COMPUTER SCIENCE

BANKING PROJECT FILE

**(SESSION- 2020-21)**

Prepared By: - Abhishek Yadav

Class : - XII

Under Guidance of: - Dr. Subhash

**CONTENT**

|  |  |  |
| --- | --- | --- |
| **S No.** | **CONTENT** | **PG No.** |
| **1** | Certificate | 3 |
| **2** | Acknowledgement | 4 |
| **3** | About Python | 5 |
| **4** | File handling in python | 6 |
| **5** | About Modules | 7 |
| **6** | Introduction | 8 |
| **7** | Project Source Code | 10 |
| **8** | Requirements | 20 |
| **9** | Bibliogrphy | 21 |

**Certificate**

This is to certify that **Abhishek Yadav**, student of class XII has successfully completed his python assignment session 2020-2021.

I certify that this project is up to my expectations and as per the guidelines issued by CBSE.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my computer teacher for their able guidance and support in completing my project.

I would also like to express my gratitude to the principal for providing me with all facility that was required.

ABOUT PYTHON

Python is a high level, object-oriented programming language. It was developed in 1991 by Guido Van Rossum. Its syntax is similar to the English Language and that is why it enhances code readability. It uses indentation for defining scopes of loops if-else construct, class, etc.

Python Logo - File Handling in Python

Python can be used to create server-side applications.

Python can help to do task automation using scripting.

Python is used to create stand-alone applications.

Python is used in Big Data, data science, Machine Learning etc.

File handling in python

If you are working in a large software application where they process a large number of data, then we cannot expect those data to be stored in a variable as the variables are volatile in nature.

Hence when are you about to handle such situations, the role of files will come into the picture,

As files are non-volatile in nature, the data will be stored permanently in a secondary device like Hard Disk and using python we will handle these files in our applications.

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files.

ABOUT MODULES

Modules refer to a file containing Python statements and definitions. A file containing Python code, for e.g.: example.py, is called a module and its module name would be example.

We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code.

We can define our most used functions in a module and import it. instead of copying their definitions into different programs. We can import the definitions inside a module to another module or the interactive interpreter in Python.

We use the import keyword to do this.

Python has a ton of standard modules available.

INTODUCTION

Banking Project in Python

**The Problem**

Customer experience is an integral part of a bank’s operations. That’s why banks focus a lot on improving customer experience by removing hassles and enhancing the facilities they provide. Opening a new account in a bank usually requires a person to visit the bank, fill out a form, and submit the necessary papers. All of these tasks take up a lot of time and dampen the overall customer experience. Moreover, many people have to take time out of their schedules to go to a bank.

**The Solution**

You can solve this problem by creating a software solution where people can sign up and open a new account in a bank digitally. This way, the person wouldn’t have to visit the bank physically and thus, would save a lot of time and effort. The banking management system can also allow the user to make transactions, deposit and withdraw funds, and check the account balance.

Your solution would need an admin section which would look after the users’ accounts and the overall wellbeing of the database. You’ll have to connect the software to a database which will store all user information in distinct storage.

## ****Database Management****

A large section of our project focuses on [database management](https://www.upgrad.com/blog/dbms-project-ideas-for-beginners/). You’ll have to create a database for the banking solution to facilitate its functioning. You should be familiar with the basics of database management.

**PROJECT SOURCE CODE**

**import pickle**

**import os**

**import pathlib**

**class Account :**

**accNo = 0**

**name = ''**

**deposit=0**

**type = ''**

**def createAccount(self):**

**self.accNo= int(input("Enter the account no : "))**

**self.name = input("Enter the account holder name : ")**

**self.type = input("Ente the type of account [C/S] : ")**

**self.deposit = int(input("Enter The Initial amount(>=500 for Saving and >=1000 for current):"))**

**print("\n\n\nAccount Created")**

**def showAccount(self):**

**print("Account Number : ",self.accNo)**

**print("Account Holder Name : ", self.name)**

**print("Type of Account",self.type)**

**print("Balance : ",self.deposit)**

**def modifyAccount(self):**

**print("Account Number : ",self.accNo)**

**self.name = input("Modify Account Holder Name :")**

**self.type = input("Modify type of Account :")**

**self.deposit = int(input("Modify Balance :"))**

**def depositAmount(self,amount):**

**self.deposit += amount**

**def withdrawAmount(self,amount):**

**self.deposit -= amount**

**def report(self):**

**print(self.accNo, " ",self.name ," ",self.type," ", self.deposit)**

**def getAccountNo(self):**

**return self.accNo**

**def getAcccountHolderName(self):**

**return self.name**

**def getAccountType(self):**

**return self.type**

**def getDeposit(self):**

**return self.deposit**

**def intro():**

**print("\t\t\t\*\*\*\*\*\*\*\*BANK MANAGEMENT SYSTEM\*\*\*\*\*\*\*\*")**

**print("")**

**print("\t\t\t\tBrought To You By:")**

**print("\t\t\t\t Abhishek Yadav")**

**print("\t\t\t\t School: Chinmaya Vidyalaya Unchahar")**

**print("\t\t\t\t Roll No.: 03")**

**print("")**

**input("Press any key to continue......")**

**def writeAccount():**

**account = Account()**

**account.createAccount()**

**writeAccountsFile(account)**

**def displayAll():**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**mylist = pickle.load(infile)**

**for item in mylist :**

**print(item.accNo," ", item.name, " ",item.type, " ",item.deposit )**

**infile.close()**

**else :**

**print("No records to display")**

**def displaySp(num):**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**mylist = pickle.load(infile)**

**infile.close()**

**found = False**

**for item in mylist :**

**if item.accNo == num :**

**print("Your account Balance is = ",item.deposit)**

**found = True**

**else :**

**print("No records to Search")**

**if not found :**

**print("No existing record with this number")**

**def depositAndWithdraw(num1,num2):**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**mylist = pickle.load(infile)**

**infile.close()**

**os.remove('accounts.data')**

**for item in mylist :**

**if item.accNo == num1 :**

**if num2 == 1 :**

**amount = int(input("Enter the amount to deposit : "))**

**item.deposit += amount**

**print("Your account is updted")**

**elif num2 == 2 :**

**amount = int(input("Enter the amount to withdraw : "))**

**if amount <= item.deposit :**

**item.deposit -=amount**

**else :**

**print("You cannot withdraw larger amount")**

**else :**

**print("No records to Search")**

**outfile = open('newaccounts.data','wb')**

**pickle.dump(mylist, outfile)**

**outfile.close()**

**os.rename('newaccounts.data', 'accounts.data')**

**def deleteAccount(num):**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**oldlist = pickle.load(infile)**

**infile.close()**

**newlist = []**

**for item in oldlist :**

**if item.accNo != num :**

**newlist.append(item)**

**os.remove('accounts.data')**

**outfile = open('newaccounts.data','wb')**

**pickle.dump(newlist, outfile)**

**outfile.close()**

**os.rename('newaccounts.data', 'accounts.data')**

**def modifyAccount(num):**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**oldlist = pickle.load(infile)**

**infile.close()**

**os.remove('accounts.data')**

**for item in oldlist :**

**if item.accNo == num :**

**item.name = input("Enter the account holder name : ")**

**item.type = input("Enter the account Type : ")**

**item.deposit = int(input("Enter the Amount : "))**

**outfile = open('newaccounts.data','wb')**

**pickle.dump(oldlist, outfile)**

**outfile.close()**

**os.rename('newaccounts.data', 'accounts.data')**

**def writeAccountsFile(account) :**

**file = pathlib.Path("accounts.data")**

**if file.exists ():**

**infile = open('accounts.data','rb')**

**oldlist = pickle.load(infile)**

**oldlist.append(account)**

**infile.close()**

**os.remove('accounts.data')**

**else :**

**oldlist = [account]**

**outfile = open('newaccounts.data','wb')**

**pickle.dump(oldlist, outfile)**

**outfile.close()**

**os.rename('newaccounts.data', 'accounts.data')**

**# start of the program**

**ch=''**

**num=0**

**intro()**

**while ch != 8:**

**print("\tMAIN MENU")**

**print("\t1. NEW ACCOUNT")**

**print("\t2. DEPOSIT AMOUNT")**

**print("\t3. WITHDRAW AMOUNT")**

**print("\t4. BALANCE ENQUIRY")**

**print("\t5. ALL ACCOUNT HOLDER LIST")**

**print("\t6. CLOSE AN ACCOUNT")**

**print("\t7. MODIFY AN ACCOUNT")**

**print("\t8. EXIT")**

**print("\tSelect Your Option (1-8) ")**

**ch = input()**

**if ch == '1':**

**writeAccount()**

**elif ch =='2':**

**num = int(input("\tEnter The account No. : "))**

**depositAndWithdraw(num, 1)**

**elif ch == '3':**

**num = int(input("\tEnter The account No. : "))**

**depositAndWithdraw(num, 2)**

**elif ch == '4':**

**num = int(input("\tEnter The account No. : "))**

**displaySp(num)**

**elif ch == '5':**

**displayAll();**

**elif ch == '6':**

**num =int(input("\tEnter The account No. : "))**

**deleteAccount(num)**

**elif ch == '7':**

**num = int(input("\tEnter The account No. : "))**

**modifyAccount(num)**

**elif ch == '8':**

**print("\tThanks for using bank managemnt system")**

**break**

**else :**

**print("Invalid choice")**

**ch = input("Press any key ......")**

REQUIREMENTS

HARDWARE REQUIREMENTS

1. **Computer** for coding and typing the required documents of the project
2. **Printer**, to print the required documents of the project.
3. **Compact drive**
4. **Processor:** Pentium quad core
5. **Ram; 64 GB**
6. **Hard disk: 5 TB**

SOFTWARE REQUIREMENTS

Python 3.3 Or higher.

Python IDLE

Computer with 125 MB of RAM

Operating system: Windows 7 or higher

BIBILIOGRAPHY

Sumita Arora’s Book for CS Class XI

Sumita Arora’s Book for CS Class XII

GitHub

StackExchange

Gradup.com