The Pentecostal Assembly School, Bokaro

COMPUTER SCIENCE PROJECT

LIBRARY MANAGEMENT SYSTEM

Name – Harshit Aditya

Class – 12/A

Board Roll Number –

Subject – Computer Science

Guide Teacher- M.K. Ojha Sir

Content:

Sl. No.	Index	Page No.
1.	Certificate	3
2.	Acknowledgement	4
3.	General Description	5
4.	Flow Chart	8
5.	Source code	10
6.	Output	31
7.	Bibliography	34

Certificate

This is certified to be the bonafide Project Work done by “Harshit Aditya” of class XII in The Pentecostal Assembly School , Bokaro during the academic year 2019-2020.

Date:

Teacher In charge: Mr. M.K. Ojha

Submitted for AISSCE practical examination held on

Project Title: Library Management System

Name of Candidate : Harshit Aditya

Board Roll Number:

Center Code:

Internal Signature External Signature

Acknowledgement

In the accomplishment of this project successfully, many people have best owned me their blessings and the heart pledged support.

I would like to thank my Principal Mrs. Rita Prasad and my Computer science teacher Mr. M.K. Ojha Sir, whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His suggestions and his instructions has served as the major contributor towards the completion of this project.

General Description

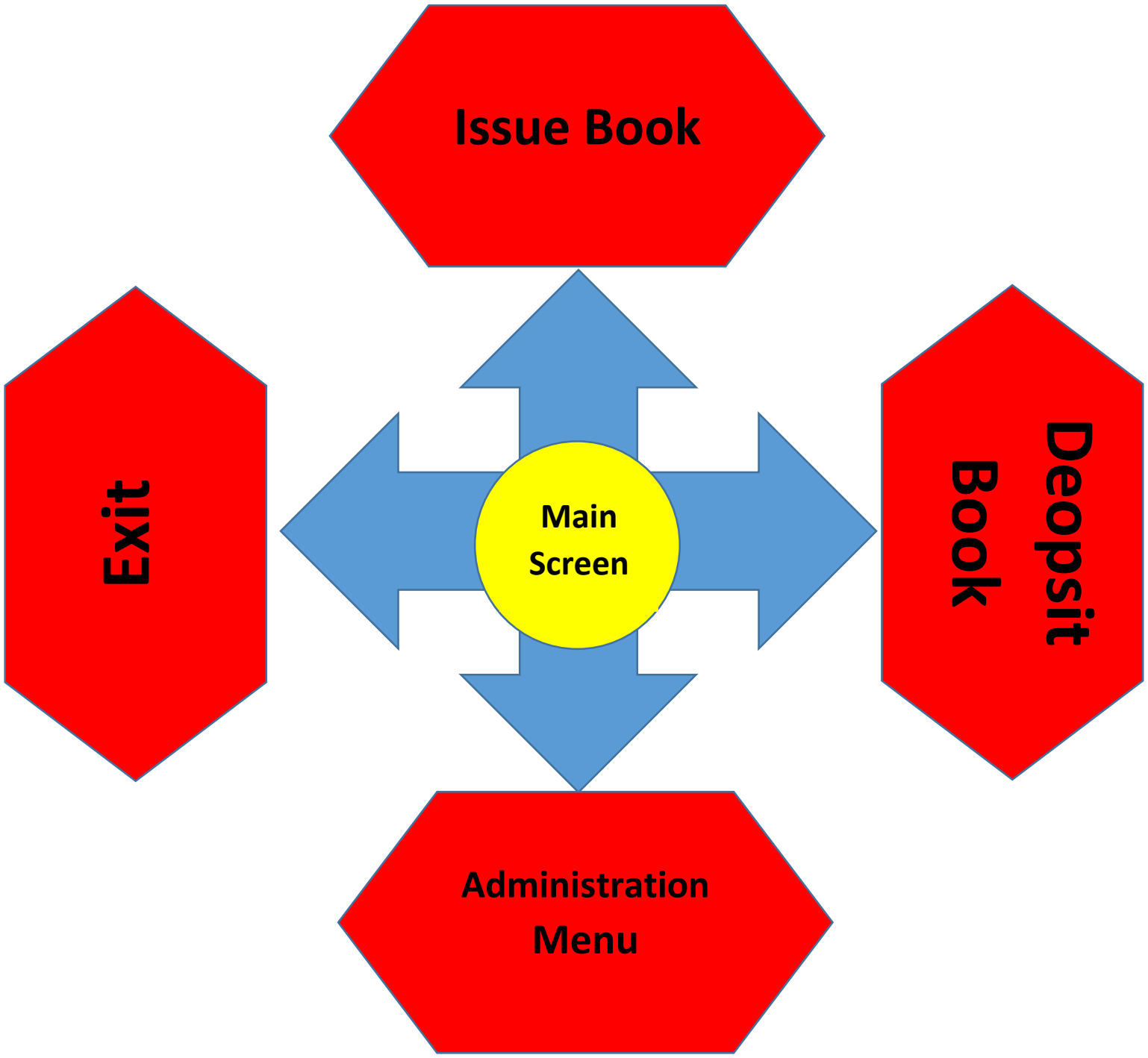
Library Management System is a software used to manage the catalogue of a library. This helps to keep the records of whole transactions of the books available in the library. Library Management System which is very easy to use and fulfills all the requirement of a librarian. There are many features to keep records of available books as well as issued books...

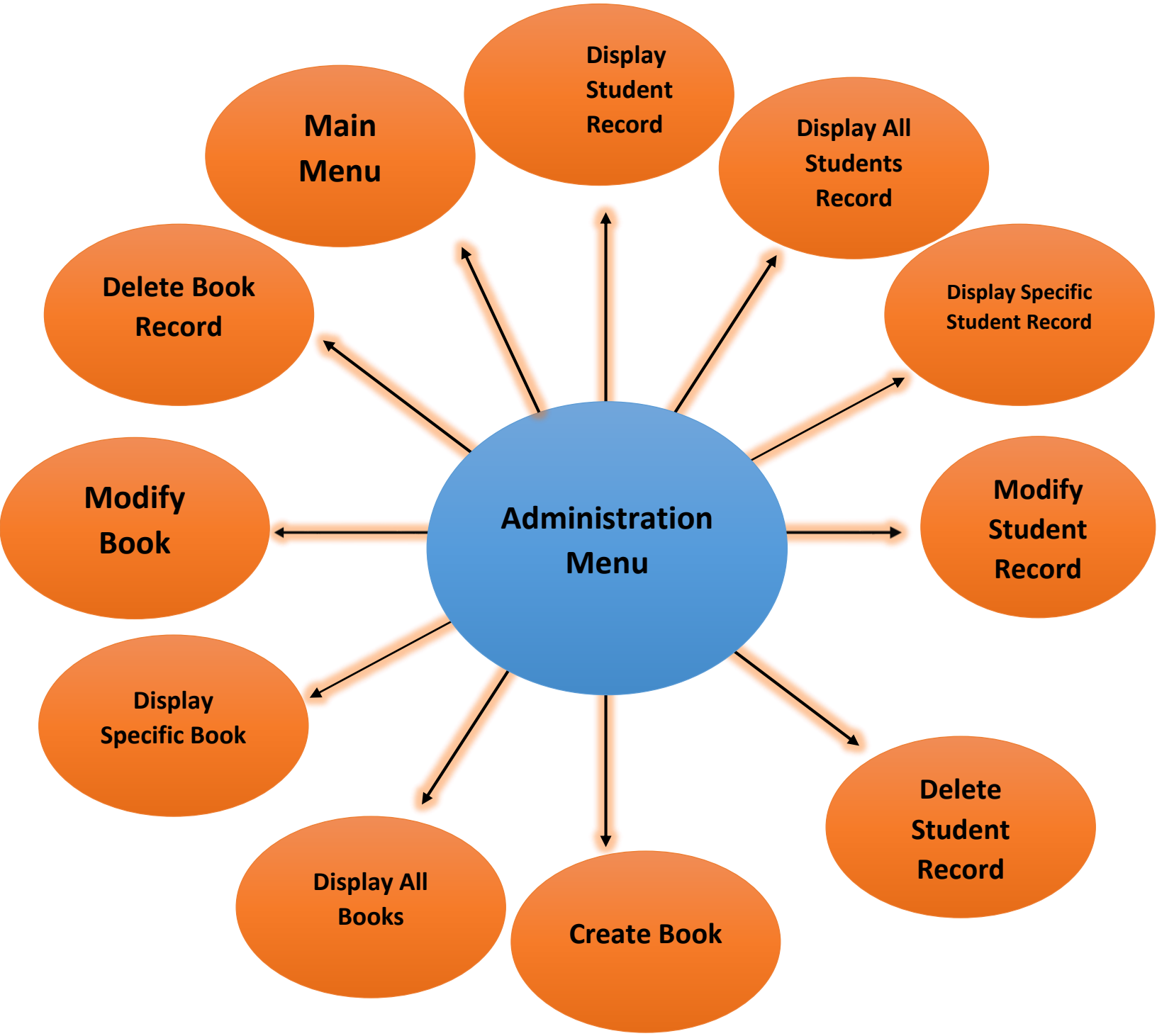
Features provided by Program:

- Easy way to enter new books.
- Keep record of complete information of books like; Book name, Author name
- Easy way to make a check-out
- Automatic fine calculation for late returns.
- Easy to enter new student record
- Easy way to modify or delete a student/ book record
- And much more

Files Generated by the Program

1. Student1.dat- File to save Student information
2. Book1.dat- File to save book information
3. Temp.dat- File to store issued books information


Issue Book
Exit
Main Screen
Deopsit Book
Administration Menu


Administration Menu
Main Menu
Display Student Record
Display All Students Record
Display Specific Student Record
Modify Student Record
Delete Student Record
Create Book
Display All Books
Display Specific Book
Modify Book
Delete Book Record

Source Code

```
from pickle import load,dump
from os import remove,rename
import os
import random
import time
print()
print()
print()
print()
print()
print ("\t \t \t WELCOME TO THE PENTECOSTAL ASSEMBLY SCHOOL LIBRARY \t \t \t")
print()
print()
print()
print()
print()
print()
print()
print ("\t \t \t \t \t \t \t PROJECT DONE BY : HARSHIT ADITYA")
print()
print()
print("\t \t \t \t \t \t \..... ")
print()
```

```
print("\t \t \t \t \t \t \t……. ")
print()
print()
a=input("PRESS ENTER TO CONTINUE:")
a=str(a)
if a.isalpha():
   pass
"""*******************************************************************************
                                    CLASS USED IN PROJECT
*******************************************************************************"""
class book:
   def __init__(self,bno=" ",bname=" ",auname=" "):
      self.bno=bno
      self.bname=bname
      self.auname=auname
   def create_book(self):
      print()
      print("\t\t\t Creating Book Record \t\t\t")
      print()
      self.bno=input("\t Enter Book Number:")
      print()
      self.bname=input("\t Enter Name of the Book:")
      print()
      self.auname=input("\t Enter Name of the Author:")
      print()
      print()
      print("\t \t \t Book Created \t \t \t")

   def show_book(self):
```

```
        print()
        print()
        print("\t \t Book Number:",self.bno)
        print()
        print ("\t \t Book Name:",self.bname)
        print()
        print ("\t \t Author Name:",self.auname)
        print()
        print()
    def modify_book(self):
        print()
        print()
        print ("\t \t Book No.: ",self.bno)
        print()
        self.bname=input("\t \t Enter New Book Name:")
        print()
        self.auname=input("\t \t Enter New Author Name:")
        print()
        print()
        print("\t \t Book Modified")
        print()
    def ret_bno(self):
        return (self.bno)
    def report_book(self):
        print(self.bno,self.bname,self.auname)
class student:
    def __init__(self,admno=" ",name=" ",stbno=" ",token=0):
        self.admno=admno
        self.name=name
```

```
    self.stbno=stbno
    self.token=token
def createstud(self):
    print()
    print("\t \t \t Creating Student Record \t \t \t")
    print()
    self.admno=input("\t \t Enter Admission Number:")
    print()
    self.name=input("\t \t Enter Name of the Student:")
    self.stbno=" "
    self.token=0
    print()
    print()
    print("\t \t \t Student Record Created \t \t \t")
    print()
    print("#"*70)
    print()
def showstud(self):
    print()
    print()
    print("\t Admission No.:",self.admno)
    print()
    print("\t Name:",self.name)
    print()
    print ("\t Stbno:",self.stbno)
    print()
    print("\t Token:",self.token)
    print()
```

```
def displaystud(self):
    print()
    print ("\t Admission Number of the Student is:",self.admno)
    print()
    print ("\t Name of the Student is:",self.name)
    if (self.token==1):
        print( "\t Book Number is:",self.stbno)
def modifystud(self):
    print()
    print ("\t Admission No:",self.admno)
    print()
    self.name=input("\t New Student Name:")
    print()
    print ("\t \t Student's Name Modified !!")
def ret_admno(self):
    return self.admno

def ret_stbno(self):
    return self.stbno
def ret_token(self):
    return self.token
def add_token(self):
    self.token=1
def reset_token(self):
    self.token=0
def get_stbno(self,t):
    self.stbno=t
def reportstud(self):
    print (self.admno, self.name, self.token)
```

```
"""*****************************************************************************
                        FUNCTION TO CREATE BOOK
*****************************************************************************"""
def writebook():
   ch="Y"
   fp=open("book1.dat","ab")
   while ch=="Y":
      bk.create_book()
      dump(bk,fp)
      print()
      ch=input("\t \t Do You Want to Continue<y/n>:")
      print()
      print ("#"*70)
      print()
      ch=ch.upper()
"""*****************************************************************************
                        FUNCTION TO ENTER STUDENT RECORD
*****************************************************************************"""
def writestudent():
   ch="Y"
   fp=open("student1.dat","ab")
   while ch=="Y":
      st.createstud()
      dump(st,fp)
      ch=input("\t \t Do You Want to Continue <y/n>:")
      ch=ch.upper()
      print()
```

```
"""*****************************************************************************
                        FUNCION TO DISPLAY ALL STUDENT LIST
*****************************************************************************"""
def display_alls():
    fin=open("student1.dat","rb")
    if not (fin):
        print ("\t \t File is Not Found..")
    else:
        try:
            while True:
                print()
                st=load(fin)
                st.showstud()
        except EOFError:
            pass
        fin.close()
"""*****************************************************************************
                        FUNCTION TO DISPLAY ALL BOOKS RECORD
*****************************************************************************"""
def display_allb():
    fin=open("book1.dat","rb")
    if not (fin):
        print()
        print()
        print ("Book File is Not Found..")
    else:
        try:
            while True:
```

```
            bk=load(fin)
            bk.show_book()
        except EOFError:
            pass
        fin.close()
"""*****************************************************************************
                        FUNCTION TO DISPLAY PARTICULAR BOOKS RECORD
*****************************************************************************"""
def display_spb(no):
    flag=0
    fin=open("book1.dat","rb")
    try:
        while True:
            bk=load(fin)
            if(bk.ret_bno()==no):
                bk.show_book()
                flag=1

    except EOFError:
        pass
    fin.close()
    if flag==0:
        print()
        print()
        print ("\t \t \t \t BOOK NOT PRESENT..!!")
"""*****************************************************************************
                        FUNCTION TO DISPLAY PARTICULAR STUDENT RECORD
*****************************************************************************"""
def display_sps(n):
```

```
    flag=0
    fin=open("student1.dat","rb")
    try:
        while True:
            st=load(fin)
            if(st.ret_admno()==n):
                st.showstud()
                flag=1
    except EOFError:
        pass
    fin.close()
    if flag==0:
        print()
        print ("\t \t \t STUDENT NOT PRESENT..!!")
"""*****************************************************************************
                                FUNCTION TO MODIFY BOOK RECORD
*****************************************************************************"""
def modify_bookrecord():
    found=0
    print()
    print()
    print ("\t \t \t Modify Book")
    print()
    print()
    n=input("\t \t Enter Book Number to be Modified:")
    print()
    fin=open("book1.dat","rb")
    fout=open("temp.dat","wb")
    try:
```

```
        while True:
            bk=load(fin)
            if bk.ret_bno()==n:
                print()
                print ("\t \t \t Book Details")
                bk.show_book()
                print()
                print("\t \t  Enter New Details")
                print()
                print()
                bk.modify_book()
                dump(bk,fout)
                found=1
            else:
                dump(bk,fout)
    except EOFError:
        pass
    if found==0:
        print("\t \t \t \t Book Not Present")
    fin.close()
    fout.close()
    remove("book1.dat")
    rename("temp.dat","book1.dat")
"""**********************************************************************************
                                FUNCTION TO MODIFY STUDENT RECORD
**********************************************************************************"""
def modify_student_record():
    found=0
    print()
```

```
print("\t \t \t Modify Student Record")
print()
print()
n=input("\t \t Enter Student's Admission Number to be Modified:")
print()
fin=open("student1.dat","rb")
fout=open("temp.dat","wb")
try:
   while True:
      st=load(fin)
      if st.ret_admno()==n:
         print()
         print("\t \t \t STUDENT DETAILS")
         st.showstud()
         print()
         print("\t \t \t Enter New Student Details:")
         st.modifystud()
         dump(st,fout)
         found=1
      else:
         dump(st,fout)
except EOFError:
   pass
if found==0:
   print("\t \t \t \t STUDENT NOT PRESENT")
fin.close()
fout.close()
remove("student1.dat")
rename("temp.dat","student1.dat")
```

```
"""*****************************************************************************
                        FUNCTION TO DELETE STUDENT RECORD
*****************************************************************************"""
def del_stud():
    flag=0
    print()
    print()
    n=input("\t \t Enter Admission to be Deleted:")
    print()
    fin=open("student1.dat","rb")
    fout=open("temp.dat","wb")
    try:
        while True:
            st=load(fin)
            if st.ret_admno()<n:
                dump(st,fout)
            else:
                flag=1
    except EOFError:
        pass
    fin.close()
    fout.close()
    remove("student1.dat")
    rename("temp.dat","student1.dat")
    if flag==1:
        print()
        print("\t \t \t \t RECORD DELETED..!!")
    else:
        print()
```

```
    print("\t \t \t \t SORYY..!! RECORD DOES NOT EXIST..!!..")



"""*****************************************************************************
                                FUNCTION TO DELETE BOOK RECORD
*****************************************************************************"""
def del_book():
   flag=0
   print()
   print()
   n=input("\t \t Enter Book No to be Deleted:")
   print()
   fin=open("book1.dat","rb")
   fout=open("temp.dat","wb")
   try:
      while True:
         bk=load(fin)
         if bk.ret_bno()<n:
            dump(bk,fout)
         else:
            flag=1
   except EOFError:
      pass
   fin.close()
   fout.close()
   remove("book1.dat")
   rename("temp.dat","book1.dat")
   if flag==1:
      print( "\t \t \t Record Deleted")
```

```
    else:
        print("\t \t \t SORRY..!! RECORD NOT PRESENT..")
"""*****************************************************************************
                                FUNCTION TO ISSUE BOOK
*****************************************************************************"""
def book_issue():
    sn=" "
    bn=" "
    found=0
    flag=0
    print()
    print()
    print ("\t \t \t BOOK ISSUE.. \t \t \t")
    print()
    print()
    sn=input("\t \t Enter the Student's Admission Number:")
    print()
    fin1=open("book1.dat","rb")
    fin2=open("student1.dat","rb")
    fout=open("temp.dat","ab")
    try:
        while True:
            st=load(fin2)
            if (st.ret_admno()==sn):
                st.showstud()
                found=1
                if st.ret_token()==0:
                    bn=input("\t \t Enter Book Number:")
                    try:
```

```
            while True:
                bk=load(fin1)
                if bk.ret_bno()==bn:
                    bk.show_book()
                    flag=1
                    st.add_token()
                    st.get_stbno(bk.ret_bno())
                    dump(st,fout)
                    os.system("cls")
                    print()
                    print()
                    print("\t \t \t Book Issued Successfully \t \t \t")
                    print()
                    print("\t PLEASE NOTE : Write the current date in backside of your book ")
                    print( "\t \t and submit within 15 days")
                    print ()
                    print("\t \t Fine Rs.20 for each day after 15 days period..!!")
                    print()

            except EOFError:
                pass
        else:
            print("\t You have not returned the last book..")
    except EOFError:
        pass

"""**********************************************************************************
                                FUNCTION TO DEPOSIT BOOK
**********************************************************************************"""
```

```
def book_deposit():
    print("\t \t \t BOOK DEPOSITING.")
    sn=" "
    found=0
    flag=0
    day=0
    fine=0
    print()
    print()
    sn=input("\t \t \t Enter Students Admission Number:")
    print()
    fin1=open("student1.dat","rb")
    fin2=open("book1.dat","rb")
    fout=open("temp.dat","rb")
    try:
        while True:
            st=load(fout)
            if st.ret_admno()==sn:
                found=1
                print()
                print("\t Student Token Number",st.ret_token())
                if st.ret_token()==1:
                    try:
                        while True:
                            bk=load(fin2)
                            if bk.ret_bno()==st.ret_stbno():
                                bk.show_book()
                                flag=1
                                print()
```

```
                    days=int(input("\t Book Deposited In no. of days:"))
                    if days>=15:
                       fine=(days-15)*20
                       print()
                       print("\t fine : Rs.",fine)
                    st.reset_token()
                    st.get_stbno(0)
                    st.showstud()
                    print()
                    print("\t \t BOOK DEPOSITED !!!")
                except EOFError:
                   pass
             else:
                print()
                print("\t \t You have not issued the book..")
    except EOFError:
       pass
    if(found==0):
       print()
       print("\t No Such Student Exists")
    fin1.close()
    fin2.close()
    fout.close()
bk=book()
st=student()
"""*****************************************************************************
                                        ADMINISTRATOR MENU
*****************************************************************************"""
```

def adminmenu():

```
ch="Y"
while ch=="Y":
    print()
    print()
    print ("\t \t \t ADMINISTRATION MENU \t \t \t")
    print()
    print()
    print ("\t 1. CREATE STUDENT RECORD")
    print()
    print ("\t 2. DISPLAY ALL STUDENTS RECORD")
    print()
    print ("\t 3. DISPLAY SPECIFIC STUDENT RECORD")
    print()
    print ("\t 4. MODIFY STUDENT RECORD")
    print()
    print ("\t 5. DELETE STUDENT RECORD")
    print()
    print ("\t 6. CREATE BOOK")
    print()
    print("\t 7. DISPLAY ALL BOOKS")
    print()
    print("\t 8. DISPLAY SPECIFIC BOOK")
    print()
    print("\t 9. MODIFY BOOK")
    print()
    print("\t 10.DELETE BOOK RECORD")
    print()
    ch1=int(input("\t \t Enter Your Choice:"))
    print()
```

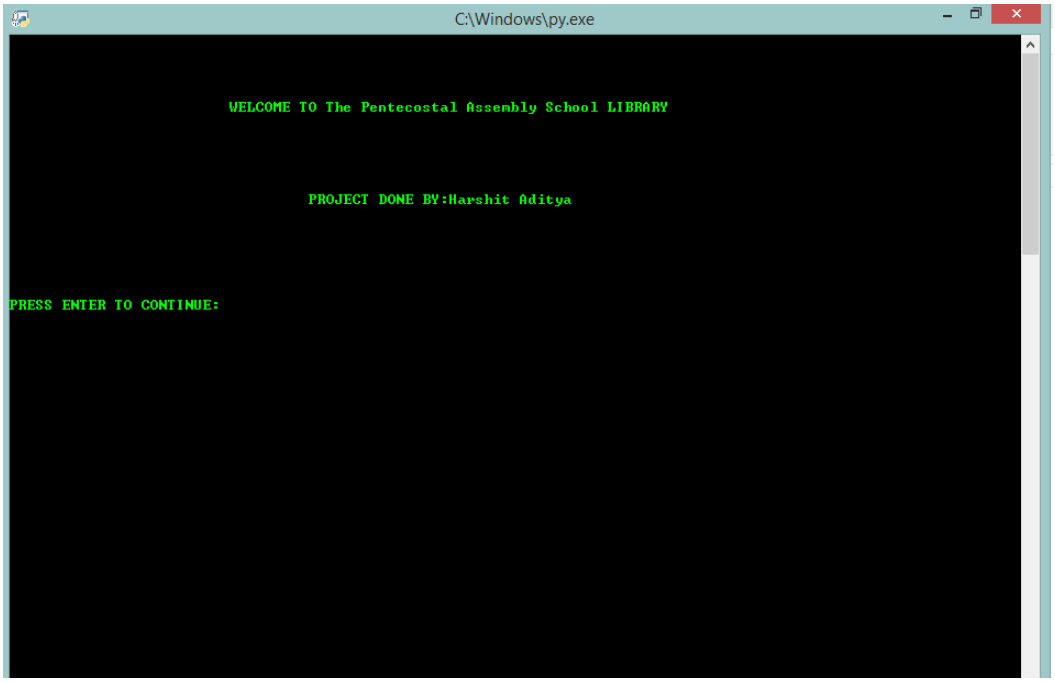
```
os.system("cls")
if ch1==1:
   writestudent()
elif ch1==2:
   display_alls()
elif ch1==3:
   print()
   print()
   ad=input("\t \t Enter Student's Admno to be Displayed:")
   display_sps(ad)
elif ch1==4:
   modify_student_record()
elif ch1==5:
   del_stud()
elif ch1==6:
   writebook()
elif ch1==7:
   display_allb()
elif ch1==8:
   print()
   print()
   bn=input("\t \t ENTER BOOK NUMBER TO BE DISPLAYED:")
   display_spb(bn)
elif ch1==9:
   modify_bookrecord()
elif ch1==10:
   del_book()
print()
ch=input("\t \t Do you want Continue with ADMINMENU<y/n>:")
```

```
    ch=ch.upper()
    print()
    os.system("cls")
    if ch=="Y":
       continue
    else:
       mainmenu()

"""*********************************************************************************
                                    MAIN MENU
*********************************************************************************"""
def mainmenu():
   ch="Y"
   while ch=="Y":
      print()
      print()
      print("\t \t \t MAIN MENU \t \t \t")
      print()
      print("\t 1. BOOK ISSUE")
      print()
      print("\t 2. BOOK DEPOSIT")
      print()
      print("\t 3. ADMINISTRATION MENU")
      print()
      print("\t 4. EXIT")
      print()
      ch1=int(input("\t \t Enter Your Choice:"))
      print()
      print("\t \t Loading ....")
```

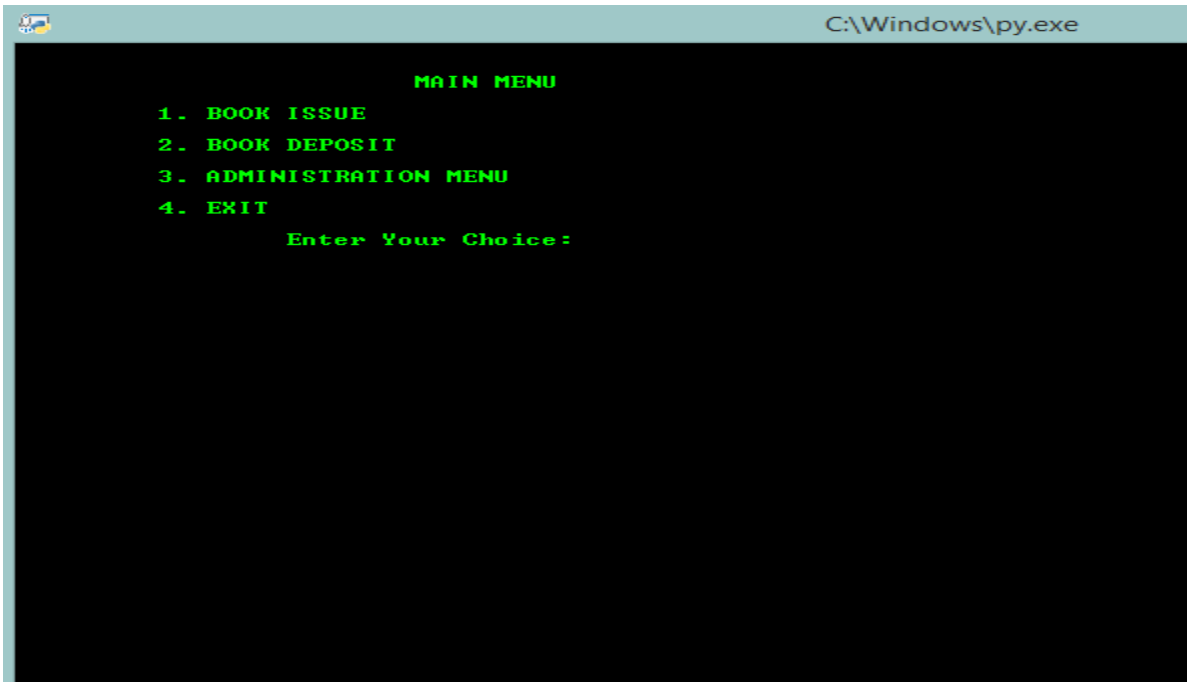
```
        os.system("cls")
        if ch1==1:
            book_issue()
        elif ch1==2:
            book_deposit()
        elif ch1==3:
            adminmenu()
        else:
            exit(0)
        print()
        ch=input("\t \t \t Do You Want to Continue <y/n>:")
        ch=ch.upper()
        if  ch=="N":
            break
        os.system("cls")

os.system("cls")
mainmenu()
```

OUTPUT

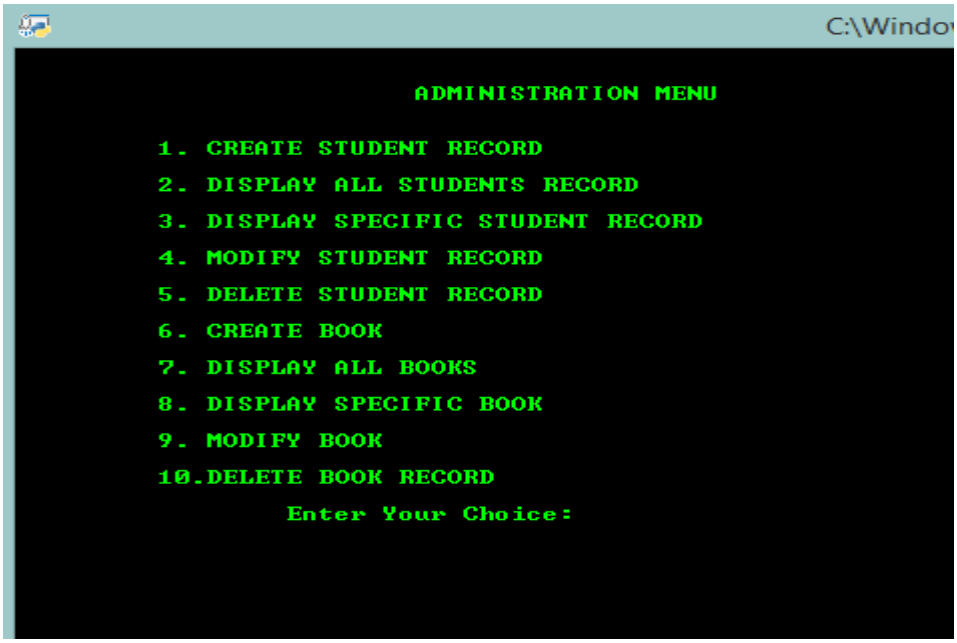
1. Introduction



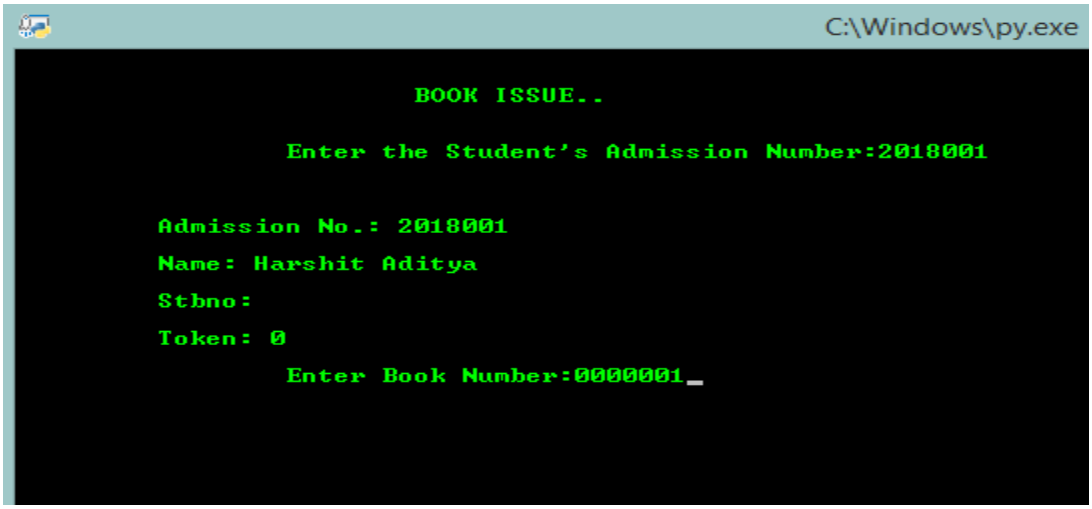
2. Main Menu

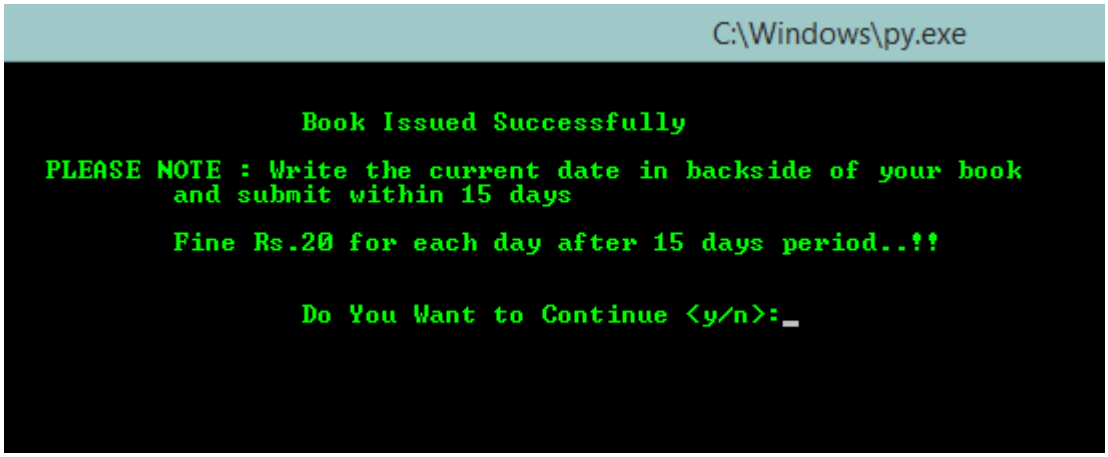


3. Administration Menu

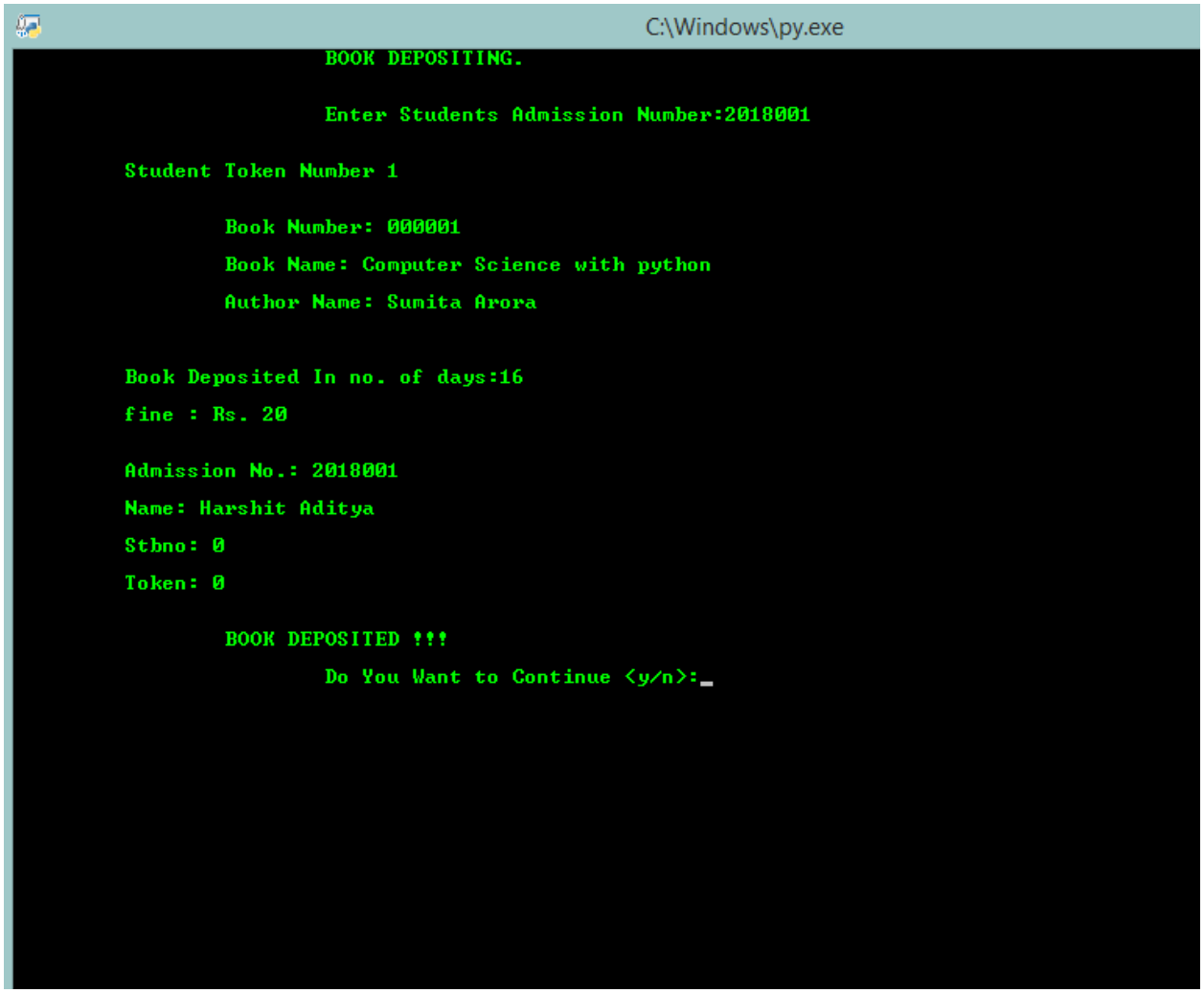


4. Issue Book





5. Deposit Book



Bibliography

Help from these resources

- Computer Science with Python – Sumita Arora
- http://www.github.com
- www.google.com
- http://slideshare.in