**Internship Project**

**On**

**COMPUTER LAB ALLOCATOR (CLA)**

**Programming language used: Python**

***Presented By***

Roll No:18CE5004 Batch/Sr No: B3/15 Name: **Amisha Pupala**

Roll No:18CE5022 Batch-Sr No: B3/17 Name: **Neha Pandit**

Roll No:18CE5504 Batch/Sr No: C1/11 Name: **Nikita Bhoir**

Roll No:18CE8015 Batch/Sr No: C3/02 Name: **Saili Kakade**

Academic Year:2018-2019**Challenge 2: Computer Lab Allocator (CLA)**

**Introduction**

Your systems in labs are currently unenrolled. Write a code to accept name & roll number of a student as soon as he/she logs in.

**Subtask 1**

Create a python file, which, on execution, opens a menu asking if user is student or admin.

If the user is student, ask for name & roll no. (Later you can consider using a password based login)

(Create) Write these values with timestamp to a CSV file (prefer writing in a hidden CSV file)

(Read) Create an admin login (username: admin) who can login to your program and view all records

(Delete) Admin can delete entry of a particular student

(Update) Admin can change name/roll no. of a given student: Should ask if need to update name or roll no. (show previous name & roll no. for reference).

**Subtask 2**

Keep the CSV file password protected- Only your code should be able to access the file.

Only keep your executable (.pyc) file on the system, not .py file.

**Subtask 3**

Make it up in GUI.

**Subtask 4 (Tough)**

Use MySQL database for storing data instead of csv files: you’ll need to use python libraries for accessing MySQL server.

Use a centralised MySQL database: Install MySQL only on one server and make network calls via internet/intranet to the database server.

**Final**

Send a working video of any and all subtask(s) to avinash [at] outlook [dot] com - I’ll gladly feature it here on this page and invite you for my next hackathon tour. With a few tweaks, it might be your first step towards open source contribution.

**Code:**

#AdminLogin.py

from tkinter import \*

import datetime

import csv

from tkinter import messagebox

window3=Tk()

window3.title(&quot;Admin Login&quot;)

window3.geometry(&#39;700x500&#39;)

class AdminLoginP( object ):

def AdminLoginPage(self):

global e1

global e2

e1=Label(window3, text=&quot;Username:&quot;,font=(&quot;Arial Bold&quot;,10)).grid(row=0)

e2=Label(window3, text=&quot;Password:&quot;,font=(&quot;Arial Bold&quot;,10)).grid(row=1)

e1 = Entry(window3)

e2 = Entry(window3,show=&quot;\*&quot;)

e1.grid(row=0, column=1,padx=60, pady=2,sticky=NSEW)

e2.grid(row=1, column=1,padx=60, pady=2,sticky=NSEW)

B=Button(window3, text =&quot;Submit&quot;,command=lambda:self.retrieve\_input())

B.place(x=100, y=50)

mainloop()

def retrieve\_input(self):

uname=e1.get()

passwd=e2.get()

if(uname==&#39;admin&#39; and passwd==&#39;admin&#39;):

window3.destroy()

from AdminPage import AdminP

else:

messagebox.showerror(&quot;Error&quot;,&quot;Invalid Username or password&quot;)

w1=AdminLoginP()

w1.AdminLoginPage()

#AdminPage.py

from tkinter import \*

window4 = Tk()

class AdminP():

def view(self):

print(&#39;v&#39;)

window4.destroy()

from ViewRecord import ViewRec

#self.openRec()

def insert(self):

print(&#39;i&#39;)

window4.destroy()

from InsertRecord import InsertRec

InsertRec()

def update(self):

print(&#39;u&#39;)

window4.destroy()

from UpdateRecord import UpdateStudRec

def delete(self):

print(&#39;d&#39;)

window4.destroy()

from DeleteRecord import DeleteRec

def adminPg(self):

window4.title(&quot;Welcome to Computer Lab Alloctor&quot;)

window4.geometry(&#39;700x500&#39;)

page1text = Label(window4, text=&quot;This is page 1&quot;)

lbl=Label(window4,text=&#39;What operation do you want to perform?&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=140, y=110)

Vbtn=Button(window4, text=&quot; View Record &quot;,font=(&quot;Arial Bold&quot;,9), command=lambda:self.view())

Ibtn = Button(window4, text=&quot;Insert Record&quot;,font=(&quot;Arial Bold&quot;,9), command=lambda:self.insert())

Ubtn =Button(window4, text=&quot;Update Record&quot;,font=(&quot;Arial Bold&quot;,9),

command=lambda:self.update())

Dbtn=Button(window4, text=&quot;Delete Record&quot;,font=(&quot;Arial Bold&quot;,9),

command=lambda:self.delete())

Vbtn.place(x=210, y=180)

Ibtn.place(x=390, y=180)

Ubtn.place(x=210, y=240)

Dbtn.place(x=390, y=240)

mainloop()

ap=AdminP()

ap.adminPg()

#DeleteRecord.py

from tkinter import \*

from tkinter import messagebox

import csv

window6=Tk()

window6.title(&quot;Computer Lab Alocator&quot;)

window6.geometry(&#39;700x500&#39;)

class DeleteRec():

def DelDetail(self):

&#39;&#39;&#39;window6=Tk()

window6.title(&quot;Computer Lab Alocator&quot;)

window6.geometry(&#39;700x500&#39;)&#39;&#39;&#39;

global e1

l1=Label(window6, text=&quot;Enter the roll number of the student whose data you want to

delete&quot;,font=(&quot;Arial Bold&quot;,10))

e1 = Text(window6, height=1, width=20)

lbl=Label(window6,text=&#39;Delete a record:&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=160, y=110)

l1.place(x=160, y=170)

e1.place(x=280, y=200)

B=Button(window6, text =&quot;Submit&quot;,command=lambda:self.DelRec())

B.place(x=320, y=250)

B2=Button(window6, text =&quot; Go Back &quot;,command=lambda:self.GoBack())

B2.place(x=560, y=410)

mainloop()

def DelRec( self ):

val=e1.get(&quot;1.0&quot;,END)

print(val)

with open(&#39;Library.csv&#39;, &#39;r&#39;) as inp, open(&#39;first\_edit.csv&#39;, &#39;w&#39;) as out:

writer = csv.writer(out)

for row in csv.reader(inp):

if(len(row)&gt;0):

if(row[1]!=str(val)):

writer.writerow(row)

with open(&#39;first\_edit.csv&#39;, &#39;r&#39;) as inp, open(&#39;Library.csv&#39;, &#39;w&#39;) as out:

writer = csv.writer(out)

for row in csv.reader(inp):

if(len(row)&gt;0):

writer.writerow(row)

messagebox.shonfo(title=&quot;Successful&quot;, message=&quot;Record is deleted successfully&quot;)

def GoBack(self):

window6.destroy()

from AdminPage import AdminP

d1=DeleteRec()

d1.DelDetail()

#FirstPage.py

from tkinter import \*

#from AdminLogin import \*

window1=Tk()

window1.title(&quot;Welcome to Computer Lab alloctor&quot;)

window1.geometry(&#39;700x500&#39;)

class FP():

def FirstP(self):

global var

var = StringVar(window1)

var.set(&quot;~~Select here~~&quot;)

lbl=Label(window1,text=&#39;Select Your Designation&#39;,font=(&quot;Arial Bold&quot;,10))

lbl.place(x=90, y=155)

drop\_menu = OptionMenu(window1, var, &quot;Admin&quot;, &quot;Student&quot;,command=self.GetOption)

drop\_menu.grid(row=0, column=0)

drop\_menu.place(x=290, y=150)

B1 = Button(window1, text =&quot;Submit&quot;, command=lambda:self.NextPage())

B1.place(x=230, y=193)

window1.mainloop()

def GetOption(self,event):

global chosen\_option

value=var

chosen\_option = str(value.get())

print(chosen\_option)

def NextPage(self):

if(chosen\_option==&quot;Admin&quot;):

window1.destroy()

from AdminLogin import AdminLoginP

else:

window1.destroy()

from StudentLogin import StudentLogin

f1=FP()

f1.FirstP()

#InsertRecord.py

from pprint import pprint

from tkinter import \*

import datetime

from tkinter import messagebox

import csv

window5=Tk()

window5.title(&quot;Student Details&quot;)

window5.geometry(&#39;700x500&#39;)

class InsertRec( object ):

def \_\_init\_\_( self ):

self.students = {}

def InsRec(self):

global e1

global e2

l1=Label(window5, text=&quot;Student Name:&quot;,font=(&quot;Arial Bold&quot;,10))

l2=Label(window5, text=&quot;Roll Number:&quot;,font=(&quot;Arial Bold&quot;,10))

e1 = Text(window5, height=1, width=20)

e2 = Text(window5, height=1, width=20)

lbl=Label(window5,text=&#39;Enter the following Details of the Student&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=160, y=110)

l1.place(x=160, y=170)

e1.place(x=300, y=170)

l2.place(x=160, y=220)

e2.place(x=300, y=220)

B=Button(window5, text =&quot;Submit&quot;,command=lambda:self.retrieve\_input())

B.place(x=270, y=270)

B2=Button(window5, text =&quot; Go Back &quot;,command=lambda:self.GoBack())

B2.place(x=560, y=410)

mainloop()

def retrieve\_input(self):

#global name=e1.get()

name=e1.get(&quot;1.0&quot;,END)

rollno=e2.get(&quot;1.0&quot;,END)

date=datetime.datetime.now().time()

print(name)

print(rollno)

print(date)

with open(&#39;Library.csv&#39;,&#39;a&#39;) as out:

csvWriter = csv.writer(out)

row = (name,rollno,date)

csvWriter.writerow(row)

#self.writeOut()

with open(&#39;Library.csv&#39;, &#39;r&#39;) as inp, open(&#39;first\_edit.csv&#39;, &#39;w&#39;) as out:

writer = csv.writer(out)

for row in csv.reader(inp):

if(len(row)&gt;0):

writer.writerow(row)

messagebox.showinfo(title=&quot;Successful&quot;, message=&quot;Record is Inserted successfully&quot;)

def GoBack(self):

window5.destroy()

from AdminPage import AdminP

&#39;&#39;&#39;def writeOut( self ):

with open(&#39;Library.csv&#39;,&#39;w&#39;) as out:

csvWriter = csv.writer(out)

csvWriter.writerow((&#39;ID&#39;,&#39;name&#39;,&#39;rollno&#39;,&#39;date&#39;))

for item in self.students:

b = self.students[item][0]

name = self.students[item][1]

rollno = self.students[item][2]

date=self.students[item][3]

row = (b.id,b.name,b.rollno,b.date)

csvWriter.writerow(row)&#39;&#39;&#39;

ir= InsertRec()

ir.InsRec()

&#39;&#39;&#39;studentLogin.studInfo(&#39;1&#39;,&#39;aaa&#39;,&#39;1111&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;2&#39;,&#39;bbb&#39;,&#39;2222&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;3&#39;,&#39;ccc&#39;,&#39;3333&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;4&#39;,&#39;ddd&#39;,&#39;4444&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;5&#39;,&#39;eee&#39;,&#39;5555&#39;,datetime.datetime.now().time())&#39;&#39;&#39;

pprint(ir.students)

#studentLogin.writeOut()

#stud1.py

from tkinter import \*

import csv

def studLoginPage():

window2=Tk()

window2.title(&quot;Student Details&quot;)

window2.geometry(&#39;700x500&#39;)

global e1

global e2

e1=Label(window2, text=&quot;Student Name:&quot;,font=(&quot;Arial Bold&quot;,10)).grid(row=0)

e2=Label(window2, text=&quot;Roll Number:&quot;,font=(&quot;Arial Bold&quot;,10)).grid(row=1)

e1 = Entry(window2)

e2 = Entry(window2)

e1.grid(row=0, column=1,padx=60, pady=2,sticky=NSEW)

e2.grid(row=1, column=1,padx=60, pady=2,sticky=NSEW)

B=Button(window2, text =&quot;Submit&quot;,command=lambda: retrieve\_input())

B.place(x=100, y=50)

mainloop()

def retrieve\_input():

global inputValue

inputValue=[e1.get(),e2.get()]

print(inputValue)

def dict():

global info

info = {}

srno=0

info[srno]=[e1.get(),e2.get()]

def writeOut(self):

with open(&#39;Compuer\_Allocated.csv&#39;,&#39;w&#39;) as out:

csvWriter = csv.writer(out)

for item in self.info:

b = books[item][0]

name = books[item][1]

rollno = books[item][2]

row = (name,rollno)

csvWriter.writerow(row)

s1=studLoginPage()

writeOut()

#Student.py

class student( object ):

def \_\_init\_\_( self,id ,name,rollno,date):

self.id = id

self.name=name

self.rollno = rollno

self.date=date

#StudentLogin.py

from Student import student

from pprint import pprint

from tkinter import messagebox

from tkinter import \*

import datetime

import csv

class StudentLogin( object ):

def \_\_init\_\_( self ):

self.students = {}

def studLoginPage(self):

window2=Tk()

window2.title(&quot;Student Details&quot;)

window2.geometry(&#39;700x500&#39;)

global e1

global e2

l1=Label(window2, text=&quot;Student Name:&quot;,font=(&quot;Arial Bold&quot;,10))

l2=Label(window2, text=&quot;Roll Number:&quot;,font=(&quot;Arial Bold&quot;,10))

e1 = Text(window2, height=1, width=20)

e2 = Text(window2, height=1, width=20)

lbl=Label(window2,text=&#39;Enter the Following Details:&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=160, y=110)

l1.place(x=160, y=170)

e1.place(x=300, y=170)

l2.place(x=160, y=220)

e2.place(x=300, y=220)

B=Button(window2, text =&quot;Submit&quot;,command=lambda:self.retrieve\_input())

B.place(x=270, y=270)

mainloop()

def retrieve\_input(self):

#global name=e1.get()

name=e1.get(&quot;1.0&quot;,END)

rollno=e2.get(&quot;1.0&quot;,END)

date=datetime.datetime.now().time()

print(name)

print(rollno)

print(date)

with open(&#39;Library.csv&#39;,&#39;a&#39;) as out:

csvWriter = csv.writer(out)

row = (name,rollno,date)

csvWriter.writerow(row)

#self.writeOut()

messagebox.showinfo(title=&quot;Successful&quot;, message=&quot;Record is Inserted successfully!&quot;)

&#39;&#39;&#39;def writeOut( self ):

with open(&#39;Library.csv&#39;,&#39;w&#39;) as out:

csvWriter = csv.writer(out)

csvWriter.writerow((&#39;ID&#39;,&#39;name&#39;,&#39;rollno&#39;,&#39;date&#39;))

for item in self.students:

b = self.students[item][0]

name = self.students[item][1]

rollno = self.students[item][2]

date=self.students[item][3]

row = (b.id,b.name,b.rollno,b.date)

csvWriter.writerow(row)&#39;&#39;&#39;

studentLogin= StudentLogin()

studentLogin.studLoginPage()

&#39;&#39;&#39;studentLogin.studInfo(&#39;1&#39;,&#39;aaa&#39;,&#39;1111&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;2&#39;,&#39;bbb&#39;,&#39;2222&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;3&#39;,&#39;ccc&#39;,&#39;3333&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;4&#39;,&#39;ddd&#39;,&#39;4444&#39;,datetime.datetime.now().time())

studentLogin.studInfo(&#39;5&#39;,&#39;eee&#39;,&#39;5555&#39;,datetime.datetime.now().time())&#39;&#39;&#39;

#pprint(studentLogin.students)

#studentLogin.writeOut()

#StudentLogin2.py

from Student import student

from pprint import pprint

from tkinter import \*

import datetime

import csv

class StudentLogin( object ):

def \_\_init\_\_( self ):

self.students = []

def retrieve\_input():

name=e1.get()

rollno=e2.get()

def studInfo( self,id,name,rollno,date):

Student = student( id,name ,rollno,date)

self.students[id] = [Student,&#39;&#39;,&#39;&#39;,&#39;&#39;,&#39;&#39;]

def writeOut( self ):

with open(&#39;Library.csv&#39;,&#39;w&#39;) as out:

csvWriter = csv.writer(out)

csvWriter.writerow((&#39;ID&#39;,&#39;name&#39;,&#39;rollno&#39;,&#39;date&#39;))

for item in len(self.students):

b = self.students[item][0]

name = self.students[item][1]

rollno = self.students[item][2]

date=self.students[item][3]

row = (b.id,b.name,b.rollno,b.date)

csvWriter.writerow(row)

&#39;&#39;&#39;with open(&#39;Library.csv&#39;, &#39;w&#39;) as out:

csvWriter = csv.writer(out)

csvWriter.writerow((&#39;ID&#39;,&#39;Title&#39;,&#39;Author&#39;, &#39;Price&#39;))

for item in self.students:

b = self.students[item][0]

rn = self.students[item][1]

doi = self.students[item][2]

row = (id, name, rollno, date)

csvWriter.writerow(row)

#id=len(self.students)+1&#39;&#39;&#39;

&#39;&#39;&#39;with open(&#39;Library.csv&#39;,&#39;a&#39;) as out:

csvWriter = csv.writer(out)

csvWriter.writerow((&#39;ID&#39;,&#39;name&#39;,&#39;rollno&#39;,&#39;date&#39;))

for item in self.students:

b = self.students[item][0]

name = self.students[item][1]

rollno = self.students[item][2]

date=self.students[item][3]

row = (b.id,b.name,b.rollno,b.date)

csvWriter.writerow(row)

#csvWriter.writerow(self.studInfo(id,name,rollno,date))&#39;&#39;&#39;

def studInfo( self,id,name,rollno,date):

Student = student( id,name ,rollno,date)

self.students[id] = [Student,&#39;&#39;,&#39;&#39;,&#39;&#39;]

studentLogin= StudentLogin()

studentLogin.studInfo(1,&#39;aaa&#39;,&#39;1111&#39;,datetime.datetime.now().time())

studentLogin.studInfo(2,&#39;bbb&#39;,&#39;2222&#39;,datetime.datetime.now().time())

studentLogin.studInfo(3,&#39;ccc&#39;,&#39;3333&#39;,datetime.datetime.now().time())

studentLogin.studInfo(4,&#39;ddd&#39;,&#39;4444&#39;,datetime.datetime.now().time())

studentLogin.studInfo(5,&#39;eee&#39;,&#39;5555&#39;,datetime.datetime.now().time())

pprint(studentLogin.students)

studentLogin.writeOut()

#test1.py

class StudentLogin( object ):

def \_\_init\_\_( self ):

self.students = {}

def studLoginPage(self):

window2=Tk()

window2.title(&quot;Student Details&quot;)

window2.geometry(&#39;700x500&#39;)

global e1

global e2

l1=Label(window2, text=&quot;Student Name:&quot;,font=(&quot;Arial Bold&quot;,10))

l2=Label(window2, text=&quot;Roll Number:&quot;,font=(&quot;Arial Bold&quot;,10))

e1 = Text(window2, height=1, width=20)

e2 = Text(window2, height=1, width=20)

lbl=Label(window2,text=&#39;Enter the Following Details:&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=160, y=110)

l1.place(x=160, y=170)

e1.place(x=300, y=170)

l2.place(x=160, y=220)

e2.place(x=300, y=220)

B=Button(window2, text =&quot;Submit&quot;,command=lambda:self.retrieve\_input())

B.place(x=270, y=270)

mainloop()

def retrieve\_input(self):

#global name=e1.get()

name=e1.get(&quot;1.0&quot;,END)

rollno=e2.get(&quot;1.0&quot;,END)

date=datetime.datetime.now().time()

print(name)

print(rollno)

print(date)

#UpdateRecord.py

from tkinter import \*

from tkinter import messagebox

import csv

window6=Tk()

window6.title(&quot;Computer Lab Allocator&quot;)

window6.geometry(&#39;700x500&#39;)

class UpdateStudRec():

def UpdateDetail(self):

&#39;&#39;&#39;window6=Tk()

window6.title(&quot;Computer Lab Allocator&quot;)

window6.geometry(&#39;700x500&#39;)&#39;&#39;&#39;

global e1

global e2

global e3

l1=Label(window6, text=&quot;Enter the roll number of the student whose data you want to

delete:&quot;,font=(&quot;Arial Bold&quot;,10))

e1 = Text(window6, height=1, width=20)

lbl=Label(window6,text=&#39;Update Record:&#39;,font=(&quot;Arial Bold&quot;,17))

lbl.place(x=250, y=65)

l1.place(x=165, y=120)

e1.place(x=280, y=150)

lnew=Label(window6, text=&quot;Enter the new Details:&quot;,font=(&quot;Arial Bold&quot;,13))

lnew.place(x=270, y=200)

l2=Label(window6, text=&quot;Enter the new Name:&quot;,font=(&quot;Arial Bold&quot;,10))

e2 = Text(window6, height=1, width=20)

l2.place(x=180, y=250)

e2.place(x=330, y=250)

l3=Label(window6, text=&quot;Enter the new Date:&quot;,font=(&quot;Arial Bold&quot;,10))

e3 = Text(window6, height=1, width=20)

l3.place(x=180, y=290)

e3.place(x=330,y=290)

B=Button(window6, text =&quot;Submit&quot;,command=lambda:self.UpdateRec())

B.place(x=300, y=330)

B2=Button(window6, text =&quot; Go Back &quot;,command=lambda:self.GoBack())

B2.place(x=560, y=410)

mainloop()

def UpdateRec( self ):

val1=e1.get(&quot;1.0&quot;,END)

val2=e2.get(&quot;1.0&quot;,END)

val3=e3.get(&quot;1.0&quot;,END)

print(val1)

print(val2)

print(val3)

with open(&#39;Library.csv&#39;, &#39;r&#39;) as inp, open(&#39;first\_edit.csv&#39;, &#39;w&#39;) as out:

writer = csv.writer(out)

for row in csv.reader(inp):

if(len(row)&gt;0):

if(row[1]==str(val1)):

updatedRec = [val2, val1,val3]

writer.writerow(updatedRec)

else:

writer.writerow(row)

with open(&#39;first\_edit.csv&#39;, &#39;r&#39;) as inp, open(&#39;Library.csv&#39;, &#39;w&#39;) as out:

writer = csv.writer(out)

for row in csv.reader(inp):

if(len(row)&gt;0):

writer.writerow(row)

messagebox.showinfo(title=&quot;Successful&quot;, message=&quot;Record is Updated Successfully!&quot;)

def GoBack(self):

window6.destroy()

from AdminPage import AdminP

up=UpdateStudRec()

up.UpdateDetail()

#ViewRecord.py

import webbrowser

class ViewRec():

def openRec(self):

webbrowser.open(&quot;Library.csv&quot;)

vr=ViewRec()

vr.openRec()