**Ashadeep Sr. Sec. School, Pawti Road (Samalkha)**

****

**Computer Science (083)**

Academic Year : 2020 – 2021

**Investigatory Project Report on School Management System**

**Submitted by :** **Submitted to :**

Vishal Tyagi Mr. Sachin

Roll No. – 20 PGT Computer

Certificate

This is to certify that **Vishal Tyagi** of class **XII** Roll no. **20** has completed his project work in the subject of **Computer Science (083)** entitled as **School Management System** as required according to the syllabus prescribed by the Central Board of Secondary Education for the academic session **2020 – 2021**.

**Examiner :**

Name : (Mr. Scahin)

Signature PGT Computer

Acknowledgement

I would like to express my special thanks of gratitude to my computer teacher **Mr. Sachin** and IT head **Mr. Dinesh** as well as our principal **Mr. Rajesh Saini** who gave me the golden opportunity to do this wonderful project on the topic **School Management System**, which also helped me in doing a lot of Research and i came to know about so many things. I have no more valuable words to express my thanks, but my heart is still full of favour received from every person.

Index

|  |  |  |
| --- | --- | --- |
| S.No. | Topic | Sign |
| 1 | Write a program to calculate sum of the digits of a number using functions. |  |
| 2 | Write a function that takes a list of numbers as argument and return the sum of all numbers which are divisible by 5. |  |
| 3 | Write a function that takes a string as an argument and check if string is Palindrome |  |
| 4 | Write a Python function that takes a list of strings as an argument and displays the strings which starts with “S” or “s”. |  |
| 5 | Write Python script to create a dictionary with players name and their score. Write a function that accepts this and displays the name of the player with highest score. |  |
| 6 | Read a text file line by line and display each word separated by #. |  |
| 7 | Read a text file and display the number of vowels/ consonants/uppercase/ lowercase characters in the file. |  |
| 8 | Write a program in python to read a text file and remove all the lines that contain the character `a’ in a file and write it to another file. |  |
| 9 | Write a Python program to Recursively find the factorial of a natural number |  |
| 10 | Write a Python program to implement a stack using a list data-structure. |  |
| 11 | Write a python Program to implement Queue using a list data structure |  |
| 12 | Write a function in Python to check the primarily of an integer number. |  |
| 13 | Write a function in Python to swap() first half of the elements with the second half of the elements. |  |
| 14 | Write a recursive function in Python to find the sum of all elements of a list. |  |
| 15 | Write a recursive Python program to test if a string is a palindrome or not. |  |
| 16 | Write a Python Program to read a text file. Find out the total number of words available in this file. |  |
| 17 | Write a random number generator that generates random numbers between 1 and 6 (simulates a dice). |  |
| 18 | Write a python program to generate a simple bar graph using Pyplot. |  |
| 19 | Write a python program to generate Pie-chart using Pyplot. |  |
| 20 | Write a Python program to plot the function y = x2 using the pyplot or matplotlib libraries. |  |
| 21 | Write a function to insert a record in table using python and MySQL interface. |  |
| 22 | Based on Sql Query, answer the given questions for tables CLIENT and PRODUCT |  |
| 23 | Based on Sql Query, answer the given questions for tables CONSIGNOR and  CONSIGNEE |  |
| 23 | **School Management System designed in Tkinter with database Connectivity.** |  |

**#1 Write a program to calculate sum of the digits of a number using functions.**

def sumOfDigits(num):

sum = 0

for ch in num:

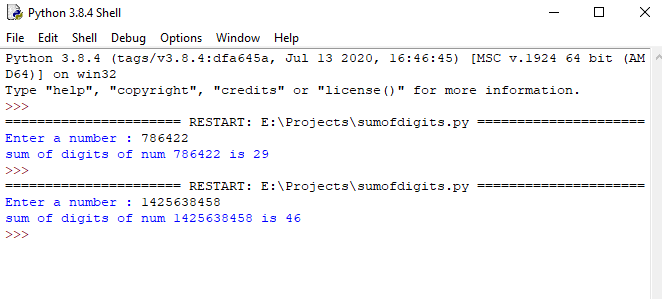
sum += int(ch)

else:

return sum

inum = input("Enter a number : ")

print("sum of digits of num",inum,"is",sumOfDigits(inum))



**#2 Write a function that takes a list of numbers as argument and return the sum of all numbers which are divisible by 5.**

def sumOfList(lst):

sum = 0

for i in lst:

if i%5 == 0:

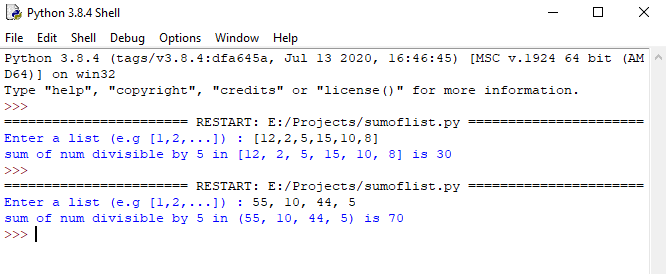
sum += i

else:

return sum

lst\_num = eval(input("Enter a list (e.g [1,2,...]) : "))

print("sum of num divisible by 5 in",lst\_num,"is",sumOfList(lst\_num))



**#3 Write a function that takes a string as an argument and return Palindrome if the string is a palindrome otherwise return Not A Palindrome.**

def checkPalindrome(string):

if string == string[::-1]:

result = string + " is palindrome"

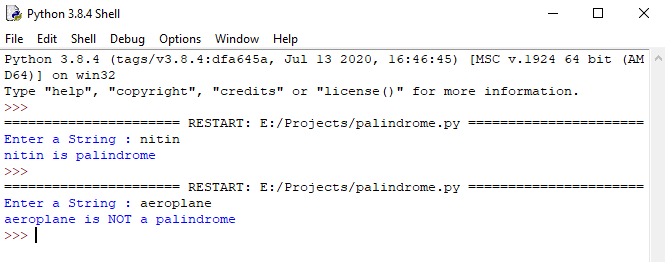
else:

result = string + " is NOT a palindrome"

return result

iStr = input("Enter a String : ")

print(checkPalindrome(iStr))



**#4 Write a Python function that takes a list of strings as an argument and displays the strings which starts with “S” or “s”. Also write a program to invoke this function.**

def strStartFromS(lst):

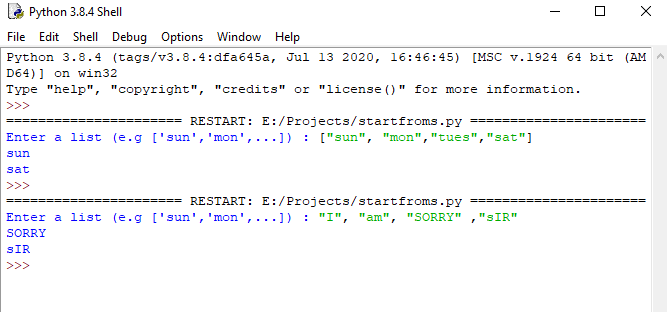
for i in lst:

if i[0].lower() == "s":

print(i)

lst\_str = eval(input("Enter a list (e.g ['sun','mon',...]) : "))

strStartFromS(lst\_str)



**#5 Write Python script to create a dictionary with players name and their score. Write a function that accepts this dictionary as an argument and displays the name of the player with highest score.**

def maxScoreFromDict(data\_dict):

lst\_name = list(data\_dict.keys())

lst\_score = list(data\_dict.values())

max\_score = max(lst\_score)

p\_name = lst\_name[lst\_score.index(max\_score)]

print(p\_name, " has the higest score",max\_score)

tplayer = int(input("Enter total no of players : "))

my\_dict = {}

for i in range(tplayer):

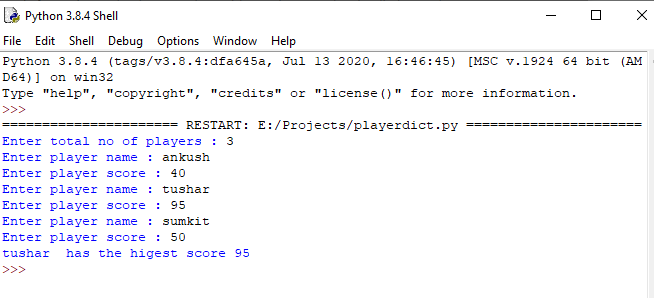
pname = input("Enter player name : ")

pscore = int(input("Enter player score : "))

my\_dict[pname] = pscore

else:

maxScoreFromDict(my\_dict)



**#6 Read a text file line by line and display each word separated by #.**

def sepFile(filename):

with open(filename,'r') as file:

for line in file:

words = line.split()

print("#".join(words))

file.close()

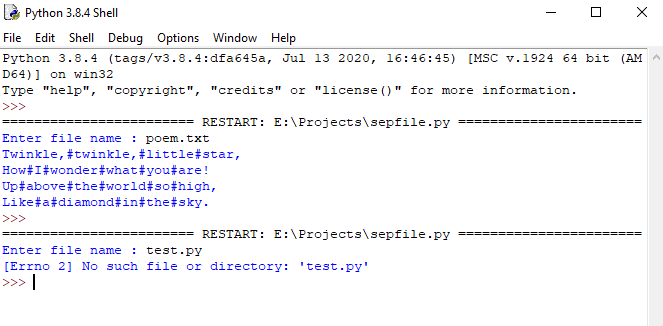
try:

ifile = input("Enter file name : ")

sepFile(ifile)

except Exception as e:

print(e)



**#7 Read a text file and display the number of vowels/ consonants/uppercase/lowercase characters in the file.**

def cntFile(filename):

cVowel = 0

cConsonant = 0

cLower = 0

cUpper = 0

vowels = ['a','e','i','o','u']

consonant = ['b','c','d','f','g','h','j','k','l','m','n','p','q','r','s','t','v','w','x','y','z']

with open(filename,'r') as file:

text = file.read()

for ch in text:

if ch in vowels:

cVowel += 1

if ch in consonant:

cConsonant += 1

if ch.islower():

cLower += 1

if ch.isupper():

cUpper += 1

file.close()

print("Total Vowels are",cVowel)

print("Total Consonants are",cConsonant)

print("Total Lowercase letters are",cLower)

print("Total Uppercase letters are",cUpper)

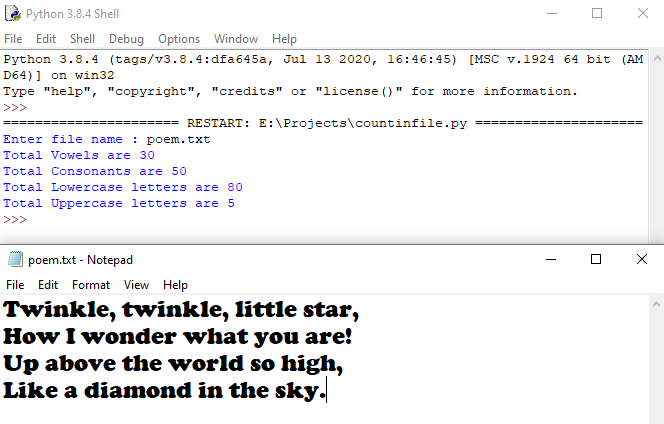
try:

ifile = input("Enter file name : ")

cntFile(ifile)

except Exception as e:

print(e)



**#8 Write a program in python to read a text file and remove all the lines that contain the character `a’ in a file and write it to another file.**

def repFile(ofilename, nfilename):

ofile = open(ofilename,'r')

nfile = open(nfilename, 'w')

text = ofile.readlines()

for line in text:

if 'a' in line:

nfile.write(line)

ofile.close()

nfile.close()

print("done!")

try:

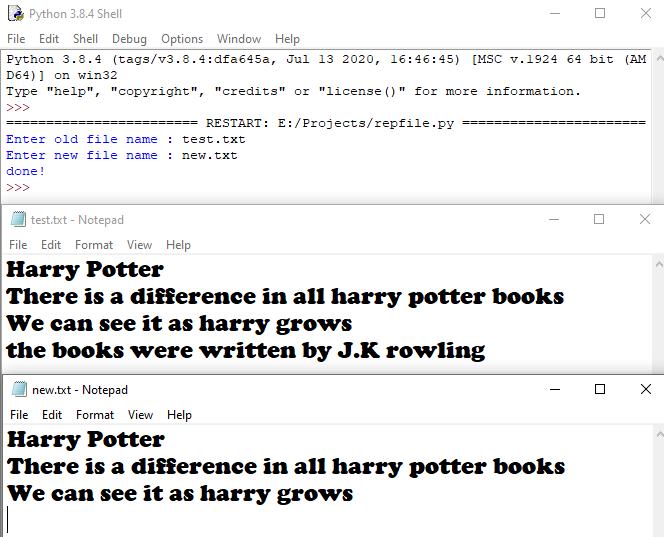
ioldfile = input("Enter old file name : ")

inewfile = input("Enter new file name : ")

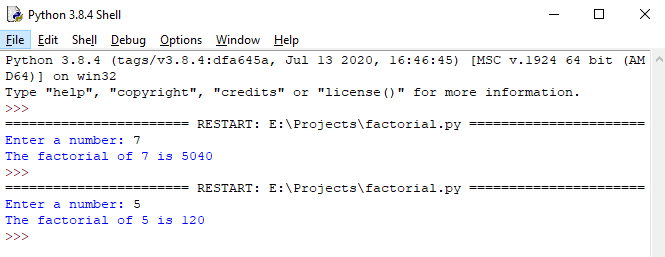
repFile(ioldfile, inewfile)

except Exception as e:

print(e)



**#9 Write a Python program to Recursively find the factorial of a natural number.**

**#10 Write a Python program to implement a stack using a list data-structure.**

def recur\_factorial(n):

if n == 1:

return n

else:

return n\*recur\_factorial(n-1)

num = int(input("Enter a number: "))

if num < 0:

print("factorial doesn’t exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of",num,"is",recur\_factorial(num))

print("The factorial of",num,"is",recur\_factorial(num))

def push(lst):

val=eval(input("Enter number to push : "))

lst.append(val)

print(val, "is pushed\n")

def pop(lst):

if lst==[]:

print("UnderFlow\n")

else:

val=lst.pop()) # for Stack

print(val, "is popped\n")

def display(lst):

if lst==[]:

print("Stack Empty\n")

else:

for i in range(len(lst)-1,-1,-1):

print(lst[i])

print("\n")

lst=[]

while True:

print("1. push\n2. pop\n3. display\n4. exit")

ch=int(input("Enter your choice : "))

if ch==1:

push(lst)

elif ch==2:

pop(lst)

elif ch==3:

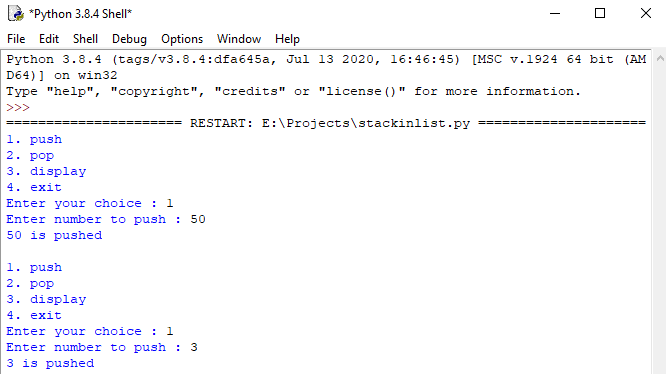
display(lst)

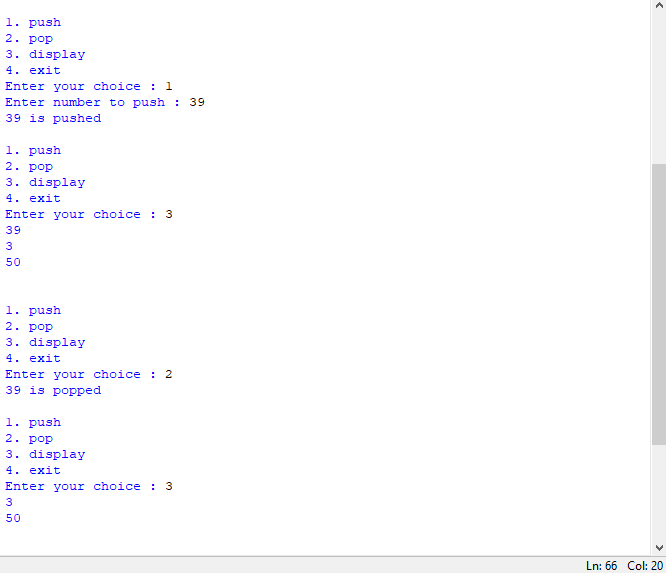
elif ch==4:

break

else:

print("invalid choice!")





**#11 Write a python Program to implement Queue using a list data structure.**

def push(lst):

val=eval(input("Enter number to push : "))

lst.append(val)

print(val, "is pushed\n")

def pop(lst):

if lst==[]:

print("UnderFlow\n")

else:

val=lst.pop(0) # for Queue

print(val, "is popped\n")

def display(lst):

if lst==[]:

print("Queue Empty\n")

else:

for i in range(len(lst)-1,-1,-1):

print(lst[i])

print("\n")

lst=[]

while True:

print("1. push\n2. pop\n3. display\n4. exit")

ch=int(input("Enter your choice : "))

if ch==1:

push(lst)

elif ch==2:

pop(lst)

elif ch==3:

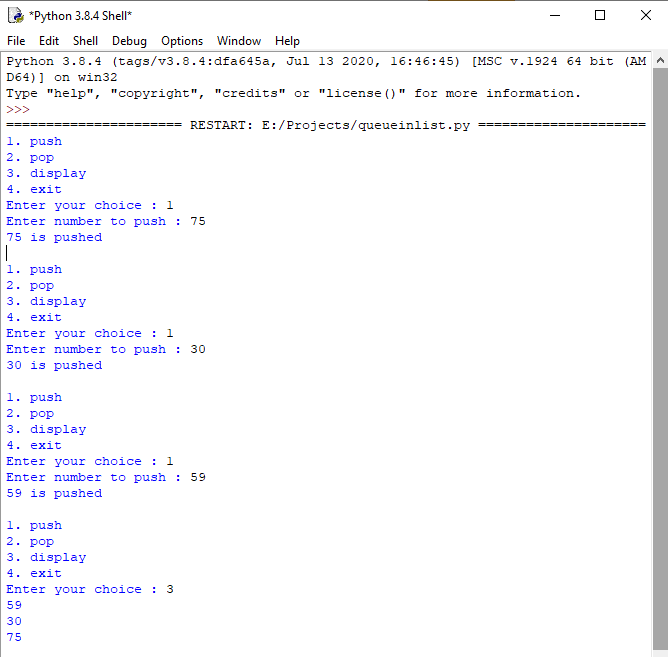
display(lst)

elif ch==4:

break

else:

print("invalid choice!")





**#12 Write a function in Python to check the primarily of an integer number.**

def check\_prime(num):

if num > 1:

for i in range(2,num):

if (num % i) == 0:

print(num,"is not a prime number")

break

else:

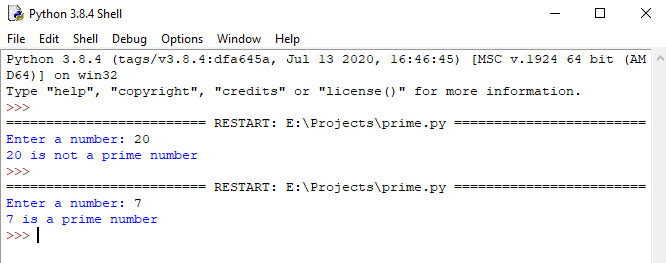
print(num,"is a prime number")

else:

print("Invalid Input!")

num = int(input("Enter a number: "))

check\_prime(num)



**#13 Write a function in Python to swap() first half of the elements with the second half of the elements.**

def swap\_list(lst):

n = len(lst)

mid = int(n/2)

result = []

if n%2 != 0:

mid\_lst = lst[mid]

fh\_lst = lst[0:mid]

lh\_lst = lst[mid+1:n]

result.extend(lh\_lst)

result.append(mid\_lst)

result.extend(fh\_lst)

else:

fh\_lst = lst[0:mid]

lh\_lst = lst[mid:n]

result.extend(lh\_lst)

result.extend(fh\_lst)

return result

smpList = [12, 35, 9, 56, 24, 38, 50, 8]

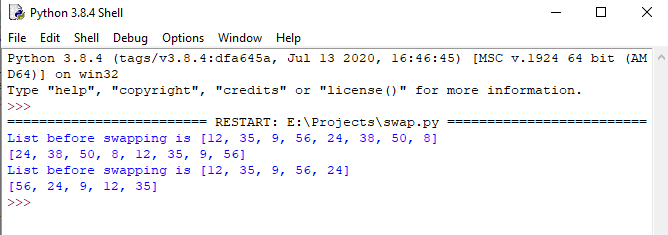
print("List before swapping is", smpList)

print(swap\_list(smpList))

smpList2 = [12, 35, 9, 56, 24]

print("List before swapping is", smpList2)

print(swap\_list(smpList2))

****

**#14 Write a recursive function in Python to find the sum of all elements of a list.**

def sum\_arr(arr,size):

if (size == 0):

return 0

else:

return arr[size-1] + sum\_arr(arr,size-1)

n=int(input("Enter the number of elements for list : "))

a=[]

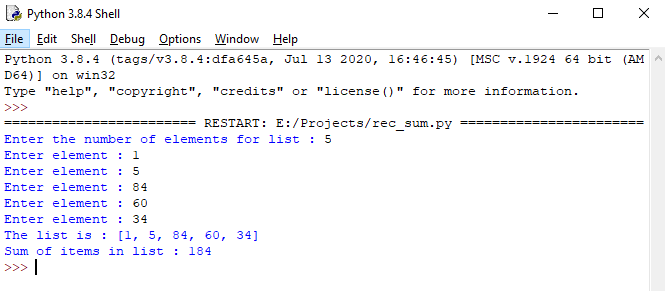
for i in range(0,n):

element=int(input("Enter element : "))

a.append(element)

print("The list is :", a)

print("Sum of items in list :",sum\_arr(a,n))

**#15 Write a recursive Python program to test if a string is a palindrome or not.**

def isPalindrome(s):

s = s.lower()

n = len(s)

if n < 2:

return True

elif s[0] == s[n - 1]:

return isPalindrome(s[1: n - 1])

else:

return False

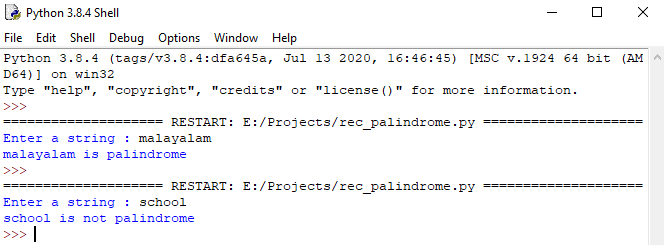
st = input("Enter a string : ")

if (isPalindrome(st)) :

print(st, "is palindrome")

else :

print(st, "is not palindrome")

****

**#16 Write a Python Program to read a text file. Find out the total number of words available in this file.**

def cntFile(filename):

file = open(filename,'r')

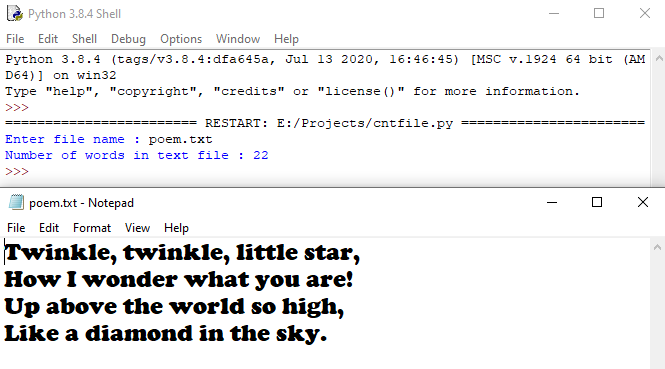
text = file.read()

words = text.split()

return len(words)

ifile = input("Enter file name : ")

print('Number of words in text file :', cntFile(ifile))

****

**#17 Write a random number generator that generates random numbers between 1 and 6 (simulates a dice).**

import random

def rollDice():

no = random.randint(1,6)

if no == 1:

print("[-----]")

print("[ ]")

print("[ 0 ]")

print("[ ]")

print("[-----]")

if no == 2:

print("[-----]")

print("[ 0 ]")

print("[ ]")

print("[ 0 ]")

print("[-----]")

if no == 3:

print("[-----]")

print("[ ]")

print("[0 0 0]")

print("[ ]")

print("[-----]")

if no == 4:

print("[-----]")

print("[0 0]")

print("[ ]")

print("[0 0]")

print("[-----]")

if no == 5:

print("[-----]")

print("[0 0]")

print("[ 0 ]")

print("[0 0]")

print("[-----]")

if no == 6:

print("[-----]")

print("[0 0 0]")

print("[ ]")

print("[0 0 0]")

print("[-----]")

while True:

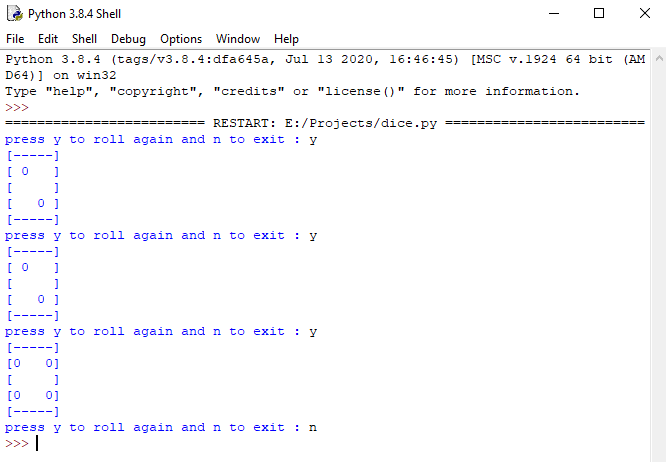
x=input("press y to roll again and n to exit : ")

if x=='y' or x=='Y':

rollDice()

else:

break

****

**#18 Write a python program to generate a simple bar graph using Pyplot.**

import numpy as np

import matplotlib.pyplot as plt

data = {'C':20, 'C++':15, 'Java':30,

'Python':35}

courses = list(data.keys())

values = list(data.values())

fig = plt.figure(figsize = (10, 5))

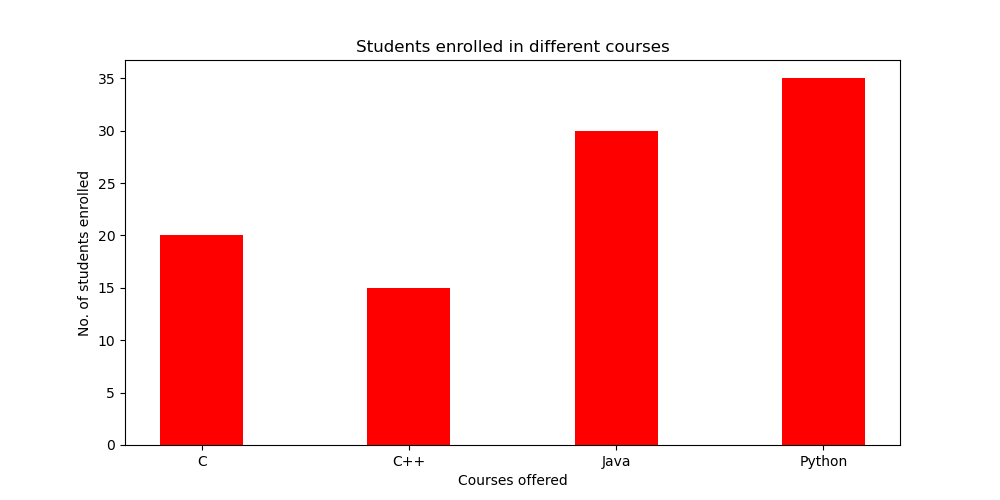
plt.bar(courses, values, color ='red', width = 0.4)

plt.xlabel("Courses offered")

plt.ylabel("No. of students enrolled")

plt.title("Students enrolled in different courses")

plt.show()

****

**#19 Write a python program to generate Pie-chart using Pyplot.**

from matplotlib import pyplot as plt

import numpy as np

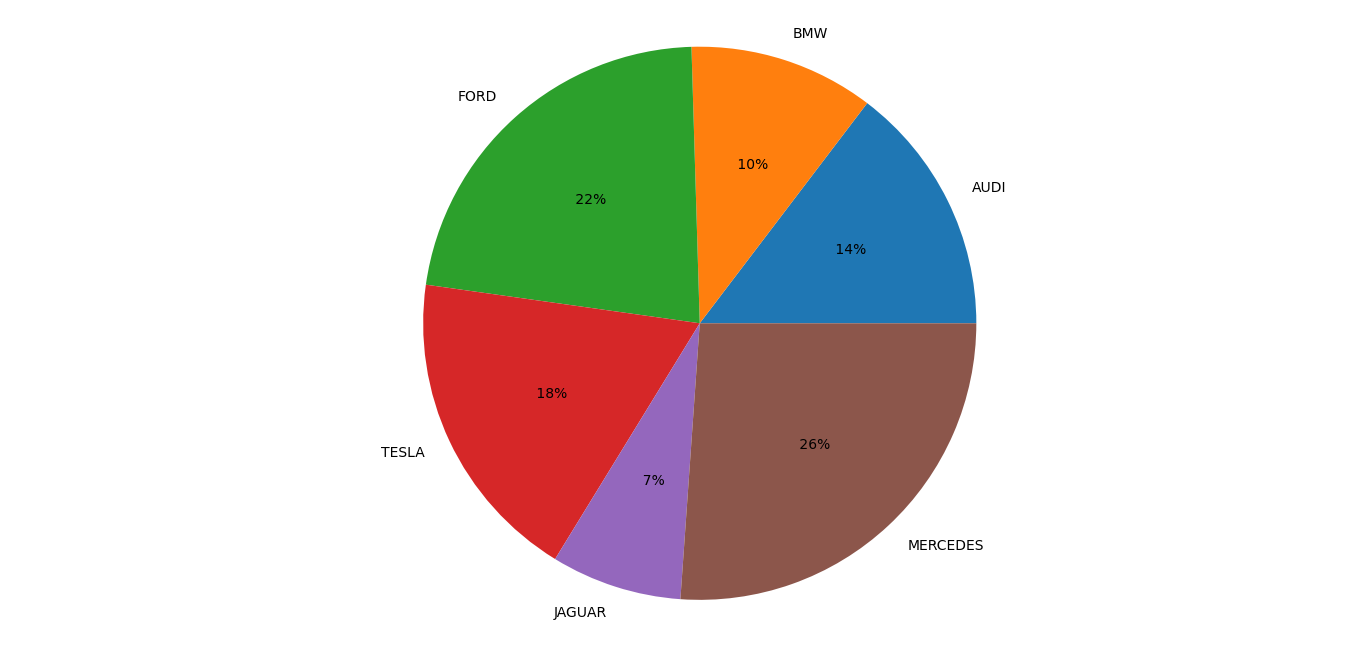
cars = ['AUDI', 'BMW', 'FORD', 'TESLA', 'JAGUAR', 'MERCEDES']

data = [23, 17, 35, 29, 12, 41]

fig = plt.figure(figsize =(10, 7))

plt.pie(data, labels = cars , autopct="%3d%%")

plt.show()

****

**#20 Write a Python program to plot the function y = x2 using the pyplot or matplotlib libraries.**

import matplotlib.pyplot as plt

x = [1,2,3,4,5]

y = [i\*\*2 for i in x]

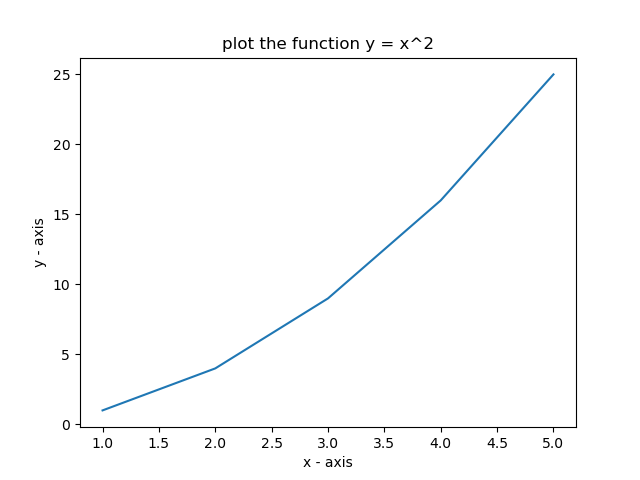
plt.plot(x, y)

plt.xlabel('x - axis')

plt.ylabel('y - axis')

plt.title('plot the function y = x^2')

plt.show()



**Based on SQL Query**

**#21 Answer the following questions based upon the table given below:-**

**TABLE:PRODUCT**

|  |  |  |  |
| --- | --- | --- | --- |
| **P\_ID** | **PRODUCTNAME** | **MANUFACTURER** | **PRICE** |
| TP01 | TALCOM POWDER | LAK | 40 |
| FW05 | FACE WASH | ABC | 45 |
| BS01 | BATH SOAP | ABC | 55 |
| SH06 | SHAMPOO | XYZ | 120 |
| FW12 | FACE WASH | XYZ | 95 |

**TABLE:CLIENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **C\_ID** | **CLIENTNAME** | **City** | **P\_ID** |
| 01 | COSMETIC SHOP | Delhi | FW05 |
| 06 | TOTAL HEALTH | Mumbai | BS01 |
| 12 | LIVE LIFE | Delhi | SH06 |
| 15 | PRETTY WOMAN | Delhi | FW12 |
| 16 | DREAMS | Banglore | TP01 |

Q.1 To display the details of those clients whose city is “delhi”?

Ans: Select \* from client where city=’Delhi’;

Q.2 To display the details of products whose price is in the range of 50 to 100 ?  
Ans: Select \* from product where price between 50 and 100; or  
select \* from product where price>=50 and price<=100;

Q.3 TO DISPLAY THE client name ,city from table client and product name and price from the table product with their corresponding matching p\_id ?  
Ans: Select clientname, city, productname, price from client, product where client.p\_id=product.p\_id;

Q.4 To increase the price of the product by 10  
Ans: Update product set price=price+10;

Q.5 Select distinct city from client;  
Ans: Delhi  
Mumbai  
Banglore

**#22 Consider the following tables CONSIGNOR and CONSIGNEE. Further answer the questions given below.**

**TABLE:CONSIGNOR**

|  |  |  |  |
| --- | --- | --- | --- |
| **CNORID** | **CNORNAME** | **CNORADDRESS** | **CITY** |
| ND01 | R SINGHAL | 24; ABC ENCLAVE | NEW DELHI |
| ND02 | AMIT KUMAR | 123; PALM AVENUE | NEW DELHI |
| MU15 | R KOHLI | 5/A; SOUTH STREET | MUMBAI |
| MU50 | S KAUR | 27-K; WESTEND | MUMBAI |

**TABLE:CONSIGNEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CNEEID** | **CNORID** | **CNEENAME** | **CNEEADDRESS** | **CNEECITY** |
| MU05 | ND01 | RAHUL KISHORE | 5; PARK AVENUE | MUMBAI |
| ND08 | ND02 | P DHINGRA | 16/J; MOORE ENCLAVE | NEW DELHI |
| KO19 | MU15 | A P ROY | 2A; CENTRAL AVENUE | KOLKATA |
| MU32 | ND02 | S MITTAL | P 245; AB COLONY | MUMBAI |
| ND48 | MU50 | B P JAIN | 13; BLOCK D; A VIHAR | NEW DELHI |

Q1. To display the names of all the consignors from Mumbai  
Ans Select cnorname From consignor Where city=’mumbai’;

Q2. To display the cneeid, cnorname, cnoradress, cneenmae, cneeaddress for every consignee  
Ans Select cneeid,cnoraddress,cneename,cneeaddess From consignor, consignee Where consignor.cnorid=consignee.cnorid;

Q3. To display consignee details in ascending order of cneename.  
Ans Select \* From consignee Order by cneename;

Q4. To display number of consignee from each city.  
Ans Select cneecity,count(cneecity) From consignee Group by cneecity;

Q5. Select distinct cneecity from consignee;  
Ans Mumbai**Project on School management System**

Introduction

This project is all about to handle whole the activities of the school. SMS has most of the facilities that a modern school requires to computerize its day-to-day jobs. It provides facilities to keep the records of student, fees, teaching and non-teaching staff with all their required details along with all required transaction handling. It has facilities to generate various types of reports, which are required by the management during normal business operations to operate the business effectively.

Objective

The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and exposed the students how programming skills helps in developing a good software.

In other words we can say that our project has the following objectives:-

* Make the system computerize
* Reduce time consumption
* Reduce error scope
* All system management are automated
* Centralized database management
* Easy Operations for operator of the system
* No paper work requirement

Package/Module used in the project

* Tkinter
* SQLite3
* Bcrypt
* Pillow
* Ttkthemes
* python-dotenv
* ttkwidgets

**Project Directory Structure**

**Coding**

**School Management System**

**main.py**

import tkinter as ttk

from ttkthemes import ThemedTk

import functions as fn

from login import Login

from panel import Panel

from dotenv import load\_dotenv

import os

class MainApp(ThemedTk):

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

ThemedTk.\_\_init\_\_(self, background=True, theme="breeze")

load\_dotenv()

self.title(os.getenv("APP\_TITLE"))

self.iconbitmap('images/icon.ico')

# self.state('zoomed')

self.minsize(960, 550)

self.geometry("1024x550")

container = ttk.Frame(self)

container.pack(side = "top", fill = "both", expand = True)

container.grid\_rowconfigure(0, weight = 1)

container.grid\_columnconfigure(0, weight = 1)

self.frames = {}

lst\_table = fn.getMenuItems()

lst\_table.append('empty')

for tbl in lst\_table:

frame = Panel(container, self, tbl)

self.frames[tbl] = frame

frame.grid(row = 0, column = 0, sticky ="nsew")

self.show\_frame("dashboard")

def show\_frame(self, cont):

frame = self.frames[cont]

self.title(fn.winTitle(cont.title()))

frame.tkraise()

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

if fn.checkAuth():

fn.startMain()

else:

fn.start(Login)

import os

from tkinter import \*

from tkinter import ttk

from PIL import ImageTk

from PIL import Image as pilImage

from dotenv import load\_dotenv

import functions as fn

class Login():

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

self.master = master

load\_dotenv()

self.master.title(fn.winTitle('Login'))

self.master.iconbitmap('images/icon.ico')

self.master.geometry('450x650')

self.master.resizable(False, False)

self.username = StringVar()

self.password = StringVar()

self.icon = ImageTk.PhotoImage(pilImage.open('images/icon.ico'))

lbl\_logo = ttk.Label(self.master, image=self.icon)

content = ttk.Frame(self.master)

**login.py**

**panel.py**

import bcrypt, os

from functools import partial

from tkinter import \*

from tkinter import ttk

from ttkwidgets.frames import ScrolledFrame

from PIL import ImageTk

from PIL import Image as pilImage

import functions as fn

from dotenv import load\_dotenv

if user and pswd:

if fn.isUserValidate(user, pswd):

fn.createAuth(user)

self.master.destroy()

fn.startMain()

else:

fn.error("Please make sure that the details are correct!")

else:

fn.error("Please enter your login details!")

if \_\_name\_\_ == "\_\_main\_\_":

fn.startMain()

class Panel(Frame):

def \_\_init\_\_(self, parent, controller, page\_name):

Frame.\_\_init\_\_(self, parent)

load\_dotenv()

self.page\_name = page\_name

lbl\_title = Label(self,height=2, text=os.getenv("APP\_TITLE"), font=("Segoe UI", 25, "bold"), bg="#1b2838",fg="white")

menu\_frame = ScrolledFrame(self)

lst\_menu = fn.fn.getMenuItems()

lst\_menu.append('empty')

for data in lst\_menu:

menuimg = ImageTk.PhotoImage(pilImage.open("images/menu/"+ data +".png").resize((50, 50), pilImage.ANTIALIAS))

menubtn = ttk.Button(menu\_frame.interior, text=data, image=menuimg, compound=LEFT, command=partial(controller.show\_frame,data))

menubtn.image = menuimg

menubtn.pack()

lbl\_title.pack(fill=X)

menu\_frame.pack(side=LEFT, fill=BOTH)

if page\_name not in fn.getTablesFromDB():

self.customState()

else:

self.initApp()

def customState(self):

content = ttk.Frame(self)

frm\_state = ttk.Frame(content)

if self.page\_name == 'dashboard':

image = ImageTk.PhotoImage(pilImage.open('images/overview.png').resize((150, 150), pilImage.ANTIALIAS))

title = "Welcome Back, {}".format(fn.currentUser())

subtitle = "<<< Browse the menu for more options."

else:

image = ImageTk.PhotoImage(pilImage.open('images/error.png').resize((150, 150), pilImage.ANTIALIAS))

title = "Sorry! This service is currently unavailable"

subtitle = "Please try again later or ask the developer for it."

lbl\_img = ttk.Label(frm\_state, image=image)

lbl\_img.image = image

lbl\_title = ttk.Label(frm\_state, text=title, font=("Segoe UI", 18, "bold"))

lbl\_subtitle = ttk.Label(frm\_state, text=subtitle, font=("Segoe UI", 14))

lbl\_img.pack()

lbl\_title.pack()

lbl\_subtitle.pack(pady=(0,20))

frm\_state.place(relx=0.5, rely=0.5, anchor=CENTER)

content.pack(side=LEFT, fill=BOTH, expand=True)

def initApp(self):

ctrl\_frame = ScrolledFrame(self)

tbl\_frame = ttk.Frame(self)

mngr\_frame = ttk.Frame(ctrl\_frame.interior)

lbl\_ctrls = ttk.Label(mngr\_frame, text='Manage Student', font=("Segoe UI", 14, "bold")).pack(pady=20)

self.lst\_entry = fn.getColumnsFromTable(self.page\_name)

self.lst\_variables = []

for column in self.lst\_entry:

entry\_var = StringVar()

mng\_frame = ttk.Frame(mngr\_frame)

mng\_lbl = ttk.Label(mng\_frame, text=column.replace('\_', ' ').title(), font=("Segoe UI", 14)).pack(side=TOP, anchor=NW)

if column == 'password':

mng\_entry = ttk.Entry(mng\_frame, textvariable=entry\_var, show="\*").pack(side=BOTTOM, fill=X, expand=True)

else:

mng\_entry = ttk.Entry(mng\_frame, textvariable=entry\_var).pack(side=BOTTOM, fill=X, expand=True)

mng\_frame.pack(fill=X, pady=5)

self.lst\_variables.append(entry\_var)

btn\_add = ttk.Button(mngr\_frame, text="add", command=self.add\_entry).pack(side=LEFT,fill=X, expand=True, padx=5, pady=30)

btn\_delete = ttk.Button(mngr\_frame, text="delete", command=self.delete\_entry).pack(side=LEFT,fill=X, expand=True, padx=5, pady=30)

btn\_update = ttk.Button(mngr\_frame, text="update", command=self.update\_entry).pack(side=LEFT,fill=X, expand=True, padx=5, pady=30)

btn\_clear = ttk.Button(mngr\_frame, text="clear", command=self.clear\_entry).pack(side=LEFT,fill=X, expand=True, padx=5, pady=30)

self.tbl\_cols = fn.getColumnsFromTable(self.page\_name)

if 'password' in self.tbl\_cols:

self.tbl\_cols.remove('password')

self.tbl\_data = ttk.Treeview(tbl\_frame, selectmode ='browse', columns=self.tbl\_cols, show='headings')

for col in self.tbl\_cols:

self.tbl\_data.heading(col, text=col.replace('\_', ' ').title())

self.tbl\_data.bind("<<TreeviewSelect>>", self.get\_entry)

self.show\_data()

verscrlbar = ttk.Scrollbar(tbl\_frame, orient='vertical', command=self.tbl\_data.yview)

horscrlbar = ttk.Scrollbar(tbl\_frame, orient='horizontal', command=self.tbl\_data.xview)

verscrlbar.pack(side=RIGHT, fill=Y)

horscrlbar.pack(side=BOTTOM, fill=X)

self.tbl\_data.configure(xscrollcommand = horscrlbar.set)

self.tbl\_data.configure(yscrollcommand = verscrlbar.set)

ctrl\_frame.pack(fill=BOTH, expand=True,side=LEFT)

mngr\_frame.pack(fill=BOTH, expand=True, padx=20)

tbl\_frame.pack(fill=BOTH, expand=True,side=LEFT)

self.tbl\_data.pack(fill=BOTH, expand=True)

def show\_data(self):

self.tbl\_data.delete(\*self.tbl\_data.get\_children())

for row in fn.fetchDataFromTable(",".join(self.tbl\_cols), self.page\_name):

self.tbl\_data.insert("", END, values=row)

def get\_entry(self, event):

item = self.tbl\_data.focus()

item\_id = self.tbl\_data.item(item)["values"][0]

for i in range(len(self.lst\_entry)):

self.lst\_variables[i].set("")

if self.lst\_entry[i] == 'password':

pass

else:

data = fn.fetchRecordFromTable(item\_id, self.page\_name, self.lst\_entry[i])

self.lst\_variables[i].set(data[0])

def clear\_entry(self):

for i in range(len(self.lst\_entry)):

self.lst\_variables[i].set("")

for item in self.tbl\_data.selection():

self.tbl\_data.selection\_remove(item)

def delete\_entry(self):

if len(self.lst\_variables[0].get()) > 0:

if fn.fetchRecordFromTable(item\_id, self.page\_name) is not None:

fn.deleteDataFromTable(item\_id, self.page\_name)

self.show\_data()

self.clear\_entry()

else:

fn.error("Record isn't exist with Id - {}".format(item\_id))

else:

fn.error("Id field must be required!")

item\_id = self.lst\_variables[0].get()

if fn.fetchRecordFromTable(item\_id, self.page\_name) is not None:

fn.deleteDataFromTable(item\_id, self.page\_name)

self.show\_data()

self.clear\_entry()

else:

fn.error("Record isn't exist with Id - {}".format(item\_id))

else:

fn.error("Id field must be required!")

def add\_entry(self):

dict\_items = {}

run\_error = False

for i in range(len(self.lst\_entry)):

if len(self.lst\_variables[i].get()) > 0:

if self.lst\_entry[i] == 'password':

self.lst\_variables[i].set(bcrypt.hashpw(self.lst\_variables[i].get().encode(), bcrypt.gensalt()))

dict\_items[self.lst\_entry[i]] = self.lst\_variables[i].get()

else:

run\_error = True

fn.error("All field are required!")

break

if not run\_error:

fn.updateDataToTable(dict\_items, self.lst\_variables[0].get(), self.page\_name)

self.show\_data()

self.clear\_entry()

if \_\_name\_\_ == "\_\_main\_\_":

fn.startMain()

**functions.py**

import sqlite3, bcrypt, os, pickle

from tkinter import \*

from tkinter import ttk, messagebox

from ttkthemes import ThemedTk

from dotenv import load\_dotenv

from main import \*

load\_dotenv()

def hashpw(pswd):

result = bcrypt.hashpw(pswd.encode(), bcrypt.gensalt())

return result.decode()

def initDb(user='admin', pswd='admin@123'):

try:

if os.path.isfile(os.getenv("DB")) and os.path.getsize(os.getenv("DB")) > 100:

pass

else:

conn = sqlite3.connect(os.getenv("DB"))

cur = conn.cursor()

dump\_file = open("database/dump.sqlite", "r")

cur.executescript(dump\_file.read())

dump\_file.close()

cur.execute('INSERT INTO users(username, password) VALUES("{}","{}")'.format(user, hashpw(pswd)))

conn.commit()

conn.close()

except Exception as e:

if conn:

conn.close()

print(e)

os.remove(os.getenv("DB"))

def isUserValidate(user, pswd):

initDb()

isValidate = False

try:

conn = sqlite3.connect(os.getenv("DB"))

cur = conn.cursor()

cur.execute("SELECT password FROM users WHERE username = '{}'".format(user, hashpw(pswd)))

result = cur.fetchall()

conn.close()

except Exception as e:

print(e)

result = []

pswd = pswd.encode()

for data in result:

if bcrypt.checkpw(pswd, data[0].encode()):

isValidate = True

break

return isValidate

def createAuth(user):

auth\_file = open("database/auth", "wb")

pickle.dump(user,auth\_file)

auth\_file.close()

def currentUser():

try:

authFile = open("database/auth", "rb")

user = pickle.load(authFile)

authFile.close()

return user

except:

return 'guest'

def checkAuth():

initDb()

isLogged = False

try:

conn = sqlite3.connect(os.getenv("DB"))

cur = conn.cursor()

cur.execute("SELECT \* FROM users WHERE username = '{}'".format(currentUser()))

result = cur.fetchone()

conn.close()

if result is None:

os.remove("database/auth")

else:

isLogged = True

except Exception as e:

if conn:

conn.close()

print(e)

return isLogged

def winTitle(title):

return '{} | {}'.format(title, os.getenv("APP\_TITLE"))

def getTablesFromDB(db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

cur.execute("SELECT name FROM sqlite\_master WHERE type='table' ORDER BY name")

result = [table[0] for table in cur.fetchall()]

conn.close()

return result

except Exception as e:

if conn:

conn.close()

print(e)

return []

def getMenuItems(db=os.getenv("DB")):

tbl\_list = getTablesFromDB()

tbl\_list.remove('sqlite\_sequence')

tbl\_list.insert(0, 'dashboard')

return tbl\_list

def getColumnsFromTable(table, db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

cur.execute("SELECT \* FROM {}".format(table))

result = [description[0] for description in cur.description]

conn.close()

return result

except Exception as e:

if conn:

conn.close()

print(e)

return []

def fetchDataFromTable(column, table, db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

sql = "SELECT {} FROM {}".format(column, table)

cur.execute(sql)

result = cur.fetchall()

conn.close()

return result

except Exception as e:

if conn:

conn.close()

error(e)

return []

def addDataToTable(dict\_items, table, db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

sql = 'INSERT INTO {}({}) VALUES({})'.format(table, ', '.join(i for i in dict\_items), ', '.join('"{}"'.format(dict\_items[i]) for i in dict\_items))

cur.execute(sql)

conn.commit()

conn.close()

success("Record added successfully!")

except Exception as e:

if conn:

conn.close()

error(e)

def updateDataToTable(dict\_items, item\_id, table, db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

sql = 'UPDATE {} SET {} WHERE id={}'.format(table, ', '.join('{}="{}"'.format(i,j) for i,j in dict\_items.items()), item\_id)

cur.execute(sql)

conn.commit()

conn.close()

success("Record updated successfully!")

except Exception as e:

if conn:

conn.close()

error(e)

conn.close()

error(e)

def deleteDataFromTable(item\_id, table, db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

sql = "DELETE FROM {} WHERE id = {}".format(table, item\_id)

cur.execute(sql)

conn.commit()

conn.close()

success("Record deleted successfully!")

except Exception as e:

if conn:

conn.close()

error(e)

def fetchRecordFromTable(item\_id, table, column="\*", db=os.getenv("DB")):

initDb()

try:

conn = sqlite3.connect(db)

cur = conn.cursor()

sql = "SELECT {} FROM {} WHERE id = {}".format(column, table, item\_id)

cur.execute(sql)

result = cur.fetchone()

conn.close()

return result

except Exception as e:

if conn:

conn.close()

print(e)

return ()

def error(e):

messagebox.showerror("Something went wrong!", e)

def success(e):

messagebox.showinfo("Success!", e)

def start(window):

try:

root = ThemedTk(background=True, theme="breeze")

app = window(root)

root.mainloop()

except Exception as e:

print(e)

def startMain():

app = MainApp()

app.mainloop()

**dump.sqlite**

CREATE TABLE IF NOT EXISTS students (

id INTEGER PRIMARY KEY AUTOINCREMENT,

roll\_no INTEGER NOT NULL,

name TEXT NOT NULL,

father\_name TEXT NOT NULL,

mother\_name TEXT NOT NULL,

email TEXT DEFAULT NULL,

gender TEXT NOT NULL,

phone TEXT DEFAULT NULL,

date\_of\_birth TEXT,

address TEXT,

class\_id INTEGER NOT NULL

);

CREATE TABLE IF NOT EXISTS teachers (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

father\_name TEXT NOT NULL,

mother\_name TEXT NOT NULL,

email TEXT DEFAULT NULL,

gender TEXT NOT NULL,

phone TEXT DEFAULT NULL,

date\_of\_birth TEXT,

address TEXT

);

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

teacher\_id INTEGER,

username TEXT NOT NULL,

password TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS class (

id INTEGER PRIMARY KEY AUTOINCREMENT,

teacher\_id INTEGER,

name TEXT NOT NULL

);

bcrypt

ttkthemes

python-dotenv

Pillow

ttkwidgets

**requirements.txt**

APP\_TITLE = "School Management System"APP\_TITLE = "School Management System"

APP\_SUBTITLE = "Authenticate yourself to continue..."

DB = "database/school.db"

APP\_SUBTITLE = "Authenticate yourself to continue..."

DB = "database/school.db"

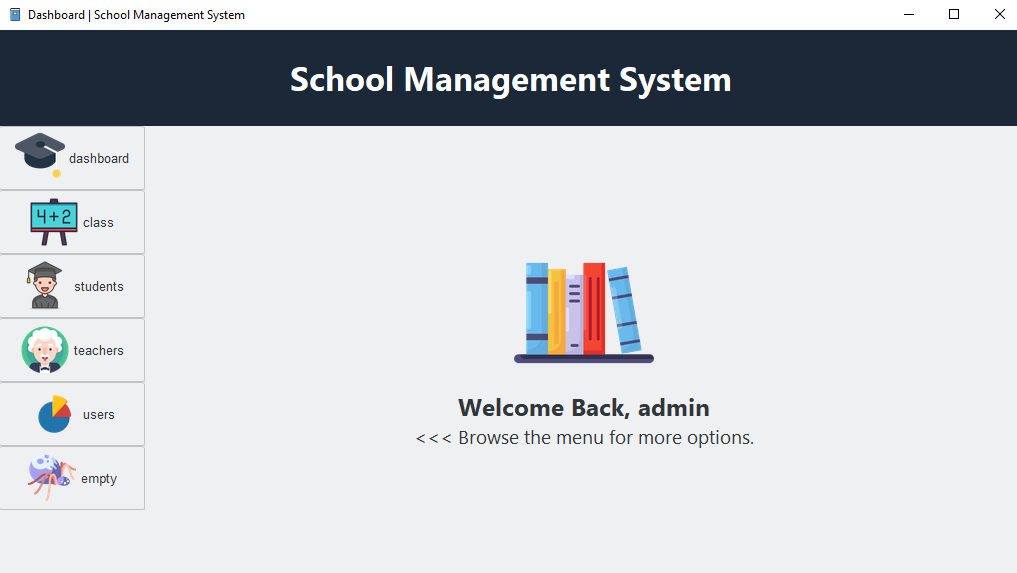
**.env**

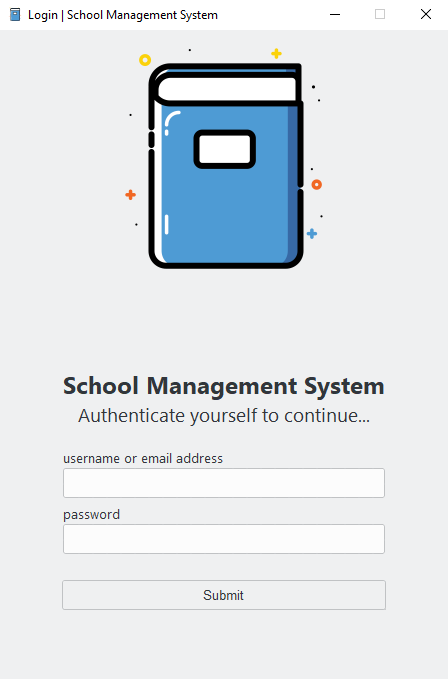
Commands to execute program

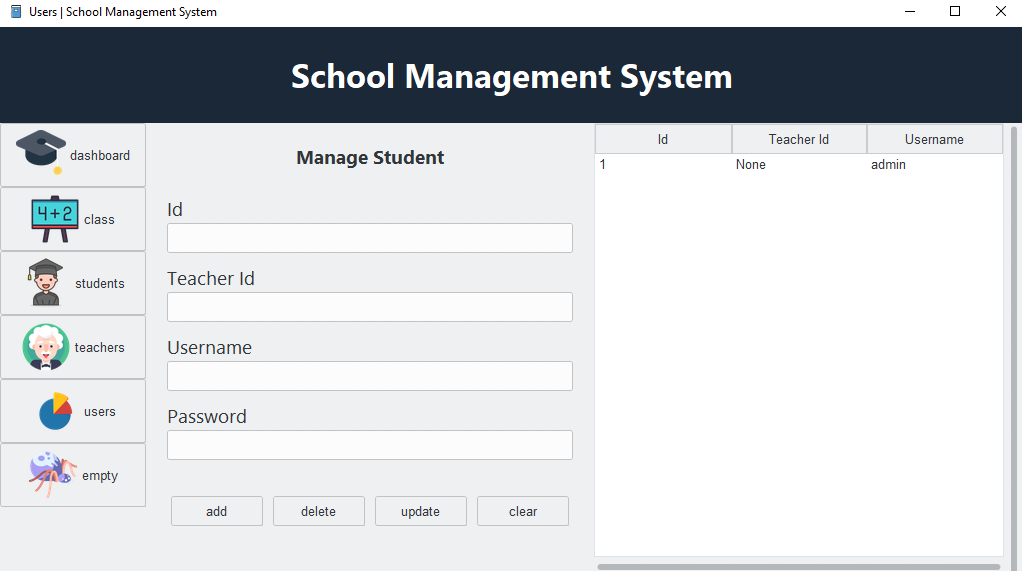
**python main.py**

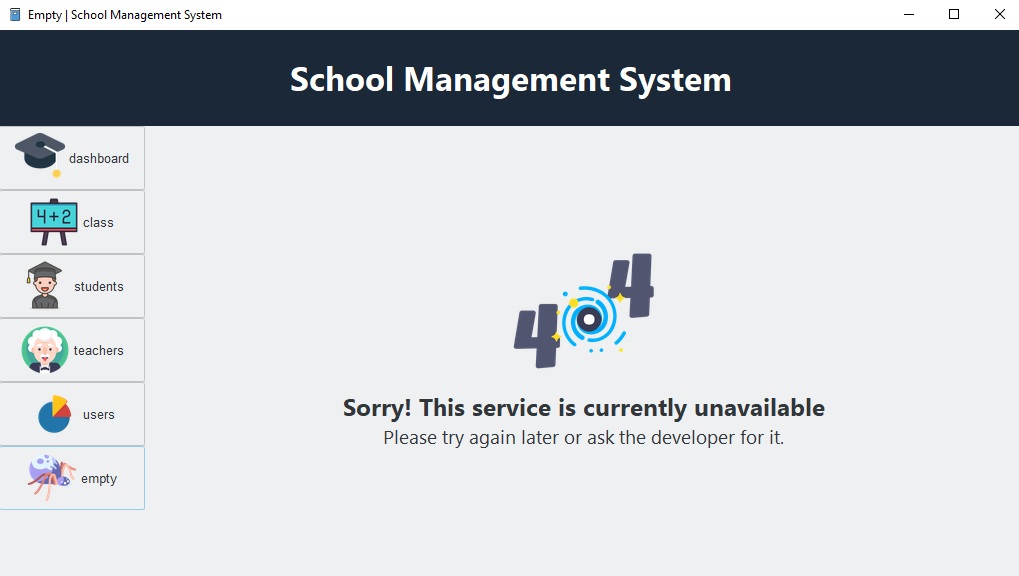
**pip install -r requirements.txt**

Screenshots









Limitations

The drawbacks in School Management System software can be counted on fingers; with mostly only benefits, these systems have a few countable downsides. Often, applications face minor technical glitches and these systems are no exception but, ratification is immediate. Only, people who are accustomed to regular use of smartphones or computers can operate this software. Extensive modules and features make it difficult for a user to utilise the application. With huge flow in traffic the application is prone to performance issues. The risk of data mishandling might be bothersome; but all these drawbacks can be evaded by choosing proper, cost-efficient and best software that best benefits an organization.

Requirements

* **Hardware**
* Modern Operating System
* x86 64-bit CPU (Intel / AMD architecture)
* 4 GB RAM
* 5 GB free disk space
* **Software**
* Python 3.9.2
* MySQL 8.0 or SQLite

Bibliography

Special thanks to those listed sites which help me a lot to learn new coding stuffs and help me to solve difficult problems that occurs during building project.

Also I hereby give credit to site iconfinder.com for all the graphics and images used in the program and assure that those images can be used even commercially.