National Public School, HSR Layout

**Students’ Academic Guide**

Computer Science Project

Vikas.K and Mahit Nandan

Year 2019-20

Grade 11

ACKNOWLEDGEMENT

We are extremely grateful to Mrs. Sumathi Kumaresan, for her able guidance and useful suggestions, which helped us in completing the project within the allotted deadline. We would also thank our respected school Principal, Mrs. Shefali Tyagi for proving us with the infrastructure for doing this project. Most importantly, we would like to thank our friends and classmates for lending their support and help in the successful completion of this project.

Table of Contents

**S.No Topic Page Number**

|  |  |  |
| --- | --- | --- |
| **1** | Overview of Python | 3 |
| 2 | Project Synopsis | 4 |
| 3 | System Requirements | 4 |
| 4 | Instruction Manual | 5 |
| 5 | UI Components and functions used | 6 |
| 6 | Program Code | 8 |
| 7 | Screenshots | 31 |
| 9 | Limitations | 34 |
| 10 | Further improvements | 34 |
| 11 | Bibliography | 34 |

Overview of Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python's features include −

* A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh
* Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* Scalable − Python provides a better structure and support for large programs than shell scripting.

Project Synopsis

Students’ Academic Companion is a project meant for Science Students studying in grade 11 and 12.The project enables student to practice chapters in Physics and Chemistry and provides the user with all theory, practice problems, solved examples and mock tests associated with every chapter in Physics and Chemistry

In 11th and 12th grade .The application will also allow students to keep a track of all chapters he or she has already finished and make notes in any chapter which can be visited whenever required which will be helpful during revision.

The project will include using several aspects of python such as file handling and basic

I/O.

System Requirements

**Hardware**:

256 MB RAM and a 1.5 GHz processor

**Software:**

Operating System-Compatible with Windows XP, Windows 7,Windows 8 and Windows 10-32 bit as well as 64 but versions

Instruction Manual

1. The first window in a greeting window, asking the user whether he/she is a new student or an already existing user. Two options will be displayed:

* Yes
* No

1. Choosing Yes, a window will open asking for :

* Name
* Password

1. Choosing No, a window will open asking for Name, Password. If the entered username or password doesn’t match with existing one. A window will open throwing up an error.
2. After entering the username and password successfully a new window will open featuring two options:

* Physics
* Chemistry

1. On Choosing first option a series of options with the names of Physics Chapters of Grade 11 and Grade 12.
2. On Choosing second option a series of options with the names of Chemistry Chapters of Grade 11 and Grade 12.
3. Choosing any of the chapters will open a window with the following option

* Assignments
* Notes
* Tests
* Theory

1. Choosing the first option will open assignments pdf related to the topic.
2. Choosing the second option will open a notepad dedicated to the chosen topic where the user can fill their own notes.
3. Choosing the third option will open assignments pdf related to the topic.
4. Choosing the fourth option will open assignments pdf related to the topic.
5. The user can continue to choose different topic from Physics and Chemistry at the same time.

UI Components and Functions Used

UI Components:

Windows.geometry : Positioning of the window.

Label: A **Label** is a Tkinter Widget class, which is used to display text or an image.

Grid : The master widget is split into a number of rows and columns,

Button: The **Button** widget is used to add **buttons** in a **Python** application.

Functions

def setpassword()

Storing the password into a file.

def makechapter()

Function to create Folder of a subject in the User's Folder

def makechapters(a,subject)

Function to create All Chapters of Physics and Chemistry

def if\_no() and def norandom():

Data to be presented if the user has used the application before.

def Chemistry():

Lists the options if user chooses chemistry button.

def fun():

Opens the window of the chapter which is selected.

def run1():

Opens theory of the chosen chemistry chapter

def run2():

Opens assignments of the chosen chemistry chapter

def run3():

Opens Tests of the chosen chemistry chapter

def notes():

Opens a note pad for each chosen chapter where user can maintain their own notes.

def Physics():

Lists the options if user chooses physics button

def if\_yes() and def yesrandom():

Opens window asking username and password of new user.

Program Code

from tkinter import \*

import os

#Function to create password for new user

def setpassword(password,a):

f = open(a + "\\Password.txt" ,'a')

f.write(password)

f.close()

#Function to create Folder of a subject in the User's Folder

def makechapter(a,subject,chapter):

fpath = a + "\\" + subject

os.mkdir(fpath + "\\" + chapter)

#Function to create All Chapters of Physics and Chemistry

def makechapters(a,subject):

#Making chapters in Chemistry

if subject == "Chemistry":

makechapter(a,subject,"SBCOC")

makechapter(a,subject,"Structure of atom")

makechapter(a,subject,"Periodic Properties")

makechapter(a,subject,"Chemical Bonding")

makechapter(a,subject,"States of Matter")

makechapter(a,subject,"Thermodynamics")

makechapter(a,subject,"Equilibrium")

makechapter(a,subject,"Redox Reactions")

makechapter(a,subject,"Hydrogen")

makechapter(a,subject,"S Block Elements")

makechapter(a,subject,"P Block Elements")

makechapter(a,subject,"GOC")

makechapter(a,subject,"Hydrocarbons")

makechapter(a,subject,"Environmental Chemistry")

#Making chapters in Physics

if subject == "Physics":

makechapter(a,subject,"Units and Measurement")

makechapter(a,subject,"Motion in Straight Line")

makechapter(a,subject,"Motion in a Plane")

makechapter(a,subject,"Laws of Motion")

makechapter(a,subject,"WPE")

makechapter(a,subject,"System of Particles")

makechapter(a,subject,"Gravitation")

makechapter(a,subject,"Solids")

makechapter(a,subject,"Fluids")

makechapter(a,subject,"Thermal Properties of Matter")

makechapter(a,subject,"Thermodynamics")

makechapter(a,subject,"Kinetic Theory")

makechapter(a,subject,"Oscillations")

makechapter(a,subject,"Waves")

#If the user has used application before(No is pressed)

def if\_no():

nowindow=Tk()

nowindow.title("Entry:")

username=StringVar()

password=StringVar()

nowindow.geometry('350x200')

l2=Label(nowindow,text="Please enter your username")

uentry=Entry(nowindow,textvariable=username)

uentry.grid(row=0,column=1)

l3=Label(nowindow,text="Please enter your password")

pentry=Entry(nowindow,textvariable=password,show="\*")

pentry.grid(row=1,column=1)

l2.grid(row=0,column=0)

l3.grid(row=1,column=0)

def norandom():

password = pentry.get()

stu=uentry.get()

made = os.getcwd() + "\\our"

directory = os.getcwd() + "\\" + stu

files = os.listdir(os.getcwd())

#Checking if folder exists

if stu not in files:

window2 = Tk()

window2.geometry('250x50')

window2.title("Folder not Found")

l1= Label(window2, text="Folder doesn't exist.")

l2 = Label(window2, text= "Please close this window and enter again.")

l1.grid(row=0,column=0)

l2.grid(row=1,column=0)

else:

#Checking password

f2 = open(os.getcwd() + "\\" + stu + "\\Password.txt","r")

pas = f2.read()

if pas == password:

porc = Tk()

porc.title("Phy or Chem:")

porc.geometry('400x100')

label = Label(porc,text="Physics or Chemistry?")

label.grid(row=0,column=0)

#If user wants to study chemistry

def Chemistry():

directory = os.getcwd() + "\\" + stu

chemistry = Tk()

chemistry.title("Chemistry")

def fun(a):

Sb=Tk()

Sb.title("Entry:")

directory = os.getcwd() + "\\" + uentry.get()

l1= Label(Sb, text="Theory")

l2= Label(Sb, text="Assignments")

l3= Label(Sb, text="Tests")

l4= Label(Sb, text="Notes")

l1.grid(column=0,row=0)

l2.grid(column=3,row=0)

l3.grid(column=6,row=0)

l4.grid(column=9,row=0)

#Theory PDFs

def run1():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Theory\\File.pdf"

os.startfile(path)

file1=Button(Sb,text="Theory1",command=run1)

#Assignment PDFs

def run2():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Assignments\\File.pdf"

os.startfile(path)

file2 = Button(Sb,text="Assignment1",command=run2)

#Test PDFs

def run3():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Tests\\File.pdf"

os.startfile(path)

file3 = Button(Sb,text="Test1",command=run3)

file1.grid(column=0,row=2)

file2.grid(column=3,row=2)

file3.grid(column=6,row=2)

#Wants to write his own notes

def notes():

name = "notes.txt"

path = uentry.get() + "\\Chemistry\\"+a+"\\"+name

f = open(path,'a')

f.close()

os.startfile(path)

file4=Button(Sb,text='notes',command=notes)

file4.grid(column=9,row=2)

#Different Chapters

b1 = Button(chemistry,text="SBCOC",width=20,command=lambda : fun("SBCOC"))

b1.grid(column=0, row=0)

b2=Button(chemistry,text="Structure of atom",width=20,command=lambda:fun("Structure of Atom"))

b2.grid(column=1, row=0)

b3=Button(chemistry,text="Periodic Properties",width=20,command=lambda:fun("Periodic Properties"))

b3.grid(column=2, row=0)

b4=Button(chemistry,text="Chemical Bonding",width=20,command=lambda:fun("Chemical Bonding"))

b4.grid(column=0, row=1)

b5=Button(chemistry,text="States of Matter",width=20,command=lambda:fun("States of Matter"))

b5.grid(column=1, row=1)

b6=Button(chemistry,text="Thermodynamics",width=20,command=lambda:fun("Thermodynamics"))

b6.grid(column=2, row=1)

b7=Button(chemistry,text="Equilibrium",width=20,command=lambda:fun("Equilibrium"))

b7.grid(column=0, row=2)

b8=Button(chemistry,text="Redox Reactions",width=20,command=lambda:fun("Redox Reactions"))

b8.grid(column=1, row=2)

b9=Button(chemistry,text="Hydrogen",width=20,command=lambda:fun("Hydrogen"))

b9.grid(column=2, row=2)

b10=Button(chemistry,text="S Block Elements",width=20,command=lambda : fun("S Block Elements"))

b10.grid(column=0, row=3)

b11=Button(chemistry,text="P Block Elements",width=20,command=lambda:fun("p block Elements"))

b11.grid(column=1, row=3)

b12=Button(chemistry,text="GOC",width=20,command=lambda:fun("GOC"))

b12.grid(column=2, row=3)

b13=Button(chemistry,text="Hydrocarbons",width=20,command=lambda:fun("Hydrocarbons"))

b13.grid(column=0, row=4)

b14=Button(chemistry,text="Environmental Chemistry",width=20,command=lambda:fun("Environmental Chemistry"))

b14.grid(column=1, row=4)

#Physics has same code structure as Chemistry

def Physics():

directory = os.getcwd() + "\\" + stu

physics = Tk()

physics.title("Physics")

def fun(a):

Sb=Tk()

Sb.title("Entry:")

directory = os.getcwd() + "\\" + uentry.get()

l1= Label(Sb, text="Theory")

l2= Label(Sb, text="Assignments")

l3= Label(Sb, text="Tests")

l4= Label(Sb, text="Notes")

l1.grid(column=0,row=0)

l2.grid(column=3,row=0)

l3.grid(column=6,row=0)

l4.grid(column=9,row=0)

def run1():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Theory\\File.pdf"

os.startfile(path)

file1=Button(Sb,text="Theory1",command=run1)

def run2():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Assignments\\File.pdf"

os.startfile(path)

file2 = Button(Sb,text="Assignment1",command=run2)

def run3():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Tests\\File.pdf"

os.startfile(path)

file3 = Button(Sb,text="Test1",command=run3)

file1.grid(column=0,row=2)

file2.grid(column=3,row=2)

file3.grid(column=6,row=2)

def notes():

name = "notes.txt"

path = uentry.get() + "\\Physics\\"+a+"\\"+name

f = open(path,'a')

f.close()

os.startfile(path)

file4=Button(Sb,text='notes',command=notes)

file4.grid(column=9,row=2)

b1=Button(physics,text="Units and Measurement",width=20,command=lambda:fun("Units and Measurement"))

b1.grid(column=0, row=0)

b2=Button(physics,text="Motion in Straight Line",width=20,command=lambda:fun("Motion in a Straight Line"))

b2.grid(column=1, row=0)

b3=Button(physics,text="Motion in a Plane",width=20,command=lambda:fun("Motion in a Plane"))

b3.grid(column=2, row=0)

b4=Button(physics,text="Laws of Motion",width=20,command=lambda:fun("Laws of Motion"))

b4.grid(column=0, row=1)

b5=Button(physics,text="WPE",width=20,command=lambda:fun("WPE"))

b5.grid(column=1, row=1)

b6=Button(physics,text="System of Particles",width=20,command=lambda:fun("System of Particles"))

b6.grid(column=2, row=1)

b7=Button(physics,text="Gravitation",width=20,command=lambda:fun("Gravitation"))

b7.grid(column=0, row=2)

b8=Button(physics,text="Solids",width=20,command=lambda:fun("Solids"))

b8.grid(column=1, row=2)

b9=Button(physics,text="Fluids",width=20,command=lambda:fun("Fluids"))

b9.grid(column=2, row=2)

b10=Button(physics,text="Thermal Properties of Matter",width=20,command=lambda:fun("Thermal Properties of Matter"))

b10.grid(column=0, row=3)

b11=Button(physics,text="Thermodynamics",width=20,command=lambda:fun("Thermodynamics"))

b11.grid(column=1, row=3)

b12=Button(physics,text="Kinetic Theory",width=20,command=lambda:fun("Kinetic Theory"))

b12.grid(column=2, row=3)

b13=Button(physics,text="Oscillations",width=20,command=lambda:fun("Oscillations"))

b13.grid(column=0, row=4)

b14=Button(physics,text="Waves",width=20,command=lambda:fun("Waves"))

b14.grid(column=1, row=4)

chem\_btn = Button(porc,text="Chemistry",width=10,command=Chemistry)

chem\_btn.grid(row=1,column=0)

phy\_btn = Button(porc,text="Physics",width=10,command=Physics)

phy\_btn.grid(row=1,column=1)

porc.mainloop()

#Incorrect Password Entered

else:

wpassword = Tk()

wpassword.geometry('250x50')

wpassword.title("Wrong Password")

l1= Label(wpassword, text="Incorrect password entered.")

l2 = Label(wpassword, text= "Please close this window and enter again.")

l1.grid(row=0,column=0)

l2.grid(row=1,column=0)

submit=Button(nowindow,text="Submit",width=10, command = norandom)

submit.grid(row=3,column=2)

nowindow.mainloop()

#If the user is a new user(Yes is pressed)

def if\_yes():

#Inputting username and password

yeswindow=Tk()

yeswindow.title("Entry:")

username=StringVar()

password=StringVar()

yeswindow.geometry('350x200')

l2=Label(yeswindow,text="Please enter your username")

uentry=Entry(yeswindow,textvariable=username)

uentry.grid(row=0,column=1)

l3=Label(yeswindow,text="Please enter your password")

pentry=Entry(yeswindow,textvariable=password,show="\*")

pentry.grid(row=1,column=1)

l2.grid(row=0,column=0)

l3.grid(row=1,column=0)

def yesrandom():

password = pentry.get()

stu=uentry.get()

mades = os.listdir(os.getcwd())

#Checking if folder already exists

if stu in mades:

window2 = Tk()

window2.geometry('250x50')

window2.title("Folder Exists")

l1= Label(window2, text="Folder already exists.")

l2 = Label(window2, text= "Please close this window and enter again.")

l1.grid(row=0,column=0)

l2.grid(row=1,column=0)

if stu not in mades:

print("Done")

path = os.getcwd() + "\\"+ stu

print(path)

os.mkdir(path)

chemistry = os.getcwd() + '\\'+ stu + '\\'+ 'Chemistry'

physics = os.getcwd() + '\\' + stu + '\\'+ 'Physics'

os.mkdir(chemistry)

os.mkdir(physics)

makechapters(path,"Chemistry")

makechapters(path,"Physics")

setpassword(password,path)

print("Folder made.")

porc = Tk()

porc.title("Phy or Chem:")

porc.geometry('400x100')

label = Label(porc,text="Physics or Chemistry?")

label.grid(row=0,column=0)

#Same code structure as before

def Chemistry():

chemistry = Tk()

chemistry.title("Chemistry")

def fun(a):

print(a)

Sb=Tk()

Sb.title("Entry:")

l1= Label(Sb, text="Theory")

l2= Label(Sb, text="Assignments")

l3= Label(Sb, text="Tests")

l4= Label(Sb, text="Notes")

l1.grid(column=0,row=0)

l2.grid(column=3,row=0)

l3.grid(column=6,row=0)

l4.grid(column=9,row=0)

def run1():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Theory\\File.pdf"

os.startfile(path)

file1=Button(Sb,text="Theory1",command=run1)

def run2():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Assignments\\File.pdf"

os.startfile(path)

file2 = Button(Sb,text="Assignment1",command=run2)

def run3():

path = os.getcwd()+"\\our\\Chemistry\\"+a+"\\Tests\\File.pdf"

os.startfile(path)

file3 = Button(Sb,text="Test1",command=run3)

file1.grid(column=0,row=2)

file2.grid(column=3,row=2)

file3.grid(column=6,row=2)

def notes():

name = "notes.txt"

path = uentry.get() + "\\Chemistry\\"+a+"\\"+name

f = open(path,'a')

f.close()

os.startfile(path)

file4=Button(Sb,text='notes',command=notes)

file4.grid(column=9,row=2)

directory = os.getcwd() + "\\" + stu

b1=Button(chemistry,text="SBCOC",width=20,command=lambda:fun("SBCOC"))

b1.grid(column=0, row=0)

b2=Button(chemistry,text="Structure of atom",width=20,command=lambda:fun("Structure of atom"))

b2.grid(column=1, row=0)

b3=Button(chemistry,text="Periodic Properties",width=20,command=lambda:fun("Periodic Properties"))

b3.grid(column=2, row=0)

b4=Button(chemistry,text="Chemical Bonding",width=20,command=lambda:fun("Chemical Bonding"))

b4.grid(column=0, row=1)

b5=Button(chemistry,text="States of Matter",width=20,command=lambda:fun("States of Matter"))

b5.grid(column=1, row=1)

b6=Button(chemistry,text="Thermodynamics",width=20,command=lambda : fun("Thermodynamics"))

b6.grid(column=2, row=1)

b7=Button(chemistry,text="Equilibrium",width=20,command=lambda:fun("Equilibrium"))

b7.grid(column=0, row=2)

b8=Button(chemistry,text="Redox Reactions",width=20,command=lambda:fun("Redox Reactions"))

b8.grid(column=1, row=2)

b9=Button(chemistry,text="Hydrogen",width=20,command=lambda:fun("Hydrogen"))

b9.grid(column=2, row=2)

b10=Button(chemistry,text="S Block Elements",width=20,command=lambda:fun("S Block Elements"))

b10.grid(column=0, row=3)

b11=Button(chemistry,text="P Block Elements",width=20,command=lambda:fun("p block Elements"))

b11.grid(column=1, row=3)

b12=Button(chemistry,text="GOC",width=20,command=lambda:fun("GOC"))

b12.grid(column=2, row=3)

b13=Button(chemistry,text="Hydrocarbons",width=20,command=lambda:fun("Hydrocarbons"))

b13.grid(column=0, row=4)

b14=Button(chemistry,text="Environmental Chemistry",width=20,command=lambda:fun("Environmental Chemistry"))

b14.grid(column=1, row=4)

def Physics():

directory = os.getcwd() + "\\" + stu

physics = Tk()

physics.title("Physics")

def fun(a):

Sb=Tk()

directory = os.getcwd() + "\\" + uentry.get()

l1= Label(Sb, text="Theory")

l2= Label(Sb, text="Assignments")

l3= Label(Sb, text="Tests")

l4= Label(Sb, text="Notes")

l1.grid(column=0,row=0)

l2.grid(column=3,row=0)

l3.grid(column=6,row=0)

l4.grid(column=9,row=0)

def run1():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Theory\\File.pdf"

os.startfile(path)

file1=Button(Sb,text="Theory1",command=run1)

def run2():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Assignments\\File.pdf"

os.startfile(path)

file2 = Button(Sb,text="Assignment1",command=run2)

def run3():

path = os.getcwd()+"\\our\\Physics\\"+a+"\\Tests\\File.pdf"

os.startfile(path)

file3 = Button(Sb,text="Test1",command=run3)

file1.grid(column=0,row=2)

file2.grid(column=3,row=2)

file3.grid(column=6,row=2)

def notes():

name = "notes.txt"

path = uentry.get() + "\\Physics\\"+a+"\\"+name

f = open(path,'a')

f.close()

os.startfile(path)

file4=Button(Sb,text='notes',command=notes)

file4.grid(column=9,row=2)

b1=Button(physics,text="Units and Measurement",width=20,command=lambda:fun("Units and Measurement"))

b1.grid(column=0, row=0)

b2=Button(physics,text="Motion in Straight Line",width=20,command=lambda:fun("Motion in a Straight Line"))

b2.grid(column=1, row=0)

b3=Button(physics,text="Motion in a Plane",width=20,command=lambda:fun("Motion in a Plane"))

b3.grid(column=2, row=0)

b4=Button(physics,text="Laws of Motion",width=20,command=lambda:fun("Laws of Motion"))

b4.grid(column=0, row=1)

b5=Button(physics,text="WPE",width=20,command=lambda:fun("WPE"))

b5.grid(column=1, row=1)

b6=Button(physics,text="System of Particles",width=20,command=lambda:fun("System of Particles"))

b6.grid(column=2, row=1)

b7=Button(physics,text="Gravitation",width=20,command=lambda:fun("Gravitation"))

b7.grid(column=0, row=2)

b8=Button(physics,text="Solids",width=20,command=lambda:fun("Solids"))

b8.grid(column=1, row=2)

b9=Button(physics,text="Fluids",width=20,command=lambda:fun("Fluids"))

b9.grid(column=2, row=2)

b10=Button(physics,text="Thermal Properties of Matter",width=20,command=lambda:fun("Thermal Properties of Matter"))

b10.grid(column=0, row=3)

b11=Button(physics,text="Thermodynamics",width=20,command=lambda:fun("Thermodynamics"))

b11.grid(column=1, row=3)

b12=Button(physics,text="Kinetic Theory",width=20,command=lambda:fun("Kinetic Theory"))

b12.grid(column=2, row=3)

b13=Button(physics,text="Oscillations",width=20,command=lambda:fun("Oscillations"))

b13.grid(column=0, row=4)

b14=Button(physics,text="Waves",width=20,command=lambda:fun("Waves"))

b14.grid(column=1, row=4)

chem\_btn = Button(porc,text="Chemistry",width=10,command=Chemistry)

chem\_btn.grid(row=1,column=0)

phy\_btn = Button(porc,text="Physics",width=10,command=Physics)

phy\_btn.grid(row=1,column=1)

porc.mainloop()

submit=Button(yeswindow,text="Submit",width=10, command = yesrandom)

submit.grid(row=3,column=2)

yeswindow.mainloop()

#Opening window

window = Tk()

window.geometry('550x350')

window.title("Students' Academic Guide")

l1= Label(window, text="Are You a New Student?")

l1.pack()

#Checking for returning or new user

yes\_btn = Button(window,text="Yes",width=10,command=if\_yes)

yes\_btn.pack()

no\_btn = Button(window,text="No",width=10,command=if\_no)

no\_btn.pack()

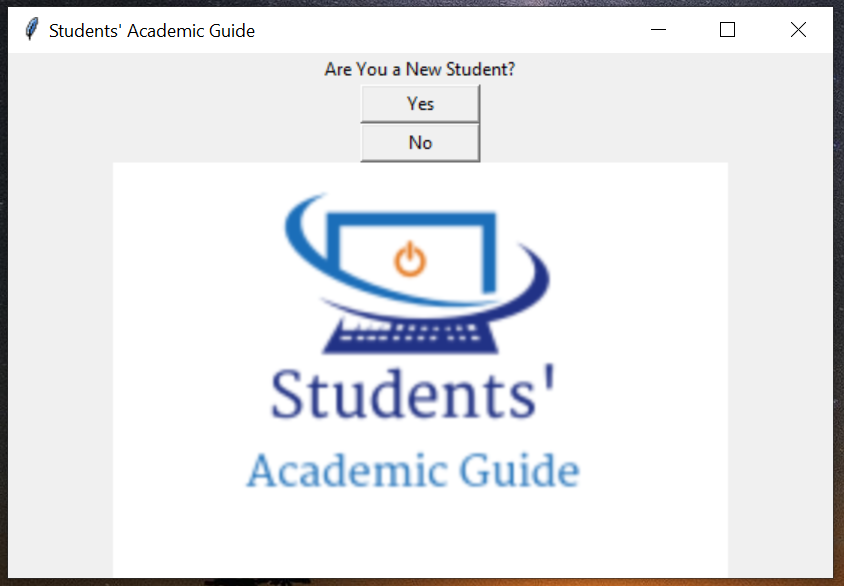
x= PhotoImage(file='logo.png')

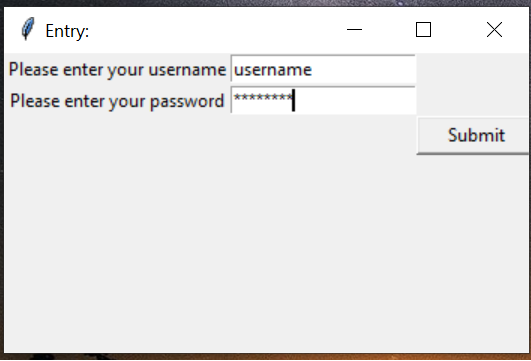
logo=Label(window,image=x)

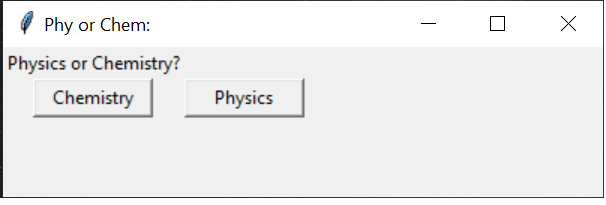
logo.pack()

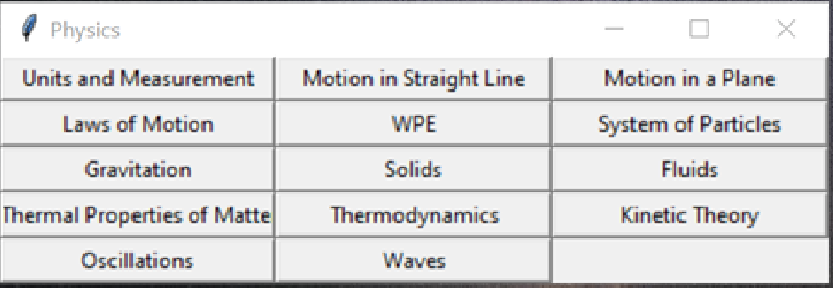
window.mainloop()

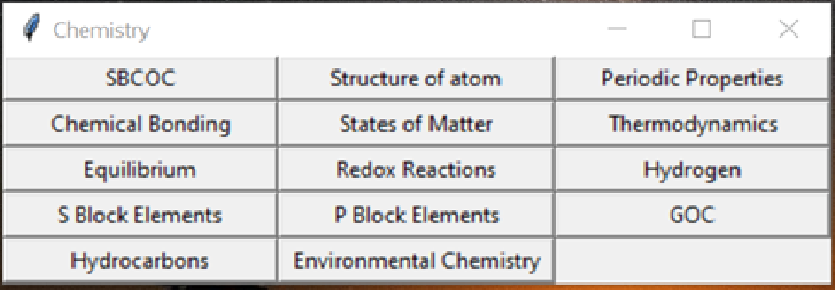
Screenshots

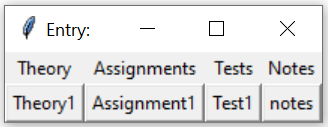


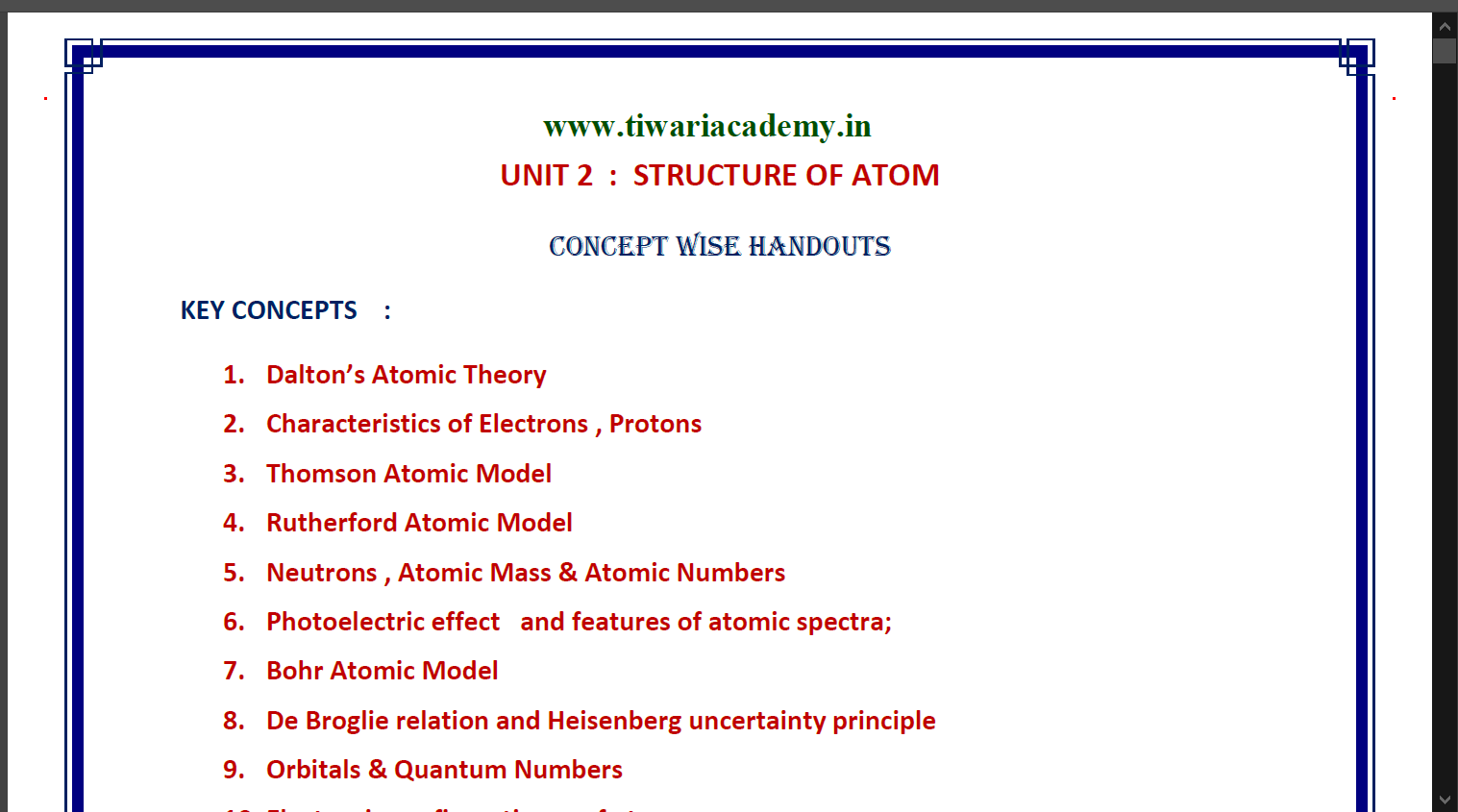












Limitations

This program is far from perfect, and there are certain limitations we would like to acknowledge. They are listed below:

* The program could have been more robust, given time.
* The ‘ Our’ folder with all information containing theory, test and assignments which

Can be deleted by as it can be accessed by any user.

Improvements

Some improvements that can be made to the project are:

* Database could have been added
* A user authentication program could have been made to strengthen the validation.
* Online correction of test and assignments could have been added.

Bibliography

The following resources were immensely useful for the completion for this program :

* <http://www.stackoverflow.com>
* https://www.geeksforgeeks.org
* Computer Science with Python by Sumita Arora