**RMK SENIOR SECONDARY SCHOOL**

**THIRUVERKADU, CHENNAI-77**



(Affiliated to the Central Board of Secondary Education ,Delhi)

RSM Nagar, Sundracholapuram Road Thiruverkadu,

Chennai-600 077 Website: rmkschool.in

ROLL NO:

**ALL INDIA SENIOR SCHOOL CERTIFICATE EXAMINATION**

**Academic year: 2020-2021**

**DEPARTMENT OF COMPUTER SCIENCE**

**RMK SENIOR SECONDARY SCHOOL**

**THIRUVERKADU, CHENNAI-77**



**DEPARTMENT OF COMPUTER SCIENCE**

*This is to certify that S.Sakthi Sairam of class 12 has performed ONLINE CLASS TIMETABLE SCHEDULER during the year 2020-2021 at the school laboratory and that his/her project report is certified as bonafide .*

*His/Her examination roll no is*

*.....................................*

*Head of Department*

*Computer Science*

*Date...................*

*Submitted for the All-India Senior Secondary School Certificate Examination-2021 (Practical Examination) in Computer Science at***RMK SENIOR SECONDARY SCHOOL***On ....................................*

*........................................ ........................................*

*Internal Examiner External Examiner Date .............................. Date ..............................*

*Signature of Principal*

*…….............................*

**ONLINE CLASS SCHEDULER**

BY: S.Sakthi Sairam

XII-A

**ACKNOWLEDGEMENT**

I hereby acknowledge all those people, who have helped for the successful completion of my project to a great extent.

I would also like to convey a special thanks to our principal, **Mrs. SudhaMalini**, and the school management for having provided me with complete access to all materials and information sources available in the laboratory.

I express my deepest gratitude to our Computer Teacher, **Smt. L. Ramadevi**, for her untiring help and encouragement during the course of the project titled ::

“**ONLINE CLASS SCHEDULER**”

I would also like to extend my thanks to other members of the Computer Department for their co-operation.

Index

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Title** | **Page No.** |
| 1. | Introduction | 1&2 |
| 2. | Algorithm | 2 |
| 3. | Source Code | 10 |
| 4. | Output Screens | 30 |
| 5. | Bibliography  - | 41 |

Introduction

In this online class time table scheduler project, Teacher can add the particular subject's class at particular time slot available. Principal has a separate login where he/she can add the subject techers so that teachers can login and add the classes. Principal can also perform adding a subject class, view the timetable, update a particular thing in the timetable, search a particular class with respect to timetable numer, class,time or section and etc, or delete a particular record in timetable. Teachers can also add, update, search and delete the timetable but teachers can only alter their subject class, they don't have the rights to alter other subjects other than their subject. Students can view, search in the timetable. the class will be automatically deleted from the timetable after the class is over. Principal and teachers login details are stored in binary files and the timetable is maintained in the MySQL database

Algorithm

INBUILT FUNCTIONS::**print(),input(),len(),format(),upper(),lower(),Int(),str(),open()**

Imported function::**connect(),is\_connected(),cursor(),execute(),**

**commit(),fetchall(),load(),dump(),exit(),tabulate(),date()**

user defined function:

**insert(),showtt(),deletion(),autodel(),update(),search()**

---------------------------------------

FUNCTION NAME:print()

PARAMETERS PASSED**:**variable/sequence

RETURN TYPE: None

* **Prints any string/variables**

---------------------------------------

FUNCTION NAME:input()

PARAMETERS PASSED**:**prompt message

RETURN TYPE: string

* **Stores user given data**

---------------------------------------

FUNCTION NAME:len()

PARAMETERS PASSED**:**sequence

RETURN TYPE: int

* **Returns the length of given sequence**

---------------------------------------

FUNCTION NAME:format()

PARAMETERS PASSED**:**variable

RETURN TYPE: string

* **It set a variable enclosed in a tuple**

---------------------------------------

FUNCTION NAME:upper()

PARAMETERS PASSED**:**nil

RETURN TYPE: string

* **Convert the alphates to uppercase.**

---------------------------------------

FUNCTION NAME:lower()

PARAMETERS PASSED**:**nil

RETURN TYPE: string

* **Convert the alphates to lowercase.**

---------------------------------------

FUNCTION NAME:int

PARAMETERS PASSED**:**variables

RETURN TYPE: int

* **Converts the specified value into integer .**

---------------------------------------

FUNCTION NAME:str()

PARAMETERS PASSED**:**variables

RETURN TYPE: str

* **Converts the specified value into string.**

---------------------------------------

FUNCTION NAME:open()

PARAMETERS PASSED**:**path,mode

RETURN TYPE: nil

* **It opens a file and returns it as file object**

---------------------------------------

FUNCTION NAME:connect()

PARAMETERS PASSED**:**host,username,passwd

RETURN TYPE: nil

* **Used to connect MYSQL**

---------------------------------------

FUNCTION NAME:is**\_**connected()

PARAMETERS PASSED**:**nil

RETURN TYPE: bool

* **Checks whether python is connected to python**

---------------------------------------

FUNCTION NAME:cursor()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **Instantiates MYSQLcursor object**

---------------------------------------

FUNCTION NAME:execute()

PARAMETERS PASSED**:**query

RETURN TYPE: None

* **Executes the given query**

---------------------------------------

FUNCTION NAME:commit()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **Changes made in the database are made permanent**

---------------------------------------

FUNCTION NAME:fetchall()

PARAMETERS PASSED**:**nil

RETURN TYPE: list

* **Fetch all the result in the table**

---------------------------------------

FUNCTION NAME:load()

PARAMETERS PASSED**:**variable

RETURN TYPE: list

* **Load or collect the data from the file**

---------------------------------------

FUNCTION NAME:dump()

PARAMETERS PASSED**:**object,file

RETURN TYPE: nil

* **Dump or save the contents to the file**

---------------------------------------

FUNCTION NAME:exit()

PARAMETERS PASSED**:**nil

RETURN TYPE: nil

* **Allows to quit**

---------------------------------------

FUNCTION NAME:tabulate()

PARAMETERS PASSED**:**data,headers,tablefmt

RETURN TYPE: nil

* **Allows to create the data in tabular form**

---------------------------------------

FUNCTION NAME:date()

PARAMETERS PASSED**:**nil

RETURN TYPE: nil

* **Shows the current date**

---------------------------------------

FUNCTION NAME:insert()

PARAMETERS PASSED**:**subject

RETURN TYPE: None

* **Allows the data to be inserted to the table**

---------------------------------------

FUNCTION NAME:showtt()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **Allows to see the contents in the table**

---------------------------------------

FUNCTION NAME:deletion()

PARAMETERS PASSED**:**var

RETURN TYPE: None

* **Allows to delete the contents in the table**

---------------------------------------

FUNCTION NAME:autodel()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **It automatically delets the contents if expired(date)**

---------------------------------------

FUNCTION NAME:update()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **It allows to update the contents in the table**

---------------------------------------

FUNCTION NAME:search()

PARAMETERS PASSED**:**nil

RETURN TYPE: None

* **It allows to search the contents in the table**

---------------------------------------

**SOURCE CODE**

a="ONLINE CLASS TIMETABLE SHEDULER"

from pickle import load,dump #importing modules

import sys

importmysql.connector as mycon

from tabulate import tabulate

fromdatetime import date

import time

print("%100s"%" ",end="")

time.sleep(0.5)

for i in a:

print(i,end="")

time.sleep(0.1)

print(" ")

#user defined functions

def insert(subject):

clas = input("Enter Class (in Roman letters) ")

section = input("Enter Section ")

date = input("Enter Date (YYYY-MM-DD)")

if len(date)!=10:

print("DATE value is wrong, it should be in[YYYY-MM-DD] format")

print("for example: 2020-08-09")

print("program exiting")

sys.exit()

timing = input("Enter The Timing (HH:MM) ")

If len(timing)!=5:

print("timing value is wrong, it should be in[HH:MM] format")

print("for example: 08:30")

print("program exiting")

sys.exit()

ttnum=int(input("Enter time table number"))

qry = "insert into octimetable values ({},'{}','{}','{}','{}','{}')".format(ttnum,clas,section,date,timing,subject)

cur.execute(qry)

con.commit()

count = cur.rowcount

print(count," record is inserted successfully.")

print("Press Enter to continue.")

input()

def showtt():

print("SHOWING THE TIME TABLE")

qry = "Select \* from octimetable "

cur.execute(qry)

data = cur.fetchall()

count = cur.rowcount

h=['ttnum','clas','section','date','timing','subject']

print(tabulate(data,headers=h,tablefmt='grid'))

print("Press Enter to continue.")

input()

def deletion(var):

ttnum = input("Enter time table number to be deleted")

subject = var

qry = "delete from octimetable where ttnum = {} and subject='{}'".format(ttnum,subject)

cur.execute(qry)

con.commit()

count = cur.rowcount

if count>0:

print("Record is deleted successfully.")

else:

print("Record is NOT deleted")

print("Press Enter to continue.")

input()

def autodel():

today=date.today()

today=str(today)

qry = "delete from octimetable where date<'{}'".format(today)

cur.execute(qry)

con.commit()

count = cur.rowcount

def update():

print("UPDATION::")

showtt()

choice=input("WHAT YOU WANT TO UPDATE::CLASS-C,SECTION-S,DATE-D,TIME-T,SUBJECT-SUB").upper()

ttno=int(input("ENTER THE TABLE NO::"))

if choice=='C':

clas=input("ENTER THE NEW CLASS::")

qry="update octimetable set clas='{}' where ttnum={}".format(clas,ttno)

cur.execute(qry)

con.commit()

count=cur.rowcount

if count>0:

print("UPDATED SUCCESSFULLY::")

else:

print("INVALID TABLENUMBER")

elif choice=='S':

sec=input("ENTER THE NEW SECTION::")

qry="update octimetable set section='{}' where ttnum={}".format(sec,ttno)

cur.execute(qry)

con.commit()

count=cur.rowcount

if count>0:

print("UPDATED SUCCESSFULLY::")

else:

print("INVALID TABLENUMBER")

elif choice=='D':

date=input("ENTER THE NEW DATE(YYYY-MM-DD,EX-2020-01-02)::")

Len=len(date)

if Len==10:

qry="update octimetable set date='{}' where ttnum={}".format(date,ttno)

cur.execute(qry)

con.commit()

count=cur.rowcount

if count>0:

print("UPDATED SUCCESSFULLY::")

else:

print("NO RECORDS FOUND")

else:

print("INVALID DATE CHECK IT AGAIN::")

elif choice=='T':

timing=input("ENTER THE NEW TIMING(HH:MM)::")

qry="update octimetable set timing='{}' where ttnum={}".format(timing,ttno)

cur.execute(qry)

con.commit()

count=cur.rowcount

if count>0:

print("UPDATED SUCCESSFULLY::")

else:

print("NO RECORDS FOUND")

elif choice=='SUB':

subject=input("ENTER THE SUBJECT::")

qry="update octimetable set subject='{}' where ttnum={}".format(subject,ttno)

cur.execute(qry)

con.commit()

count=cur.rowcount

if count>0:

print("UPDATED SUCCESSFULLY::")

else:

print("NO RECORDS FOUND")

else:

print("INVALID OPTION::")

print("Press Enter to continue.")

input()

def search():

print("SEARCHING::")

showtt()

opt=input("SEARCH USING---TABLENUMBER-T,CLASS-C,SECTION-S,DATE-D,TIME-TI,SUBJECT-SUB").upper()

if opt=='T':

tno=int(input("ENTER THE TABLE NO::"))

qry="select \* from octimetable where ttnum='{}'".format(tno)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("NO RECORDS FOUND::")

elif opt=='C':

Class=input("ENTER THE CLASS::")

qry="select \* from octimetable where clas='{}'".format(Class)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("NO RECORDS FOUND::")

elif opt=='S':

sec=input("ENTER THE SECTION::")

qry="select \* from octimetable where section='{}'".format(sec)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("NO RECORDS FOUND::")

elif opt=='D':

date=input("ENTER THE DATE(YYYY-MM-DD,EX-2020-01-02)::")

Len=len(date)

if Len==10:

qry="select \* from octimetable where date='{}'".format(date)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("NO RECORDS FOUND")

else:

print("INVALID DATE CHECK IT AGAIN::")

elif opt=='TI':

time=input("ENTER THE TIME(HH:MM)::")

Len=len(time)

if Len==5:

qry="select \* from octimetable where timing='{}'".format(time)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("INVALID TIME::")

else:

print("PLEASE CHECK THE FORMAT AGAIN")

elif opt=='SUB':

sub=input("ENTER THE SUBJECT::")

qry="select \* from octimetable where subject='{}'".format(sub)

cur.execute(qry)

data=cur.fetchall()

count=cur.rowcount

if count>0:

h=['TABLENUMBER','CLASS','SECTION','DATE','TIME','SUBJECT']

print(tabulate(data,headers=h,tablefmt='grid'))

input("ENTER ANY KEY::")

else:

print("NO RECORDS FOUND::")

else:

print("INVALID OPTION::")

sys.exit()

#MAIN PROGRAM

con = mycon.connect(host='localhost',user='root',passwd='root') #connecting mysql

if con.is\_connected():

cur = con.cursor()

cur.execute("Create database if not exists onlineclass\_sheduler") #Creating Database

cur.execute("use onlineclass\_sheduler") #Using Database

cur.execute("create table if not exists octimetable(ttnumint(6) primary key,clas char(4),\

section char(1),date char(15),timing char(5), subject char(20))")

else:

print("mysql not connected")

#admin logins

try:

with open ('principleadmin.dat','rb') as fin:

rec=load(fin)

except:

with open ('principleadmin.dat','wb') as fout:

print("SIGNIN PRINCIPLE ADMIN")

adname=input("ENTER ADMIN USERNAME:")

adpass=input("ENTER ADIMIN PASSWORD :")

print("signin successfully")

rec=[adname,adpass]

dump(rec,fout)

#validation for admins

print(" ")

time.sleep(0.25)

print("WHO ARE YOU? student,subject teacher or principal")

print(" ")

time.sleep(0.25)

who=input("if you are a STUDENT press 's',TEACHERpress't',PRINCIPALpress'p'::").lower()

print(" ")

time.sleep(0.25)

autodel()

if who=='p': #principal interface

adname=input("ENTER ADMIN USERNAME:")

adpass=input("ENTER ADIMIN PASSWORD :")

ifadname==rec[0] and adpass==rec[1]:

print("permission granted")

else:

print("permission denied")

sys.exit()

while True:

print(" ")

time.sleep(0.5)

print("press 1:To add subject teachers")

time.sleep(0.5)

print("press 2:To insert a class in the timetable")

time.sleep(0.5)

print("press 3:To view timetable")

time.sleep(0.5)

print("press 4:To update anything in time table")

time.sleep(0.5)

print("press 5:To delete a class in time table")

time.sleep(0.5)

print("press 6:To search for anything in time table")

time.sleep(0.5)

print("press 7: To exit")

time.sleep(0.5)

print(" ")

cho=int(input("enter your choice::"))

if cho==7:

print("exited")

sys.exit()

elif cho==5:

showtt()

print("Delete a Record")

var=input("Enter The subject to be deleted ")

deletion(var)

elif cho==3:

showtt()

elif cho==4:

update()

elif cho==2:

print("Inserting a Record")

subject = input("Enter Subject ").lower()

insert(subject)

elif cho==6:

search()

elif cho==1:

try:

fin=open('teacherslogin.bin','rb')

rec1=load(fin)

fin.close()

except :

rec1=[]

ch='y'

fout=open('teacherslogin.bin','wb')

while ch=='y':

name=(input("ENTER TEACHER NAME:"))

pas=input("SET PASSWORD:")

sub=input("ENTER SUBJECT::")

rec1+=[[name,pas,sub]]

print("SUCCESSFULLY ADDED A NEW TEACHER")

ch=input("WANT TO ADD MORE LOGIN DETAILS: y or n").lower

dump(rec1,fout)

fout.close()

else:

print("wrong choice")

elif who=='t': #teacher interface

try:

fin=open('teacherslogin.bin','rb')

r=load(fin)

fin.close()

except:

print("first principal should add you first")

sys.exit()

tname=(input("ENTER TEACHER NAME:"))

tpas=input("PASSWORD:")0

for i in r:

if i[0]==tname and i[1]==tpas:

print("teacher access granted")

dsub=i[2]

break

else:

print("access denied")

sys.exit()

while True:

print(" ")

time.sleep(0.5)

print("press 1:To insert a class in the timetable")

time.sleep(0.5)

print("press 2:To deletion")

time.sleep(0.5)

print("press 3:To view timetable")

time.sleep(0.5)

print("press 4:To search for anything in time table")

time.sleep(0.5)

print("press 5:To exit")

time.sleep(0.5)

print(" ")

cho=int(input("enter your choice::"))

if cho==5:

print("exited")

sys.exit()

elif cho==4:

search()

elif cho==1:

print("Inserting a Record")

Insert(dsub)

elif cho==3:

showtt()

elifcho==2:

showtt()

print("Delete a Record")

deletion(dsub)

else:

print("wrong choice")

else:#student login

while True:

print(" ")

time.sleep(0.5)

print("press 1:To view timetable")

time.sleep(0.5)

print("press 2:To search anything in timetable")

time.sleep(0.5)

print("press 3:To exit")

time.sleep(0.5)

print(" ")

cho=int(input("enter your choice"))

if cho==1:

showtt()

elif cho==3:

print("terminating")

sys.exit()

elif cho==2:

search()

else:

print("wrong choice")

**OUTPUT SCREEN**

ONLINE CLASS TIMETABLE SHEDULER

SIGNIN PRINCIPLE ADMIN

ENTER ADMIN USERNAME:xyz

ENTER ADIMIN PASSWORD :123456

signin sucesfull

WHO ARE YOU? student,subject teacher or principal

#PRINCIPAL’S VIEW::

if you are a STUDENT press 's',TEACHER press't',PRINCIPAL press'p'::p

ENTER ADMIN USERNAME:xyz

ENTER ADIMIN PASSWORD :123456

permission granted

press 1:To add subject teachers

press 2:To insert a class in the timetable

press 3:To view timetable

press 4:To update anything in time table

press 5:To delete a class in time table

press 6:To search for anything in time table

press 7: To exit

1.ADDING SUBJECT TEACHERS::

enter your choice::1

ENTER TEACHER NAME:xxxx

SET PASSWORD:001

ENTER SUBJECT::computer science

SUCCESSFULLY ADDED A NEW TEACHER

WANT TO ADD MORE LOGIN DETAILS: y or n:y

2.INSERTION::

press 1:To add subject teachers

press 2:To insert a class in the timetable

press 3:To view timetable

press 4:To update anything in time table

press 5:To delete a class in time table

press 6:To search for anything in time table

press 7: To exit

enter your choice::2

Inserting a Record

Enter Subject computer science

Enter Class (in Roman letters) XII

Enter Section A

Enter Date (YYYY-MM-DD)2020-10-05

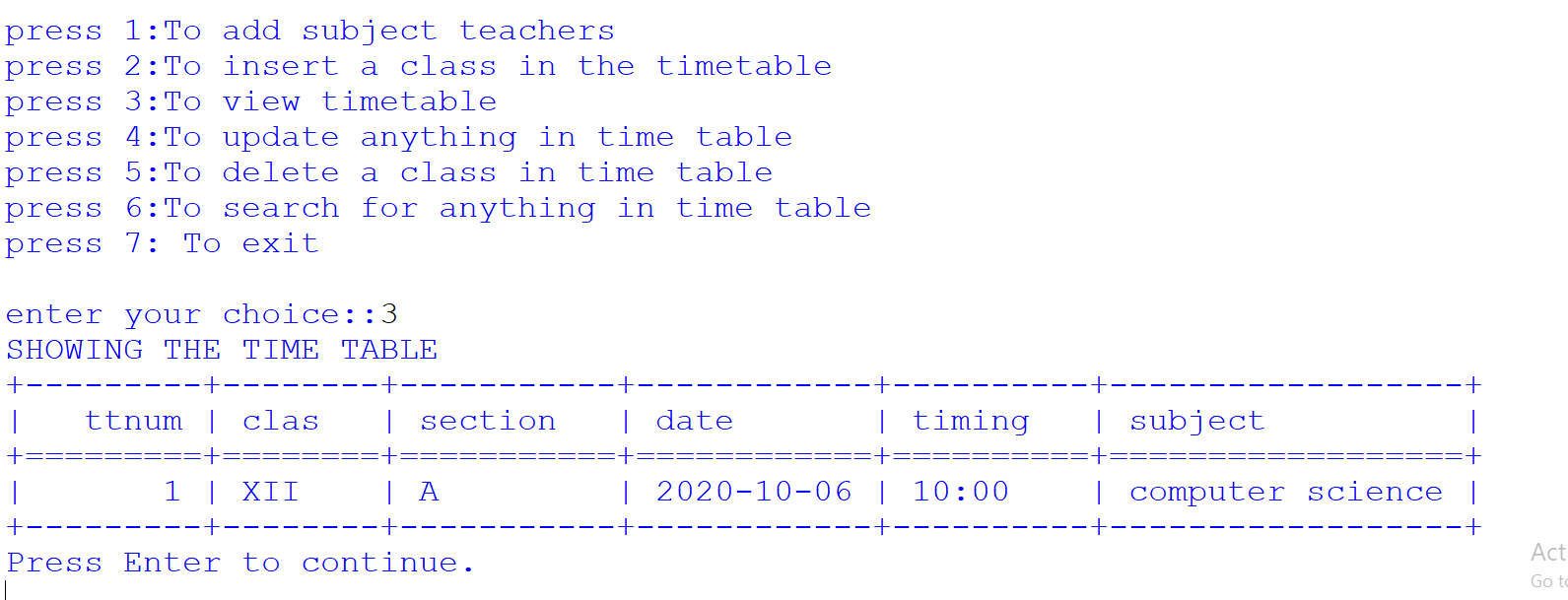
Enter The Timing (HH:MM) 10:00

Enter time table number1

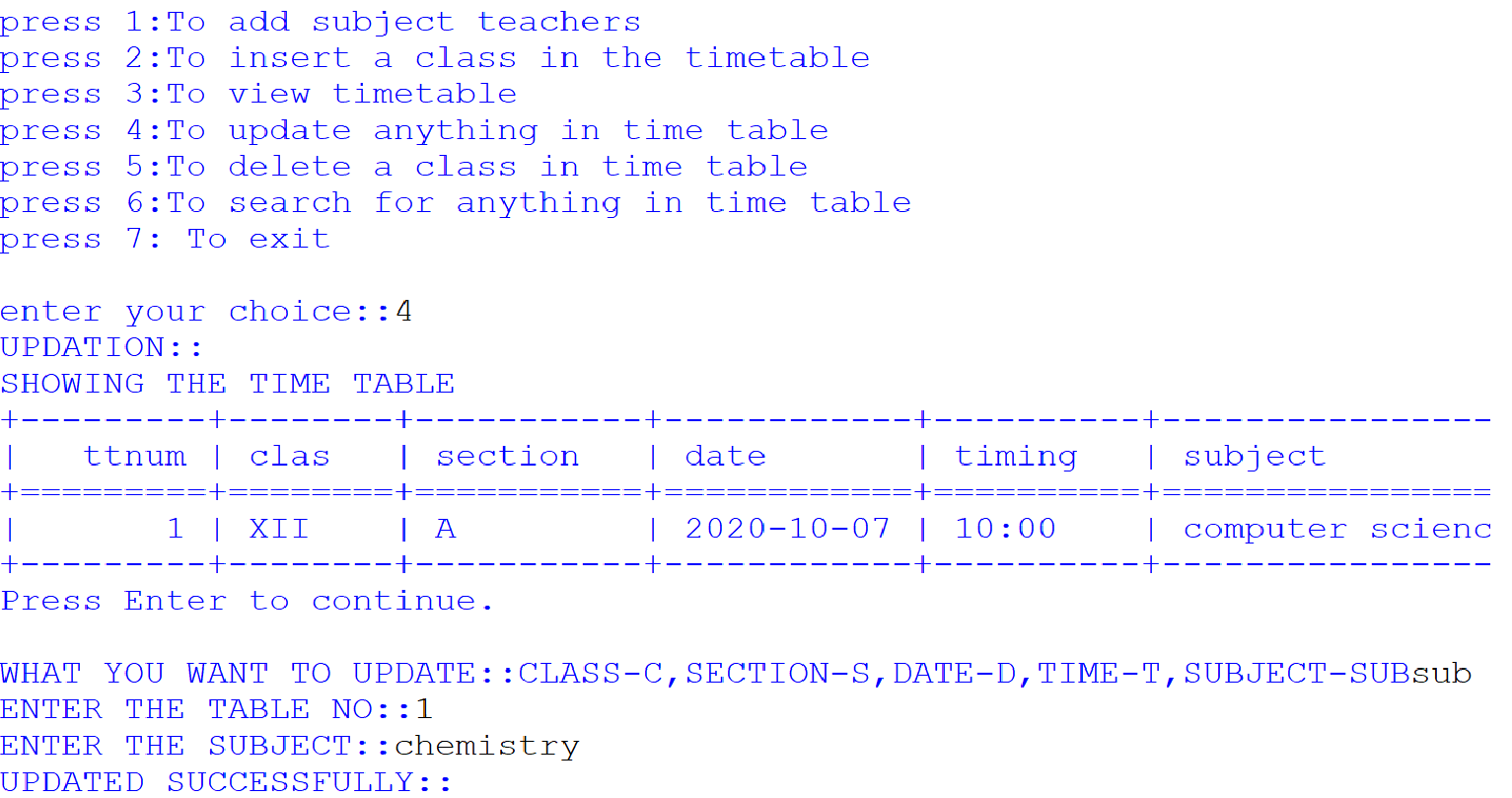
1 record is inserted successfully.

Press Enter to continue.

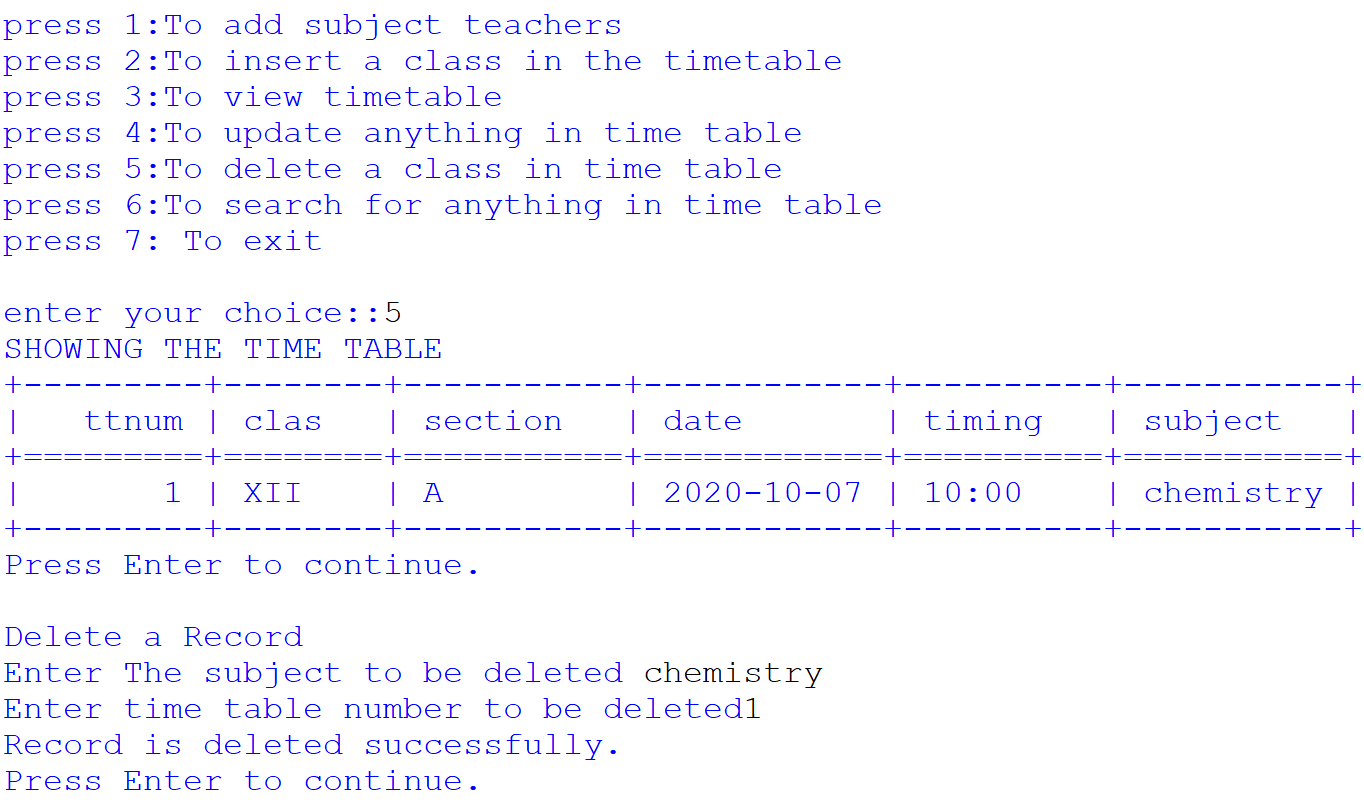
3.SHOW TABLE::

****

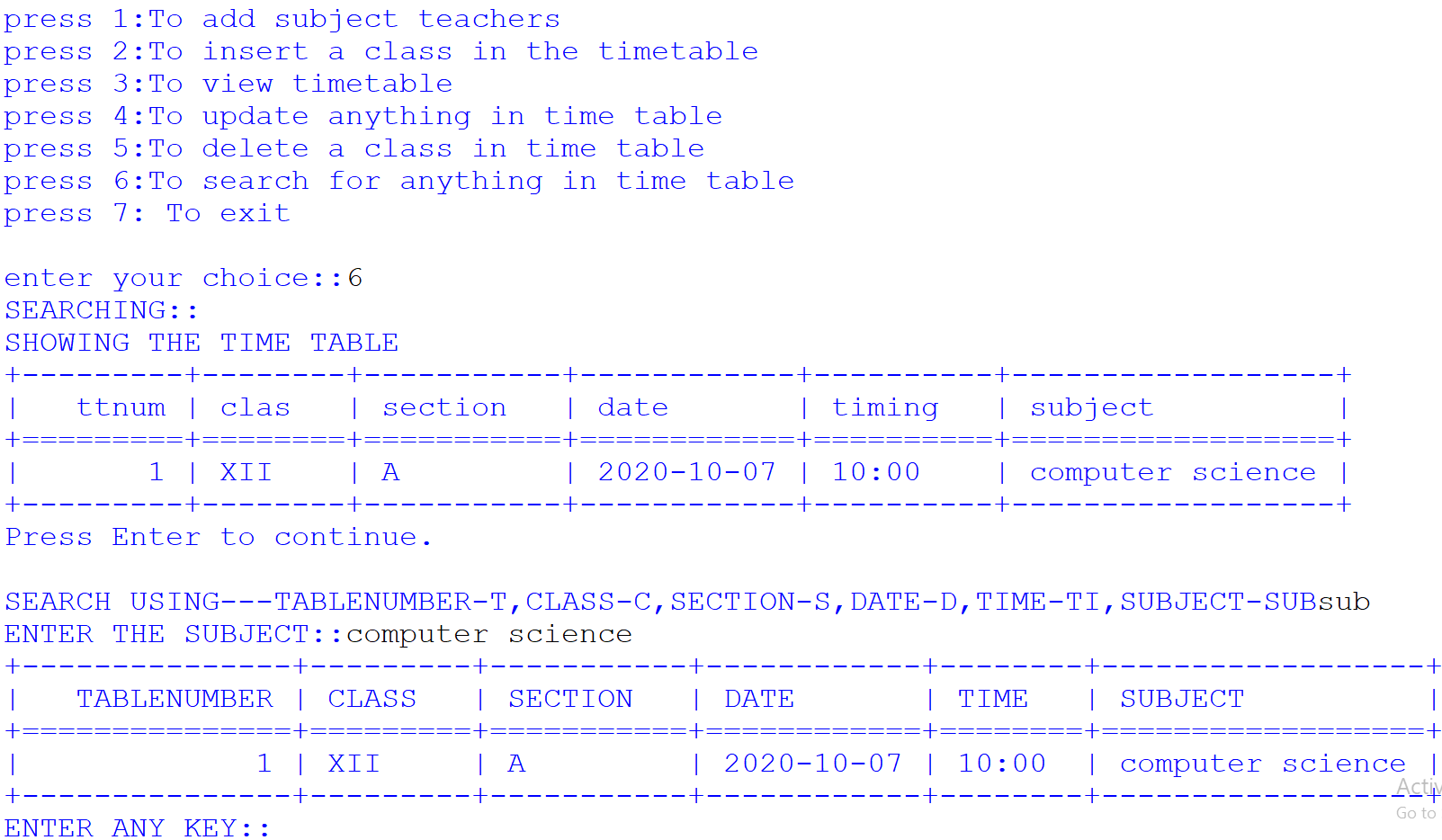
4.UPDATION::

****

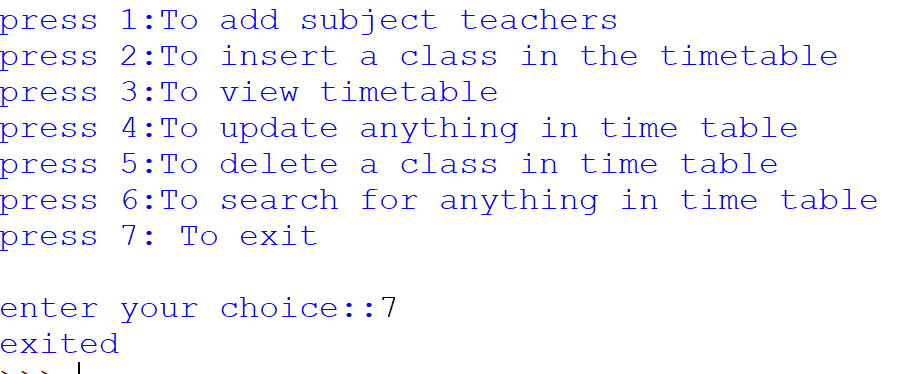
5.DELETION::

****

**6.SEARCH::**

****

**7.EXIT::**

****

**TEACHER’S VIEW::**

WHO ARE YOU? student,subject teacher or principal

if you are a STUDENT press 's',TEACHERpress't',PRINCIPAL press'p'::t

ENTER TEACHER NAME:teacher

PASSWORD:123

teacher access granted

press 1:To insert a class in the timetable

press 2:To deletion

press 3:To view timetable

press 4:To search for anything in time table

press 5:To exit

1.INSERTION::

enter your choice::1

Inserting a Record

Enter Class (in Roman letters) XII

Enter Section A

Enter Date (YYYY-MM-DD)2020-10-07

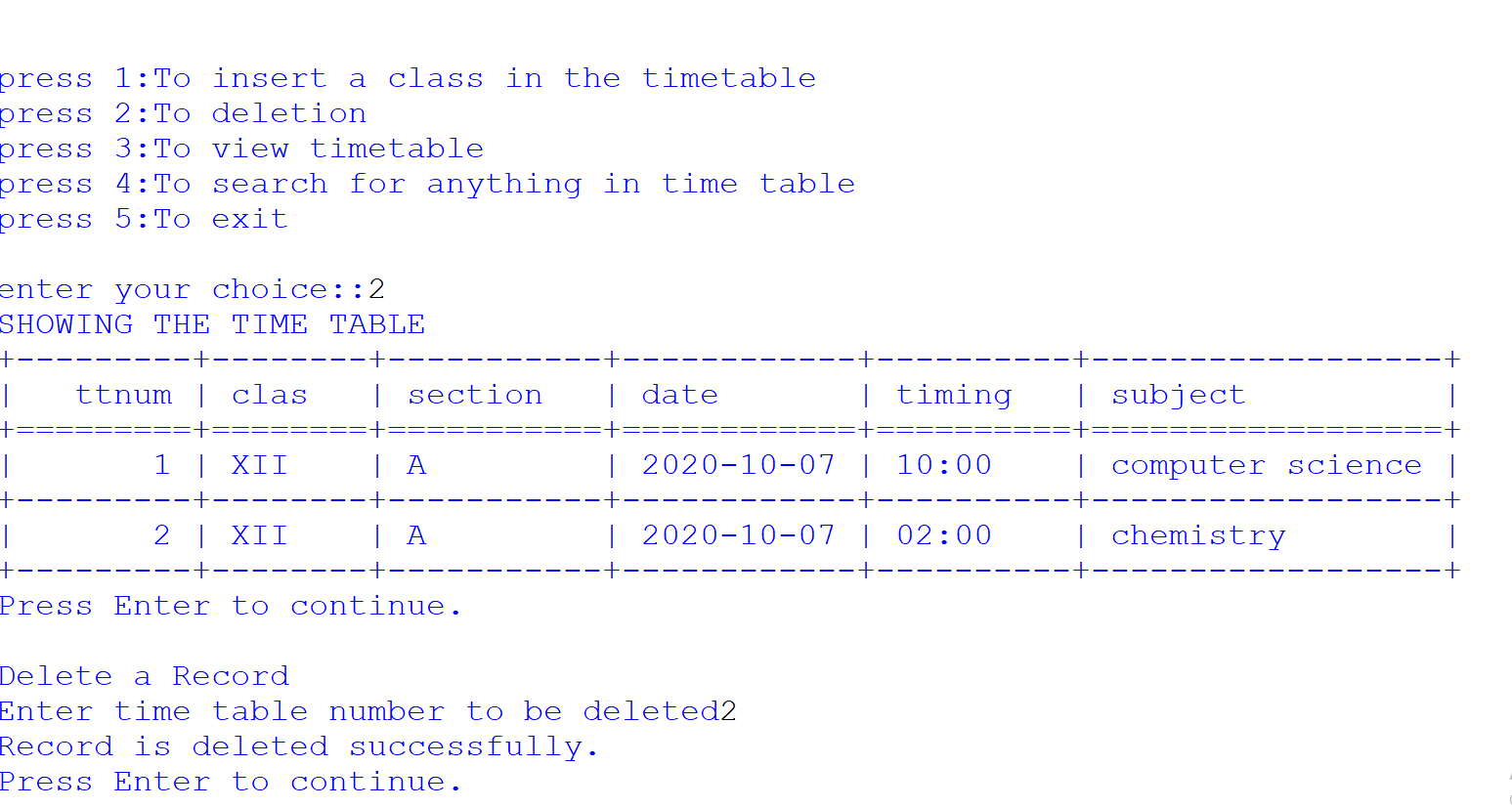
Enter The Timing (HH:MM) 02:00

Enter time table number3

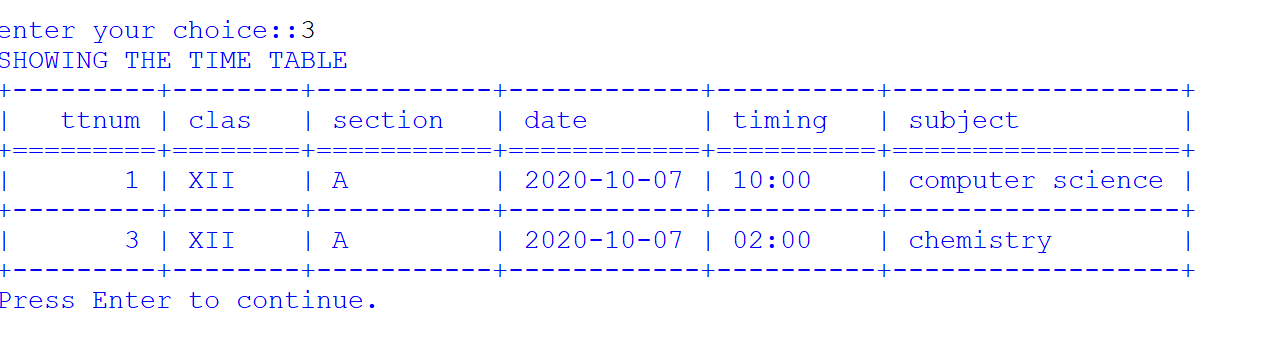
1 record is inserted successfully.

Press Enter to continue**.**

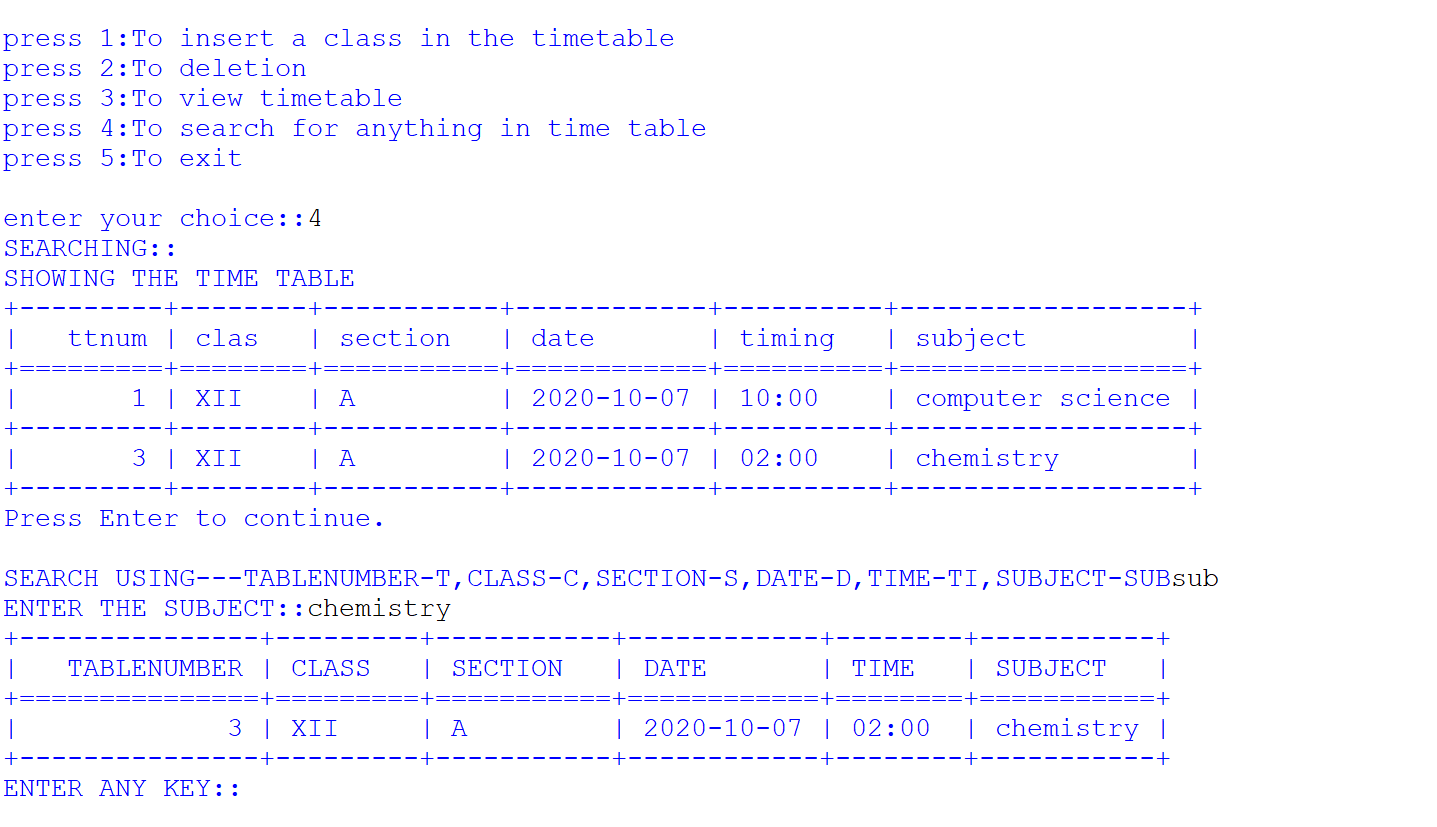
2.DELETION::

****

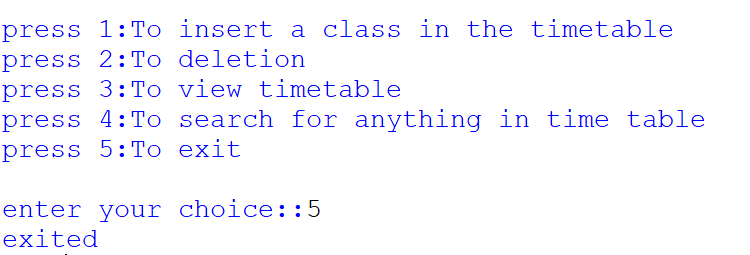
3.SHOW TABLE::

****

**4.SEARCH::**

****

**5.EXIT::**

****

**STUDENT’S VIEW::**

WHO ARE YOU? student,subject teacher or principal

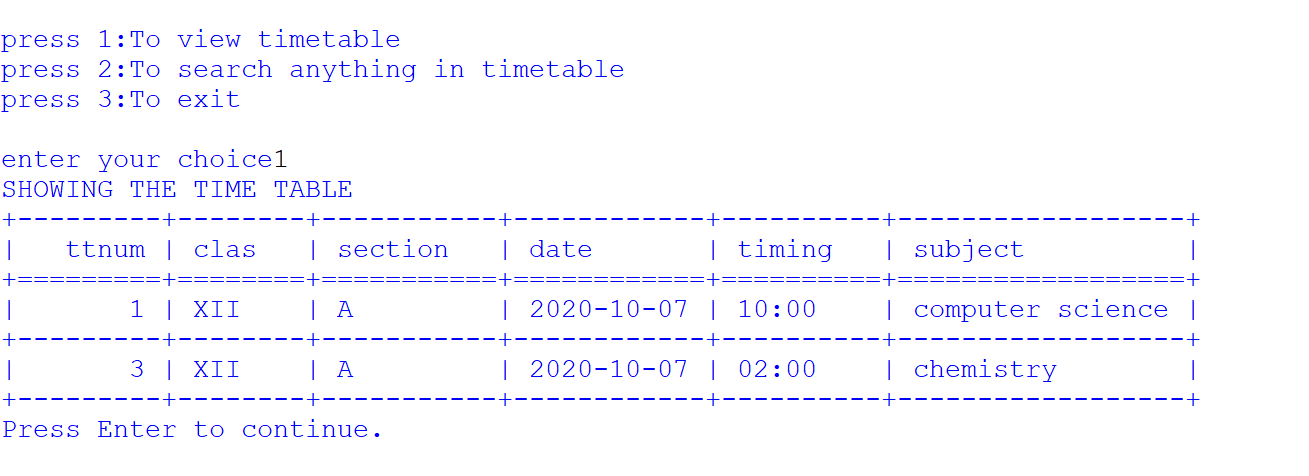
if you are a STUDENT press 's',TEACHER press't',PRINCIPAL press'p'::s

press 1:To view timetable

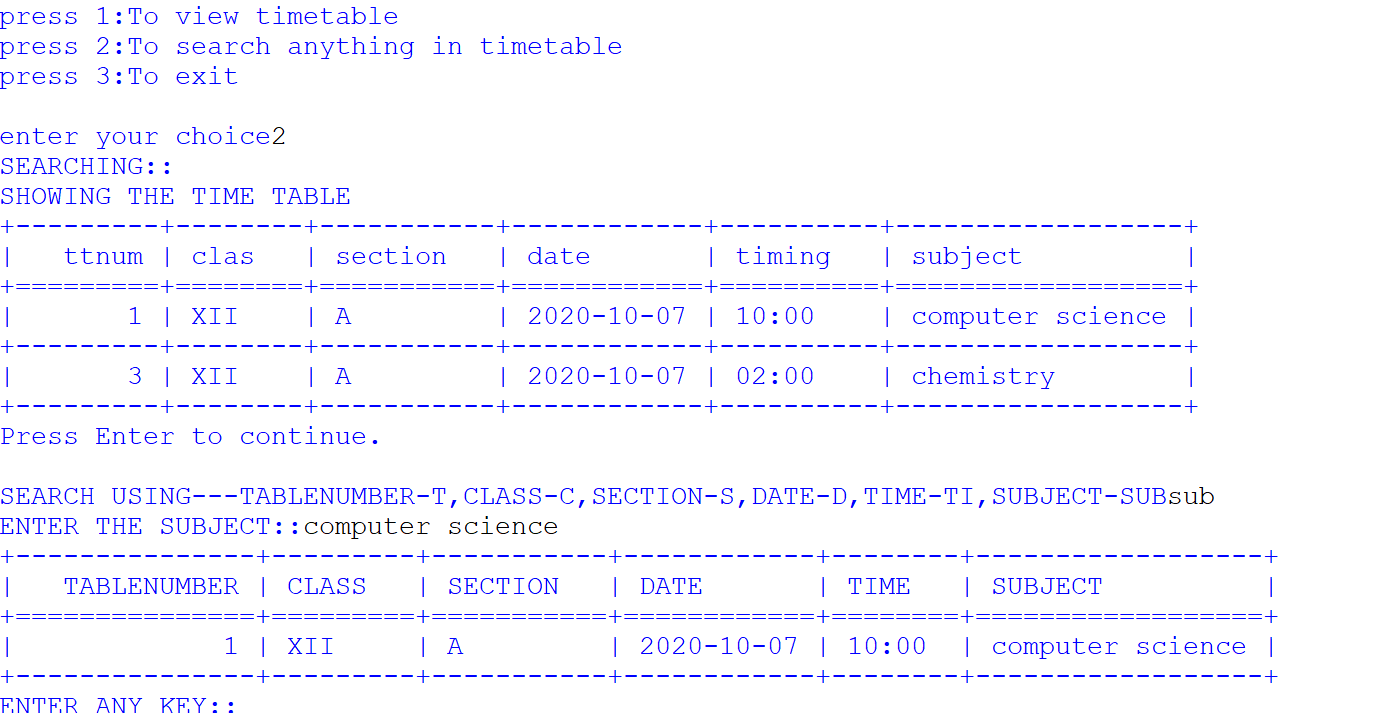
press 2:To search anything in timetable

press 3:To exit

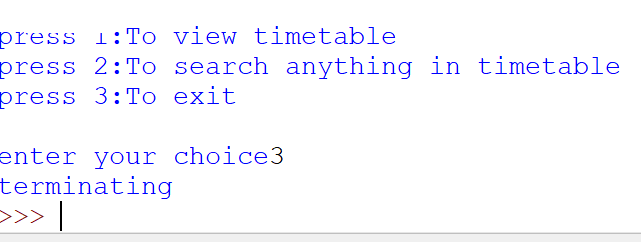
1.SHOW TABLE::

****

**2.SEARCH::**

****

**3.EXIT::**

****

*BIBLIOGRAPY*

* COMPUTER SCIENCE WITH PYTHON – sumitaArora
* GITHUB