New Horizon Public School, Airoli

Grade: XII (2021-2022)

Computer Science Project

Group members : Aaditya Ghadshi XII-A

Yash Varma XII-B

Malcom Raj XII-A

New Horizon Public School, Airoli

CERTIFICATE

This is to certify that this PROJECT _____

_____*is the work of*

_____*XII A/B Board Roll No.* _____

_____*XII A/B Board Roll No.* _____

_____*XII A/B Board Roll No.* _____

These students have satisfactorily completed the PROJECT work for the year 20 to 20 as per the guidelines laid down by CBSE.

Internal Examiner

External Examiner
Date:

Principal,
New Horizon Public School
Airoli, Navi Mumbai.

Acknowledgement

Presentation, inspiration and motivation have always played a key role in the success of any venture.

We wish to express deep profound sense of gratitude to our teacher, Ms. Merlin Chezhian for her expert help and valuable guidance, comments and suggestion.

We are glad for our team that laid our project to this extent. All thanks to our team.

Last but not the least our team would like to thank each and everyone who directly or indirectly lent their hands in the success of the project.

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 whereas the other languages use punctuations. It has fewer syntactical constructions than other languages.

Objective of the project

The main objective of our program is to get used in our day to day work, like Maintaining the contact records, Creating notes, Calculating the figures using calculator, also a calendar to keep a track of dates simultaneously, etc. This program can be used in thebackends of GUI applications which provide this features if required, editing, adding and updating of records is being improved which results to be helpful, proper for management and can reduce human efforts.

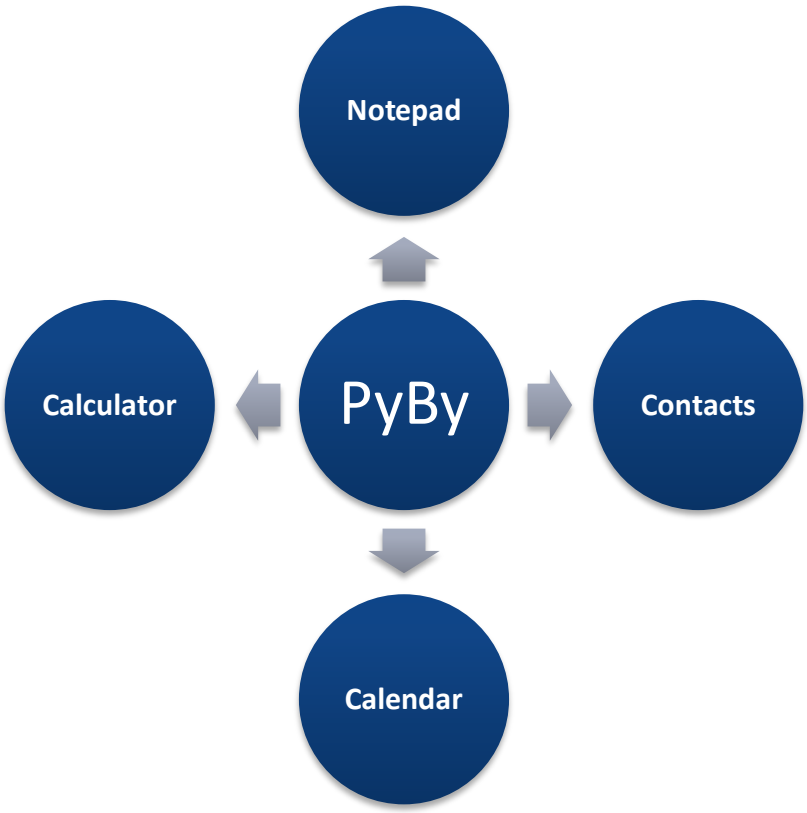
Synopsis

In our project we are trying to make such a program that it will be useful in our daily work. Program contains the working of various application like Notepad, Contacts, Calendar, Calculator, etc. which can also be used to program a application which contains all of this functioning features.

Our team has tried to make the program most flexible, to make it function in the proper manner.

Though the program cannot be made that perfect but our team has tried its best for the user to be convenient.

Basic Flow Chart of Program



Minimum System Requirement

Operating System: Windows 8

CPU: Intel Core™i3

RAM: 4GB

System Defined Functions used in program:

1) print() Function:

The print() function prints the specified message to the screen, or other standard output device.

Syntax:

>>>print(object,sep=separator,end=end)

2) input() Function:

The input() function allows user's input.

Syntax:

>>>input(prompt)

Parameters:

Prompt indicates the message before input from the user.

3) open() Function:

The open() function opens a file, and returns it as a file object.

Syntax:

>>>open(file, mode)

Parameters:

i. File: Path and name of the file to be opened.

ii. Mode: Mode of the file to be opened, i.e. Read, Write and Append modes.

4) range() Function:

The range() function returns a sequence of numbers, starting from 0 by default, and increments 1 (by default), and stops before a specified number.

Syntax:

>>>range(start, stop, step)

Parameters:

i. Start: Optional argument. An integer number specifying at which position to start. Default is 0.

ii. Stop: Required argument. An integer number specifying the incrementation. Default is 1.

iii. Step: Optional argument. An integer number specifying the incrementation. Default is 1.

5) len() Function:

The len() function returns the number of items in an object. When the object is a string, the len() function returns the number of characters in the string.

Syntax:

>>>len(object)

6) int() Function:

The int() function converts the specified values into an integer number.

Syntax:

>>>int(value, base)

Parameter:

i. Value: A number or a string that can be converted into

an integer number.

ii. Base: A number representing the number format. Default value: 10.

7) float() Function:

The float() function converts the specified values into an floating point number.

Syntax:

>>>float(value)

8) dict() Function:

The dict() function creates a dictionary. A dictionary is a collection which is unordered, changeable and indexed.

Syntax:

>>>dict(keyword arguments)

Parameter:

Required arguments. As many keyword arguments you like, separated by comma: key=value.

9) .remove() Function:

Python List remove() is an inbuilt function in the Python programming language that removes a given object from the List.

Syntax:

>>>List.remove(item)

10) .index() Function:

index() is an inbuilt function in Python, which searches for a given element from the start of the list and returns the lowest index where the element appears.

Syntax:

>>> List.index(item)

Inbuilt Modules imported in the program:

- **calendar Module:** Python defines an inbuilt module calendar that handles operations related to the calendar. The calendar module allows output calendars like the program and provides additional useful functions related to the calendar
  calendar.calendar(Year): This function prints the year calendar of the specified year.
  calendar.month(Year,Month): This function prints the month calendar of specified year and month.

- **os Module:** The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The 'os'

modules include many functions to interact with the file system.
os.remove(File): This function removes the specified file from the system.

User Defined Functions used in the modules of the program:

1. **Notepad(Module):**

i. Main() Function: It allows the user to select the task perform.
ii. NamesFile() Function: It creates the file to save the names of the notes if it does not exists.
iii. CreateNew() Function: It creates new note. There are sub-functions used in this function to perform the task:
   a. SavedNotes(): It Checks whether is thereany saved notes binary file in the system.
   b. Naming(): It helps in the naming conventions of the notes.
iv. Open() Function: It opens the notepad file in the read-mode only.
v. EOpen() Function: It opens the notepad file in edit mode.
vi. Delete() Function: It deletes the notepad file which is saved with specified name.

2. **Contacts (Module):**

i. Main () Function: It allows user to select the taskto be performed.

ii. Display() Function: It displays all the contacts saved.

iii. Search() Function: It allows the user to search the saved contact.

iv. Add() Function: It allows the user to save contact details into the saved contacts.

v. Edit() Function: It allows user to edit the contact details.

vi. Delete() Function: It allows the user to delete the saved contacts.

3. **Calendar(Module):**

Main() Function: It allows the user to select the type of calendar to be displayed using calendar module.

4. **Calculator(Module):**

This module comprises of functions of all the mathematical operations.

Source Code:

Main.py:

```
print('--------------------Program PyBy------------
--------')
import Notepad
import Contacts
import Calendar
#import Calculator
#import UnitConverter
while True:
    print('''Menu:
    1.Notepad
    2.Contacts
    3.Calendar
    4.Calculator
    5.Exit''')
    ch=int(input('Enter your choice: '))
    if ch==1:
        Notepad.Main()
    elif ch==2:
        Contacts.Main()
    elif ch==3:
        Calendar.Main()
    elif ch==4:
        Calculator.Main()
    elif ch==5:
        break
    else:
        print('Entered wrong choice!')
```

1)Notepad Module:

```
import pickle
import os
def Main():

print('**********Notepad**********
***')
    while True:
        print('''Notepad:
    1.Create New Note.
    2.Open Note(Read Only).
    3.Edit Note.
    4.Delete Note.
    5.Exit.
    ''')
        ch=int(input("Enter your choice: "))
        if ch==1:
            CreateNew()
            print()
        elif ch==2:
            Open()
            print()
        elif ch==3:
            EOpen()
            print()
        elif ch==4:
            Delete()
            print()
        elif ch==5:
            break
        else:
            print("Entered wrong choice!")
```

```
def NamesFile(): #Function to create file of names
of notes if does not exists
    try:
        Snotes=open("Notes","rb")
    except IOError:
        Snotes=open("Notes","wb+")
        L=[]
        pickle.dump(L,Snotes)
        Snotes.close()
def Naming():    #Function to do the naming
conventions.
    Snotes=open('Notes',"rb") #File to take the
names of the Notes.
    N_name=input("Enter the Name of the Note.: ")
    if N_name==str():
        N_name='NewNote'  #Default Name for New
Note.
    try:
        names=pickle.load(Snotes)
    except Exception:
        print('')
    #Also numbering the Note with same name.
    k=1
    while True:
        if N_name in names:
            if N_name[-1].isdigit():
                N_name=N_name[:-1]+str(k)
                k+=1
            else:
                N_name+=str(k)
                k+=1
        elif N_name not in names:
            break
    names.append(N_name)
    print(names)
    return names
```

```
def CreateNew():
    NamesFile()  #Function to create file of names
of notes if does not exists.
    Names=Naming() #Function to do the naming
conventions.
    #Opening NoteFile.
    print()
    print()
    l=len(Names)-1
    Note=''
    print('''Press: 1.Save.
Press: 2.Don't Save.
''')
    Flag=0
    print("\t"*5+Names[l]+"\n==>>")
    #Taking Multiline Input.
    while True:
        TEXT=input()
        if TEXT=='1':
            Flag=1
            break
        elif TEXT=='2':
            break
        Note+=TEXT+'\n'
    if Flag==1: #Writing and saving the note in the
file.
        Snotes=open("Notes","wb+")
        pickle.dump(Names,Snotes)
        Snotes.close()
        f=open(Names[l],'w')
        f.write(Note)
        f.close()
        print("Note Saved!")
def Open():#READ ONLY
    #Text file with saved notes.
    NamesFile()
```

```
    Snotes=open('Notes',"rb")
    try:
        Notes=pickle.load(Snotes)
    except Exception:
        print()
    while True:
        if Notes==[]:
            print("No Notes saved!")
            break
        else:
            print("Saved Notes: ",Notes)
        N_name=input("Enter the name of the note to
be opened: ")
        if N_name in Notes:
            fobj=open(N_name,"r+")
            note=fobj.read()
            print("\t"*5+N_name+"\n==>>")
            print(note)
            fobj.close()
        elif N_name not in Notes:
            print("Note not found!")
        break
    print()
def EOpen():#Editing Mode
    NamesFile()
    Snotes=open('Notes',"rb")
    try:
        Notes=pickle.load(Snotes)
    except Exception:
        print()
    while True:
        if Notes==[]:
            print("No Notes saved!")
            break
        else:
            print("Saved Notes: ",Notes)
```

```
        N_name=input("Enter the name of the note to
be opened: ")
        while True:
            if N_name in Notes:
                print()
            elif N_name not in Notes:
                print("Note not found!")
                break
            print('''Modes: 1.Write Mode.
       2.Edit Line Mode.
       3.Clear All
       4.Exit.''')
            ch=int(input("Enter the mode: "))
            if ch==1:
                Write(N_name)
            elif ch==2:
                EditLine(N_name)
            elif ch==3:
                fobj=open(N_name,'w+')
                fobj.close()
                print('Cleared all successfully')
            elif ch==4:
                break
            else:
                print("Entered wrong choice!")
        break
def Write(N_name):
    print('''Press: 1.Save Changes.
Press: 2.Don't Save.
''')
    fobj=open(N_name,"r+")
    print("\t"*5+N_name+"\n==>>")
    print(fobj.read())
    print("==>>")
    Note=''
    #Taking Multiline Input.
```

```
    Flag=0
    while True:
        TEXT=input()
        if TEXT=='1':
            Flag=1
            break
        elif TEXT=='2':
            break
        Note+=TEXT+'\n'
    if Flag==1: #Writing and saving the note in the
file.
        fobj.write(Note)
        fobj.close()
        print("Changes Saved!")
def EditLine(N_name):
    f=open(N_name,"r+")
    print("\t"*5+N_name+"\n==>>")
    lines=f.readlines()
    LineNum=1
    for line in lines:
        print(LineNum,line)
        LineNum+=1
    LineNum=int(input("Enter the Line Number to be
edited: "))
    print('1.Edit    2.Delete')
    while True:
        ch=int(input("Enter your choice: "))
        if ch==1:
            lines[LineNum-1]=input("Write New Line:
")
            print()
            ch=input("Save changes?(y/n): ")
            if ch.lower()=='y':
                fobj=open(N_name,'w+')
                fobj.writelines(lines)
                print("Changes Saved!")
```

```
                fobj.close()
            break
        elif ch==2:
            try:
                lines.pop(LineNum-1)
            except Exception:
                print('Entered wrong line number!')
                break
            ch=input("Save changes?(y/n): ")
            if ch.lower()=='y':
                fobj=open(N_name,'w+')
                fobj.writelines(lines)
                print("Changes Saved!")
                fobj.close()
            break
        else:
            print('Entered wrong choice!')
def Delete():
    NamesFile()
    Snotes=open('Notes',"rb")
    try:
        Notes=pickle.load(Snotes)
    except Exception:
        print()
    while True:
        if Notes==[]:
            print("No Notes saved!")
            break
        else:
            print("Saved Notes: ",Notes)
        N_name=input("Enter the name of the note to
be Deleted: ")
        ch=input('Do you want to delete Note?(y/n):
')
        Notes.remove(N_name)
        if ch.lower()=='y':
```

```
            try:
                os.remove(N_name)
                Snotes=open('Notes',"wb")
                pickle.dump(Notes,Snotes)
                Snotes.close()
            except Exception:
                print('Note not found!')
                break
            print('Note deleted successfully!')
            break
```

2)Contacts Module:

```
import pickle
def Main():

print('********************Contacts*****************
****')

    while True:
        print('''1. Display all Contacts.
2.Search Contact.
3.Add Contact.
4.Edit Contact.
5.Delete Contact.
6.Exit.
''')
        ch=int(input("Enter your choice: "))
        print()
        if ch==1:
```

```
            Display()
        elif ch==2:
            Search()
        elif ch==3:
            Add()
        elif ch==4:
            Edit()
        elif ch==5:
            Name=input("Enter the name of the
contact to be deleted:")
            Delete(Name)
        elif ch==6:
            break
        else:
            print('Entered wrong choice!')


def EmptyFile(): #Function to create empty file if
does not exists.

    try:
        Contacts=open("Contacts","rb")
    except IOError:
        Contacts=open("Contacts","wb+")
        L={}
        pickle.dump(L,Contacts)
        Contacts.close()
def Naming():    #Function to do the naming
conventions.
    Contacts=open('Contacts',"rb")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print('')
    C_name=input("New Contact Name: ")
    if C_name==str():
```

```
        C_name='NewContact'
    #Default Name for NewContact.
    #Also numbering the Contact with same name.
    k=1
    while True:
        if C_name in ContactList:
            if C_name[-1].isdigit():
                C_name=C_name[:-1]+str(k)
                k+=1
            else:
                C_name+=str(k)
                k+=1
        elif C_name not in ContactList:
            break
    return C_name
def MobileNumber():   #Function for taking Mobile
Number
    Contacts=open('Contacts',"rb")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print('')
    while True:
        try:
            Num=int(input("New Contact Number: "))
            l=len(str(Num))
            if l<10 or l>10:
                Num=0
                print('Entered wrong contact
number!')
                break
            Flag=0 #Check for the same contact
saved!
            for i in ContactList:
                if Num==ContactList[i]:
                    Flag=1
```

```
                    Saved=i
            if Flag==1:
                print("Number already saved as:
",Saved)
                break
        except Exception:
            print('Entered wrong contact number!')
            Num=0
            break
        break
    return Num
def Display():
    EmptyFile()
    Contacts=open("Contacts","rb+")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print()
    while True:
        if ContactList=={}:
            print("No Contact Saved!")
            break
        print('Saved Contacts: ')
        for i in sorted(ContactList):
            print(i,':',ContactList[i])
        break
    print()
def Search():
    EmptyFile()
    Contacts=open("Contacts","rb+")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print()
    while True:
        if ContactList=={}:
```

```
            print("No Contact Saved!")
            break
        print()
        Name=input("Enter the name of the contact
to be searched:")
        print('Contacts found:')
        print()
        try:
            for i in sorted(ContactList):
                if Name.lower() in i.lower() or
i.lower() in Name.lower():
                    print(i,':',ContactList[i])
            break
        except Exception:
            print('Contact not found!')
    print()
def Add():
    EmptyFile()
    Contacts=open("Contacts","rb+")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print()
    while True:
        try:
            Name=Naming()  #input("Name: ")
            Num=MobileNumber()
            if Num==0:
                break
            ch=input("Save contact?(y/n): ")
            if ch.lower()=='y':
                ContactList[Name]=Num
                Contacts=open("Contacts","wb")
                pickle.dump(ContactList,Contacts)
                Contacts.close()
                print("Contact Saved!")
```

```
                break
            else:
                print("Contact not saved!")
                break
        except Exception:
            print('Entered wrong contact number!')
            break
    print()
def Edit():
    EmptyFile()
    Name=input("Enter the name of the contact to be
edited:")
    while True:
        Contacts=open("Contacts","rb+")
        try:
            ContactList=pickle.load(Contacts)
        except Exception:
            print()
        try:
            print(Name,':',ContactList[Name])
        except Exception:
            print('Contact not found!')
            break
        print('''1.Edit(Name).
2.Edit(Number).
3.Delete Contact.
4.Exit.''')
        ch=int(input('Enter your choice: '))
        if ch==1:
            Num=ContactList[Name]
            del ContactList[Name]
            Name=Naming()
            ContactList[Name]=Num
            ch=input("Save edited contact?(y/n): ")
            if ch.lower()=='y':
                Contacts=open("Contacts","wb")
```

```
                pickle.dump(ContactList,Contacts)
                Contacts.close()
                print("Contact Saved!")
                continue
            else:
                print("Contact not edited!")
                continue
        elif ch==2:
            Num=MobileNumber()
            if Num==0:
                continue
            ContactList[Name]=Num
            ch=input("Save edited contact?(y/n): ")
            if ch.lower()=='y':
                Contacts=open("Contacts","wb")
                pickle.dump(ContactList,Contacts)
                Contacts.close()
                print("Contact Saved!")
                continue
            else:
                print("Contact not edited!")
                continue
        elif ch==3:
            Delete(Name)
            break
        elif ch==4:
            break
        else:
            print("Entered wrong choice!")
def Delete(Name):
    EmptyFile()
    Contacts=open("Contacts","rb+")
    try:
        ContactList=pickle.load(Contacts)
    except Exception:
        print()
```

```
    while True:
        try:
            print(Name,':',ContactList[Name])
            print()
            del ContactList[Name]
            ch=input('Delete Contact?(y/n): ')
            if ch.lower()=='y':
                Contacts=open("Contacts","wb")
                pickle.dump(ContactList,Contacts)
                Contacts.close()
                print("Contact Deleted!")
                break
            else:
                print("Not deleted!")
                break
        except Exception:
            print('Contact not found!')
            break
    print()
```

3)Calendar Module:

```
import calendar
def Main():

print('********************Calendar*****************
****')
    while True:
        print('''1.Year Calendar.
2.Month Calendar.
3.Exit.''')
        ch=int(input('Enter your choice: '))

        if ch==1:
            try:
                Year=int(input('Enter the calendar
```

```
year: '))
                print(calendar.calendar(Year))
            except Exception:
                print('Entered wrong Year!')
        elif ch==2:
            try:
                Year=int(input('Enter the calendar
year: '))
                Month=int(input('Enter the month
number: '))
                print(calendar.month(Year,Month))
            except Exception:
                print('Entered wrong Year/Month!')
        elif ch==3:
            break


        else:
            print('Entered wrong choice!')
```

4)Calculator Module:

```
def add(x,y):
    add=x+y
    return add
def sub(x,y):
    diff=x-y
    return diff
def mult(x,y):
    prod=x*y
    return prod
def div(x,y):
    quot=x/y
    return quot
def per(x,y):
```

```
    per=(x*100)/y
    return per
print('Note:- Input your number then operator(+,-
,*,/,%[Percentage]) then input your next number')
def Operators(num1,num2,oper):
    if oper=='+':
        out=add(num1,num2)
    elif oper=='-':
        out=sub(num1,num2)
    elif oper=='*':
        out=mult(num1,num2)
    elif oper=='/':
        out=div(num1,num2)
    elif oper=='%':
        out=per(num1,num2)
    else:
        print('Entered inappropriate operator!')
    return out
def Calculator():
    try:
        num1=int(input('Enter first number: '))
        oper=input('Enter the operator: ')
        num2=int(input('Enter second number: '))
    except Exception:
        print('Entered wrong figure!')
    Total=Operators(num1,num2,oper)
    while True:
        oper=input('Enter operator: ')
        if oper=='=':
            print("~~~~~~~~~~~~~~~")
            print("Result is :",Total)
            print("~~~~~~~~~~~~~~~")
            break
        num2=int(input('Enter number: '))
        Total=Operators(Total,num2,oper)
def Main():
```

```
    while True:
        try:
            print('''1.Calculate.
2.Exit.''')
            ch=int(input('Enter your choice: '))
            if ch==1:
                Calculator()
            elif ch==2:
                break
            else:
                print('Entered wrong choice!')
        except Exception:
            print('Something went wrong, try
again!')
```

Shortcoming

Our program is not the finished project, more features like advanced functioning of contacts and notepad, the calendar could have more features or functions, but due to less amount of time and boundaries in syllabus, our team cannot make it as perfect as we are using nowadays.

As our project is in the raw form so, user might find difficulty in understanding the program at the start.

Though the program cannot be made that perfect but our team has tried its best for the user to be useful.

Bibliography

- https://www.w3schools.com/python/
- https://www.geeksforgeeks.org/python-programming-language/
- https://www.programiz.com/python-programming