***BLOOD DONATING APP***

* **INTRODUCTION:**

We have created a blood donating app in order to provide the fast donation of blood. This app would contain the data of all types of blood the people are having. The blood of donor would be there and according to the receiver if their blood type matches than the blood would be donated as fast as it can according to the desired location. This app would automatically allocate rest time to the donors who had donated the blood.

* **SCOPE:**

If this app would spread than many lives can be saved as the finding of the perfect blood for donor and the receiver would be there. A healthy lifestyle would be adopted and for future emergencies there will be no hustle and bustle.

* **LANGUAGE USED:**

The language used in this is python with the help of the Kivy library.

* **CODE:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | import kivy | |  | from kivy.app import App | |  | from kivy.uix.button import Button | |  | from kivy.uix.gridlayout import GridLayout | |  | from kivy.uix.textinput import TextInput | |  | from kivy.uix.label import Label | |  |  | |  | #Layout | |  | class grid\_layout\_used(GridLayout): | |  | #counstructor | |  | def \_\_init\_\_(self, \*\*kwargs): | |  | #base class constructor | |  | super().\_\_init\_\_(\*\*kwargs) | |  | self.start\_tab() | |  | return | |  | #starting or program | |  | def start\_tab(self): | |  | #clearing screen | |  | self.clear\_widgets() | |  | self.cols = 1 | |  | self.donor\_button = Button(text="DONOR INFO", background\_color =(0,1,1,1),color =(1, 1, 1, 1)) | |  | self.donor\_button.bind(on\_press=self.on\_click\_donor) | |  | self.reciver\_button = Button(text="RECIEVER INFO" , background\_color =(0,1,1,1),color =(1, 1, 1, 1)) | |  | self.reciver\_button.bind(on\_press=self.on\_click\_reciever) | |  | self.add\_widget(self.reciver\_button) | |  | self.add\_widget(self.donor\_button) | |  | return | |  |  | |  | def bind\_start\_tab(self,instance): | |  | self.start\_tab() | |  | return | |  | #reciever function start | |  | def on\_click\_reciever(self, instance): | |  | self.clear\_widgets() | |  | self.cols=2 | |  | self.spacing = 4 | |  | #Creating label dynamically | |  | for i in range(0,active\_list.counter): | |  | arr=active\_list.Reciever\_func() | |  | self.bu1 = Label(text=str(i+1),font\_size="18sp") | |  | self.add\_widget(self.bu1) | |  | self.add\_widget(Label(text=F'{arr[0]}{arr[1]}{arr[2]}{arr[3]}')) | |  | self.bu2=Button(text="Enter Serial Number") | |  | self.bu2.bind(on\_press=self.on\_click) | |  | self.add\_widget(self.bu2) | |  | self.index=TextInput(multiline=False) | |  | self.add\_widget(self.index) | |  | self.exit=Button(text="Exit") | |  | self.exit.bind(on\_press=self.bind\_start\_tab) | |  | self.add\_widget(self.exit) | |  | return | |  |  | |  | def on\_click(self, instance): | |  | self.clear\_widgets() | |  | self.cols=1 | |  | self.add\_widget(Label(text="Have you Recieved the Blood?",font\_size="15sp")) | |  | self.butt2 = Button(text="Yes") | |  | self.butt3 = Button(text="No") | |  | self.add\_widget(self.butt2) | |  | self.add\_widget(self.butt3) | |  | self.butt2.bind(on\_press=self.on\_click\_yes) | |  | self.butt3.bind(on\_press=self.on\_click\_reciever) | |  | return | |  |  | |  | def on\_click\_yes(self, instance): | |  | self.clear\_widgets() | |  | self.cols=1 | |  | #moving information from active to inactive linked list | |  | arr=active\_list.delete\_send\_informtion(int(self.index.text)) | |  | inactive\_list.create\_node(arr[0],arr[1],arr[2],arr[3]) | |  | self.write\_file\_for\_inactive(arr) | |  | #deleting the selected node | |  | active\_list.Delete\_at(int(self.index.text)) | |  | self.del\_from\_active\_file(arr) | |  |  | |  | self.butoon=Button(text="Thank You",font\_size="16sp") | |  | self.butoon.bind(on\_press=self.bind\_start\_tab) | |  | self.add\_widget(self.butoon) | |  | return | |  | #function to del data from active file | |  | def del\_from\_active\_file(self,arr): | |  | FILE=open("Active.txt","w") | |  | for i in range(active\_list.counter): | |  | li=active\_list.backward\_traversing() | |  | for i in li: | |  | FILE.write(i) | |  | return | |  | #function to write data in inactive file | |  | def write\_file\_for\_inactive(self,arr): | |  | FILE=open("Inactive.txt","a") | |  | FILE.write(arr[0]) | |  | FILE.write(arr[1]) | |  | FILE.write(arr[2]) | |  | FILE.write(arr[3]) | |  | FILE.close() | |  | return | |  |  | |  | #donor tab function starts | |  | def on\_click\_donor(self, instance): | |  | # CLERING THE SCREEN | |  | self.clear\_widgets() | |  | self.spacing = 4 | |  | # COLUMN = 2 | |  | self.cols = 2 | |  |  | |  | # button to submit data | |  | self.button1 = Button(text="Submit", background\_color=(0, 1, 1, 1),color=(1, 1, 1, 1)) | |  | self.button1.bind(on\_press=self.on\_submit) | |  | self.button2 = Button(text="Exit", background\_color=(0, 1, 1, 1), color=(1, 1, 1, 1)) | |  | self.button2.bind(on\_press=self.bind\_start\_tab) | |  |  | |  | self.text0 = Label(text="Donor Name") | |  | self.name = TextInput(multiline=False) | |  |  | |  | self.text1 = Label(text="Donor Contact Number") | |  | self.phone = TextInput(multiline=False) | |  |  | |  | self.text2 = Label(text="Donor Blood\_Type") | |  | self.blood = TextInput(multiline=False) | |  |  | |  | self.text3 = Label(text="City (where is he currently available)") | |  | self.city = TextInput(multiline=False) | |  |  | |  | # adding widgets | |  | self.add\_widget(self.text0) | |  | self.add\_widget(self.name) | |  | # phone number | |  | self.add\_widget(self.text1) | |  | self.add\_widget(self.phone) | |  | # blood type | |  | self.add\_widget(self.text2) | |  | self.add\_widget(self.blood) | |  | # city | |  | self.add\_widget(self.text3) | |  | self.add\_widget(self.city) | |  |  | |  | self.add\_widget(self.button1) | |  | self.add\_widget(self.button2) | |  | return | |  |  | |  | def on\_submit(self, instance): | |  | #opeing active file and appending data in it | |  | FILE = open("Active.txt", "a") | |  | FILE.write("\n") | |  | FILE.write(self.name.text) | |  | FILE.write("\n") | |  | FILE.write(self.phone.text) | |  | FILE.write("\n") | |  | FILE.write(self.blood.text) | |  | FILE.write("\n") | |  | FILE.write(self.city.text) | |  | FILE.close() | |  | #creating data to be stored in active list Adding a new line after eachh variable | |  | self.name.text=self.name.text+"\n" | |  | self.phone.text=self.phone.text+"\n" | |  | self.blood.text=self.blood.text+"\n" | |  | self.city.text=self.city.text+"\n" | |  | active\_list.create\_node(self.name.text, self.phone.text,self.blood.text,self.city.text) | |  | self.start\_tab() | |  | return | |  | #donor tab function ends | |  |  | |  | #class for creating the app | |  | class Blood\_DonationApp(App): | |  | def build(self): | |  | return grid\_layout\_used() | |  |  | |  |  | |  |  | |  | # data struture | |  | class Node: | |  | #defing data to be stored | |  | def \_\_init\_\_(self, name=None, phone\_num=None, blood\_type=None, city=None, next=None, pre=None): | |  | self.name = name | |  | self.phone\_number = phone\_num | |  | self.blood\_type = blood\_type | |  | self.city = city | |  | self.next = next | |  | self.pre = pre | |  | return | |  | #data collection class | |  | class data\_collection\_Dlinkedlist: | |  | def \_\_init\_\_(self): | |  | self.head = None | |  | self.last = self.head | |  | self.reciever\_data = None | |  | self.file\_data=None | |  | self.counter=0 | |  | return | |  |  | |  | # creating a node for storing data of the donor (we are creating node at the start) | |  | def create\_node(self, name, phone\_num, blood\_type, city): | |  | node = Node(name, phone\_num, blood\_type, city) | |  | if self.head == None: | |  | self.head = node | |  | self.last=self.head | |  | self.file\_data=self.last | |  | else: | |  | node.next = self.head | |  | self.head.pre = node | |  | self.head = node | |  | self.counter += 1 | |  | self.reciever\_data = self.head | |  | return | |  |  | |  | # traversing | |  | def forward\_traversing(self,turns): | |  | temp = self.head | |  | for i in range(0, turns): | |  | temp = temp.next | |  | print("Phone Number of donor: ", temp.name) | |  | print("City Type of donor: ", temp.city) | |  | print("Phone Number of donor: ", temp.phone\_number) | |  | print("Blood Type of donor: ", temp.blood\_type) | |  | return | |  |  | |  | #USING THIS FUNCTION TO STORE VALUES in Inactive FILE | |  | def backward\_traversing(self): | |  | if self.file\_data==None: | |  | self.file\_data=self.last | |  | li=[] | |  | li.append(self.file\_data.name) | |  | li.append(self.file\_data.phone\_number) | |  | li.append(self.file\_data.blood\_type) | |  | li.append(self.file\_data.city) | |  | self.file\_data=self.file\_data.pre | |  | return li | |  |  | |  | # DISPLAYING ALL NODES THATS HAVE BEEN CREATED | |  | def Display\_all(self): | |  | temp = self.head | |  | while temp != None: | |  | print("Name of donor: ", temp.name) | |  | print("Phone Number of donor: ", temp.phone\_number) | |  | print("Blood Type of donor: ", temp.blood\_type) | |  | print("City of donor: ", temp.city) | |  | temp = temp.next | |  | return | |  |  | |  | #this function is use to take information from the node and return it in list form | |  | def Reciever\_func(self): | |  | if(self.reciever\_data==None): | |  | self.reciever\_data=self.head | |  | li = [] | |  | li.append(self.reciever\_data.name) | |  | li.append(self.reciever\_data.phone\_number) | |  | li.append(self.reciever\_data.blood\_type) | |  | li.append(self.reciever\_data.city) | |  | self.reciever\_data = self.reciever\_data.next | |  | return li | |  |  | |  | #delete any node from any location | |  | def Delete\_at(self, pos): | |  | if self.head == None: | |  | print("Node is empty") | |  | elif pos == 1: #deleting from start | |  | self.head = self.head.next | |  | if self.counter==1: | |  | pass | |  | else: | |  | self.head.pre = None | |  | elif pos==self.counter: #deleting from end | |  | temp = self.head | |  | while temp.next != None: | |  | temp = temp.next | |  | temp = temp.pre | |  | temp.next = None | |  | self.last=temp | |  | self.file\_data=self.last | |  | else: #deleting from any point | |  | temp = self.head | |  | i = 1 | |  | while i < pos: | |  | print(i) | |  | temp = temp.next | |  | i += 1 | |  | temp.pre.next = temp.next | |  | temp.next.pre = temp.pre | |  | temp.next = None | |  | temp.prev = None | |  | self.counter =self.counter-1 | |  | return | |  | #returing the information store in the node that is to be deleted so that we can store | |  | #it in inactive file | |  | def delete\_send\_informtion(self,pos): | |  | temp=self.head | |  | i=1 | |  | while i<pos: | |  | temp=temp.next | |  | i+=1 | |  | li=[] | |  | li.append(temp.name) | |  | li.append(temp.phone\_number) | |  | li.append(temp.blood\_type) | |  | if pos==1: | |  | temp.city=temp.city+"\n" | |  | li.append(temp.city) | |  | return li | |  |  | |  | # CREATING GLOBAL OBJECT OF DATA COLLLECTION CLASS SO THAT WE CAN USE IT IN ANYOTHER CLASS | |  | global active\_list | |  | global inactive\_list | |  | active\_list = data\_collection\_Dlinkedlist() | |  | inactive\_list= data\_collection\_Dlinkedlist() | |  |  | |  | if \_\_name\_\_ == "\_\_main\_\_": | |  | #opening file | |  | FILE = open("Active.txt", "r") | |  | y = 0 | |  | for i in FILE: | |  | y += 1 | |  | if y == 1: | |  | name = i | |  | elif y == 2: | |  | contact\_num = i | |  | elif y == 3: | |  | blood\_type = i | |  | elif y == 4: | |  | city = i | |  | active\_list.create\_node(name, contact\_num, blood\_type, city) | |  | y = 0 | |  | FILE.close() | |  | #closing file | |  | Blood\_DonationApp().run() | |

* CONCLUSION:

We learned how to design an app. We used the concept of linked list Which, is the important part of data structures and algorithm.