LAB 2

19F-0151

Ahmad Raza

[Company name]  [Company address]

Task no 1:

Code:

import random  
from datetime import datetime, timedelta  
  
class librarySystem():  
  
 data = dict()  
 data = ["Ahmad raza", "19F-0151",  
 "Saboor", "19F-0252"]  
  
 books = dict()  
 books = ["CN","AI"]  
 def display(self):  
 print("Books Available: ")  
 for books in self.books:  
 print(books)  
 def checkstudent(self,name,rollno):  
 for user in self.data:  
 if rollno in self.data:  
 print("User Exists")  
 print("Hello ", name, rollno)  
 self.display(self)  
 self.issueBook(self)  
 break  
  
 else:  
 self.AddStudent(self, name, rollno)  
 print("User Added")  
 self.display(self)  
 self.issueBook(self)  
 break  
  
 def AddStudent(self,name,rollno):  
 self.data.append(name)  
 self.data.append(rollno)  
  
 def issueBook(self):  
 bookname=input("Enter Book name to select: ")  
 for books in self.books:  
 if bookname in self.books:  
 print("Book Issued: ", bookname)  
 print("\tReturn Date: ",self.gen\_datetime())  
 break  
 else:  
 print("Book Not Found")  
 break  
  
 def gen\_datetime(min\_year=2022, max\_year=datetime.now().year):  
 *# generate a datetime in format yyyy-mm-dd hh:mm:ss.00000* start = datetime(min\_year, 1, 1, 00, 00, 00)  
 years = max\_year - min\_year + 1  
 end = start + timedelta(days=365 \* years)  
 return start + (end - start) \* random.random()  
  
  
def main():  
 obj=librarySystem  
 while(1):  
 choice=input("Select An Option:\n\t""1.Login\n\t0.Exit\n\tChoice: ")  
 if choice == '1':  
 name=input("Enter Name: ")  
 rollno=input("Enter Roll no: ")  
 obj.checkstudent(obj,name,rollno)  
 elif choice == '0':  
 quit()  
  
main()

Output:

Graphical user interface, text, application

Description automatically generated

Task no 2:

Code:

class AdjacencyList(object):  
  
 def \_\_init\_\_(self):  
 self.List = {}  
  
 def addEdge(self, fromVertex, toVertex):  
 *# check if vertex is already present* if fromVertex in self.List.keys():  
 self.List[fromVertex].append(toVertex)  
 else:  
 self.List[fromVertex] = [toVertex]  
  
 def printList(self):  
 for i in self.List:  
 print(i, '->', ' -> '.join([str(j) for j in self.List[i]]))  
  
  
if \_\_name\_\_=='\_\_main\_\_':  
 print("Graph: ")  
  
 obj = AdjacencyList()  
 obj.addEdge(0, 1)  
 obj.addEdge(0, 2)  
 obj.addEdge(2, 1)  
 obj.addEdge(2, 3)  
 obj.addEdge(1, 3)  
 obj.addEdge(1, 4)  
 obj.addEdge(1, 2)  
 obj.addEdge(3, 2)  
 obj.addEdge(3, 4)  
  
 obj.printList()

Output:

