**Assignment 5**

**Information structures with python**

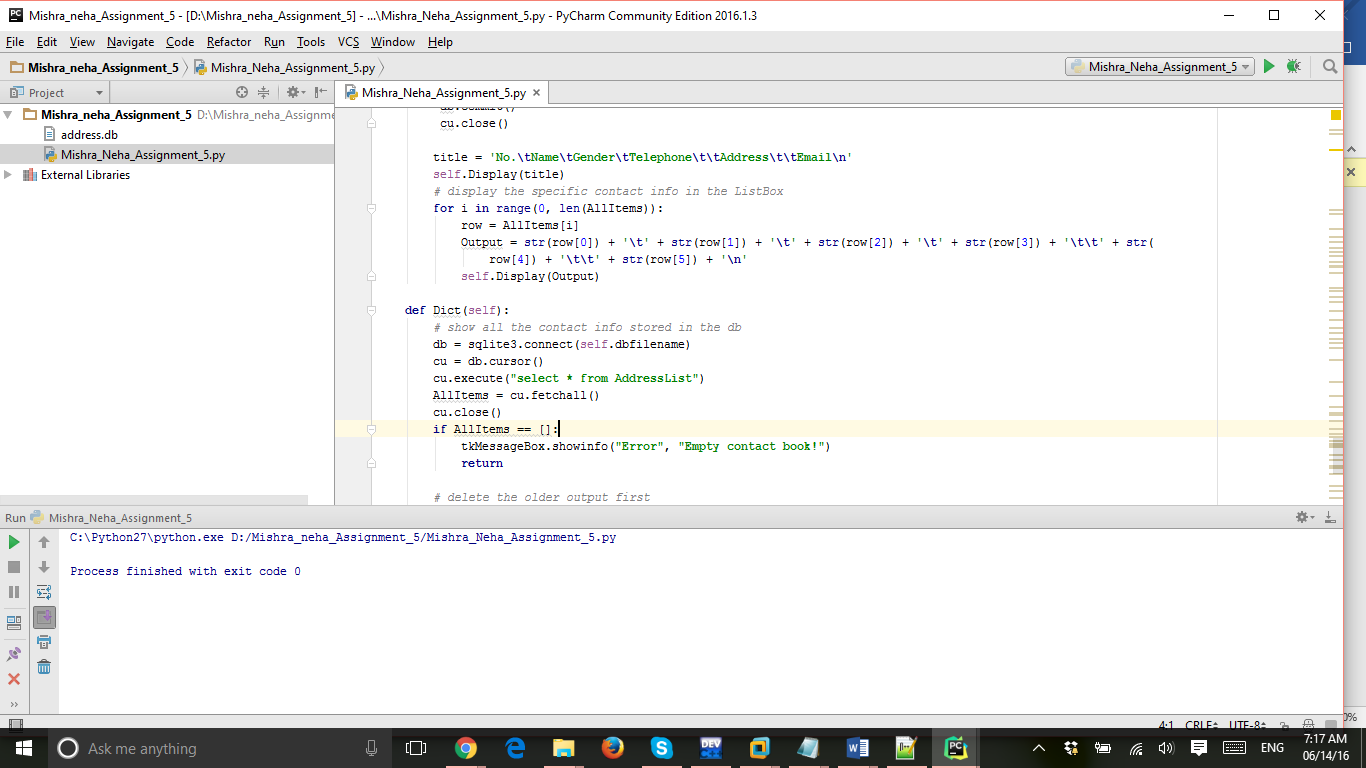
**Deliverables**

**Part 1—Overall Project Description:**

In this project I have created contact list application .

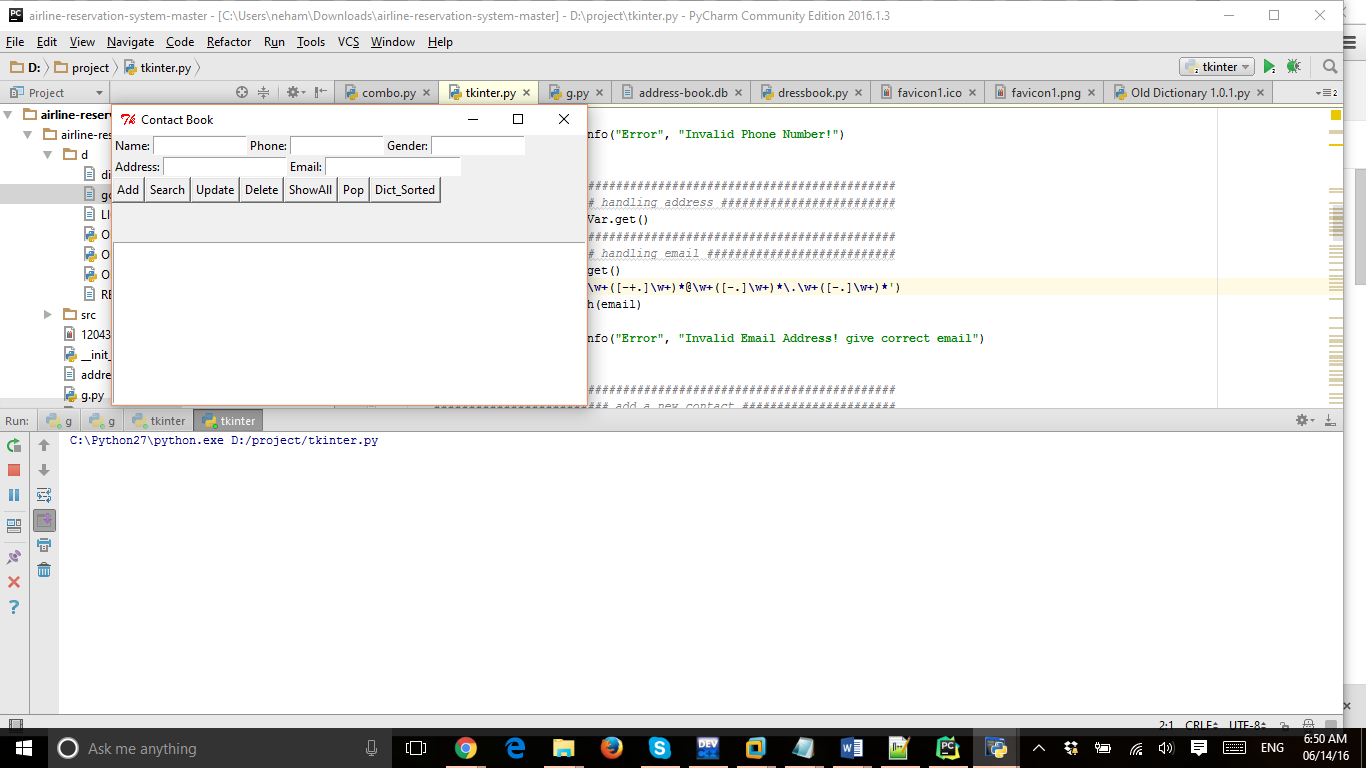
In this application I have implemented the fucntions like add,update,delete,search,pop,dict\_sort

**Part 2 –Structure of directory is**

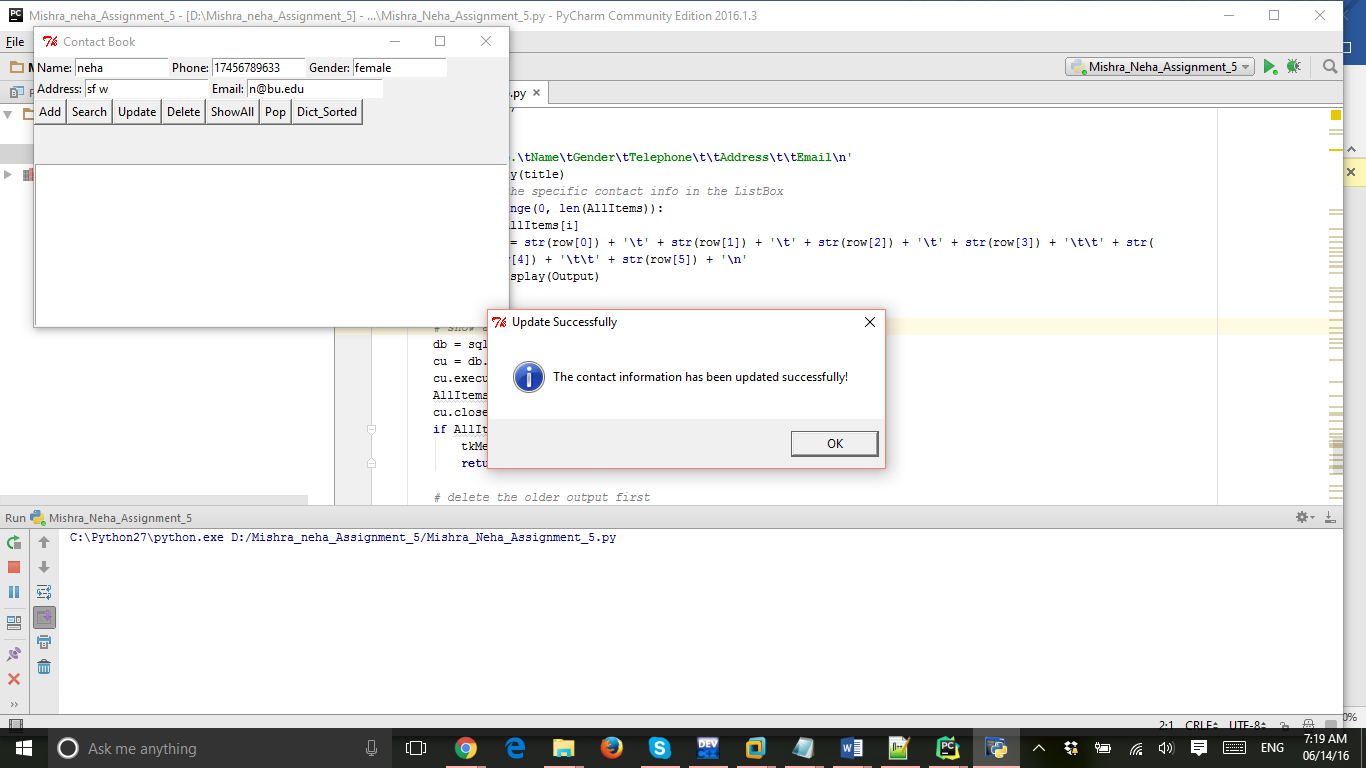


**Part 3 : Applications with test and outputs :**

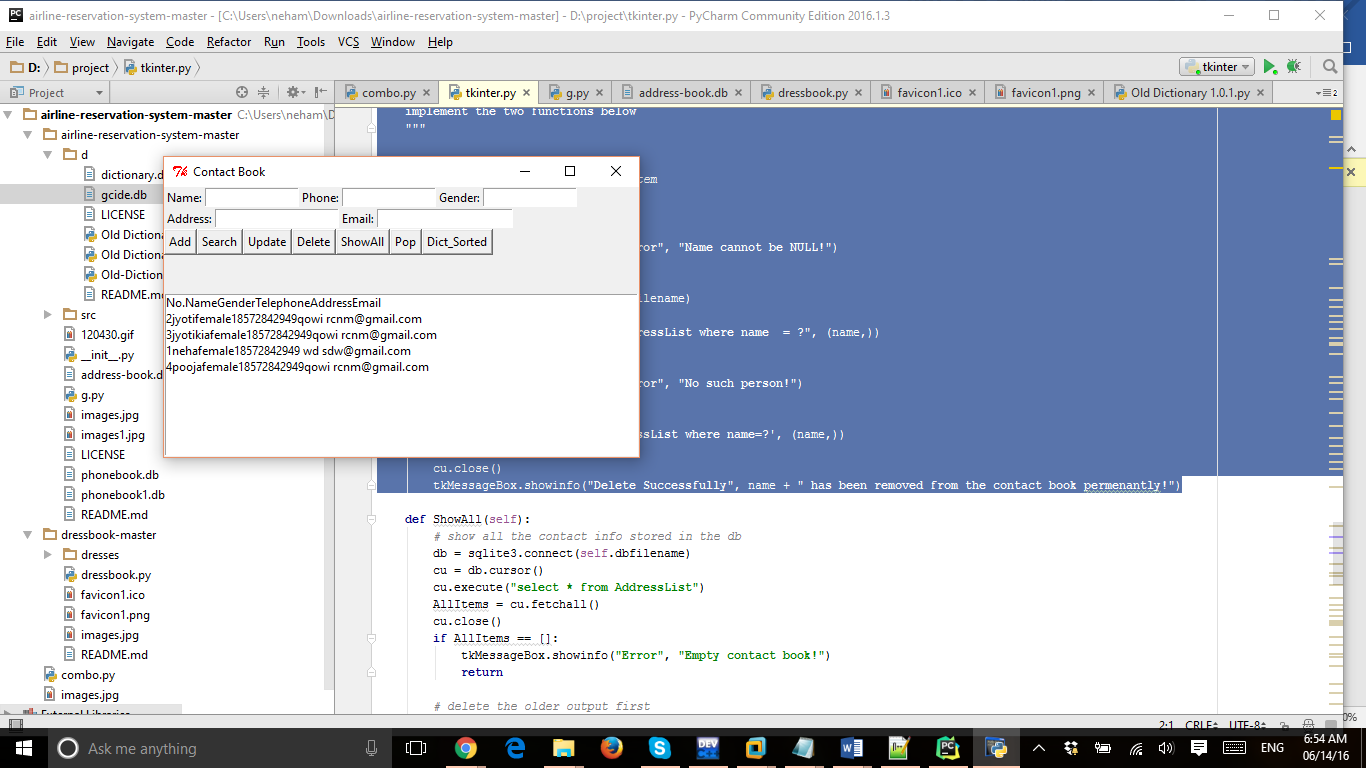
**This is how the main applications looks like**

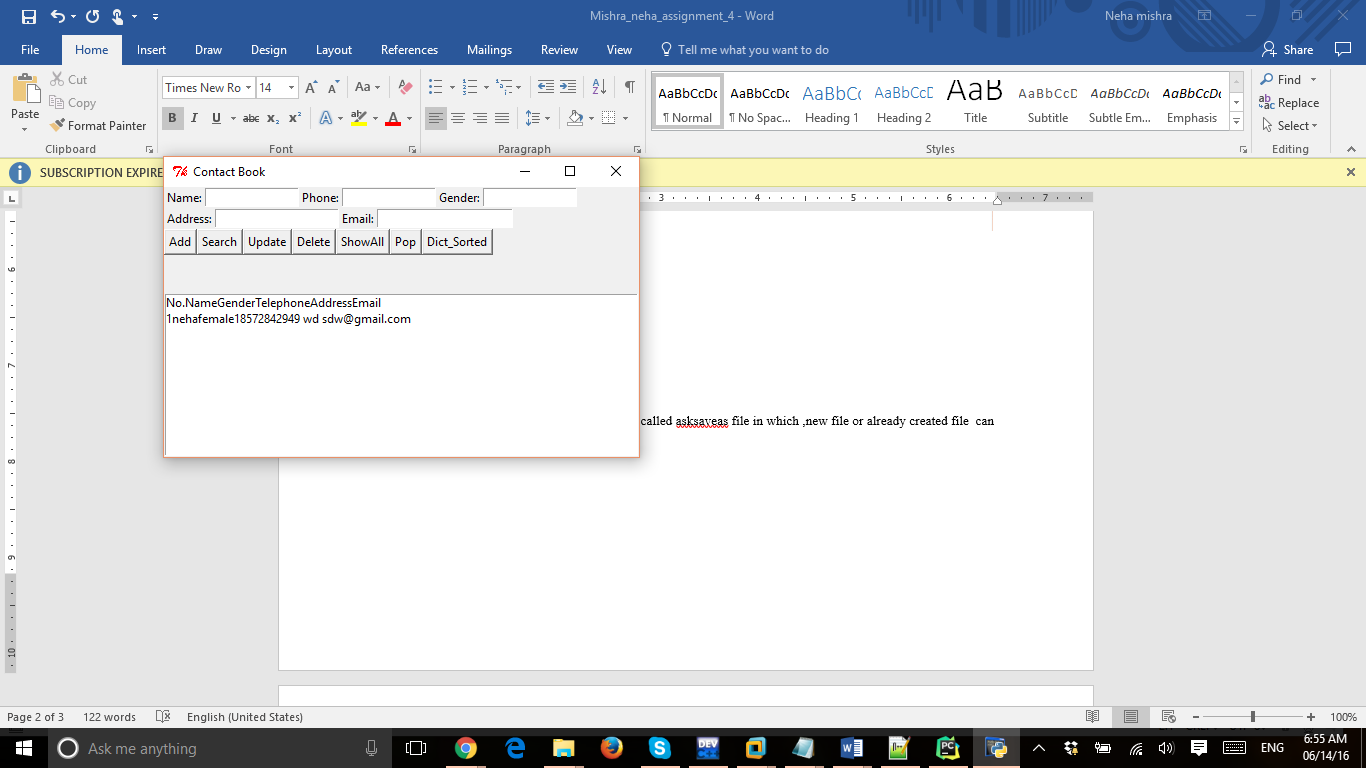


This is update query implemented :



**This is feature of sorted with the help dict\_sort**





**This is feature of poping the first value from the database similarly we can do the dequeue also. but it is not included**

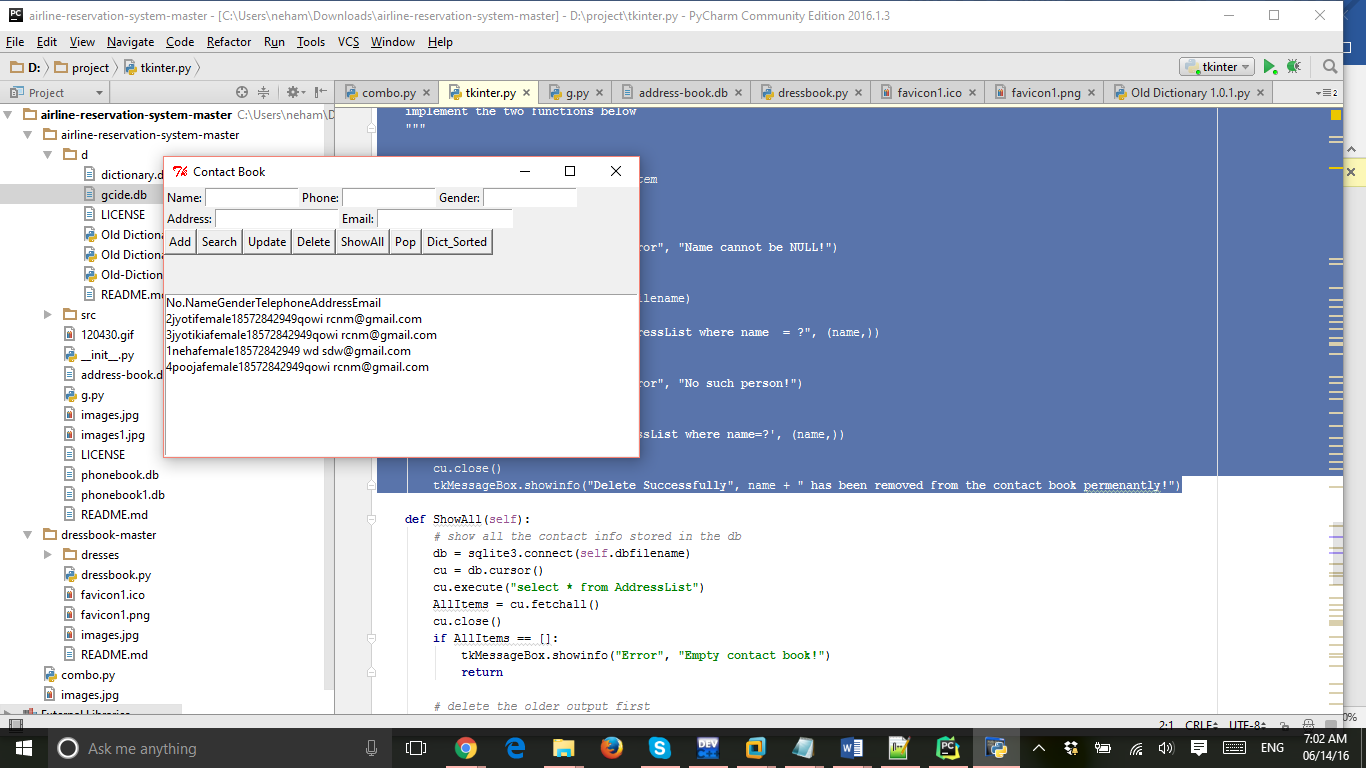
Part 4 :

**List of python elements included are :**

Dictionaries,database (sqlite),stack,set for matching value.

One example is shown below

This is snapshot of dict\_sort:



Part 5 : Example code

The code which is best is insert code , reasons:

1. If I cannot insert the code there will no list to show
2. Every name is insert is unique so it is easy to sort .
3. I have applied validation also like email ,phone number validation.

**def** Add(self):  
 db = sqlite3.connect(self.dbfilename)  
 cu = db.cursor()  
 *###################### handling name ############################* name = self.nameVar.get()  
 **if** name == **""**:  
 tkMessageBox.showinfo(**"Error"**, **"Name can't be NULL!"**)  
 cu.close()  
  
 cu.execute(**'select name from AddressList'**) *# testify name* **for** namelist **in** cu.fetchall():  
 **if** name **in** namelist:  
 tkMessageBox.showinfo(**"Error"**, **"Name already exists!"**)  
 cu.close()  
 **return** *##################################################################  
 ###################### handling gender ###########################* sex = self.sexVar.get()  
 **if** sex != **'male' and** sex != **'female'**:  
 tkMessageBox.showinfo(**"Error"**, **"Invalid Gender! type either male or female!"**)  
 cu.close()  
 **return** *##################################################################  
 ################## handling telephone number with validation#####################* phone = self.phoneVar.get()  
 pattern = re.compile(**'^[1][3-8]+\\d{9}'**)  
 correct = pattern.match(phone)  
 **if** correct == None:  
 tkMessageBox.showinfo(**"Error"**, **"Invalid Phone Number! Enter 1 then 10 digits number"**)  
 cu.close()  
 **return** *##################################################################  
 ####################### handling address #########################* address = self.addressVar.get()  
 *##################################################################  
 ####################### handling email with validation ###########################* email = self.emailVar.get()  
 pattern = re.compile(**'\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*'**)  
 correct = pattern.match(email)  
 **if** correct == None:  
 tkMessageBox.showinfo(**"Error"**, **"Invalid Email Address! give correct email"**)  
 cu.close()  
 **return** *##################################################################  
 ######################### add a new contact ######################* values = (name, sex, phone, address, email)  
 cu.execute(**'insert into AddressList(name,sex,phone,address,email) \  
values(?, ?, ?, ?, ?)'**, values)  
 db.commit()  
 cu.close()  
 tkMessageBox.showinfo(**"New Contact"**, **"A new Contact has been successfully added!"**)

All the code is attached in the zip file which has database file also for convience.