2]=tchr3

else:

final\_list[2]="self study"

if p4 != []:

tchr4=tr\_selector(p4,P4,named)

if tchr4 != None:

final\_list[3]=tchr4

else:

final\_list[3]="self study"

if p5 != []:

tchr5=tr\_selector(p5,P5,named)

if tchr5 != None:

final\_list[4]=tchr5

else:

final\_list[4]="self study"

if p6 != []:

tchr6=tr\_selector(p6,P6,named)

if tchr6 != None:

final\_list[5]=tchr6

else:

final\_list[5]="self study"

if p7 != []:

tchr7=tr\_selector(p7,P7,named)

if tchr7 != None:

final\_list[6]=tchr7

else:

final\_list[6]="self study"

if p8 != []:

tchr8=tr\_selector(p8,P8,named)

if tchr8 != None:

final\_list[7]=tchr8

else:

final\_list[7]="self study"

print()

l\_class=[]

clr=open(file\_to\_open,'r')

clr\_r=csv.reader(clr)

for clrow in clr\_r:

if clrow[0].lower()== name.lower():

for cl\_row in range(9,17):

l\_class.append(clrow[cl\_row])

per1=final\_list[0].upper()

per2=final\_list[1].upper()

per3=final\_list[2].upper()

per4=final\_list[3].upper()

per5=final\_list[4].upper()

per6=final\_list[5].upper()

per7=final\_list[6].upper()

per8=final\_list[7].upper()

print("The substitution for ",name.upper() ," is: ")

if per1 != '-':

print("For period 1: ",per1 ,"to",l\_class[0])

if per2 != '-':

print("For period 2: ",per2 ,"to",l\_class[1])

if per3 != '-':

print("For period 3: ",per3 ,"to",l\_class[2])

if per4 != '-':

print("For period 4: ",per4 ,"to",l\_class[3])

if per5 != '-':

print("For period 5: ",per5 ,"to",l\_class[4])

if per6 != '-':

print("For period 6: ",per6 ,"to",l\_class[5])

if per7 != '-':

print("For period 7: ",per7 ,"to",l\_class[6])

if per8 != '-':

print("For period 8: ",per8 ,"to",l\_class[7])

u.close()

t.close()

else:

print()

print("The entered name ",name.upper()," not found in the teachers list" )

**FLOW CHART**

**ADVANTAGE AND**

**DISADVANTAGE**

|  |  |
| --- | --- |
| **ADVANTAGE** | **DISADVANTAGE** |
| User friendly | It is fully dependent on data. |
| Reduce time consumption | Number of absent teachers should be less than 13 |
| Reduces data redundancy | It is random based , it’s not preferential |
| More precise | Less accurate |
| Non- case sensitive | It is spell sensitive. |

**PRECAUTION**

* Check whether your system requirements are satisfied for the working of program.
* It is spelling sensitive; the spelling should be correctly entered.
* The number of substitution to be given should be less than 13.

**REFERENCES**

* <https://www.geeksforgeeks.org>
* <https://stackoverflow.com>
* Computer Science with Python – Sumita Arora
* [www.google.com](http://www.google.com)