

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

****

Name of candidates:

**1**. Spoorthi S (01JST17IS062)

**2**. Pooja J N (01JST16IS027)

**3**. Aishwarya Duth G (01JST18IS400)

**4.** Divyashree R (01JST18IS402)

**Abstract**

This project unicard helps in Document linking and Management of the card. The project has three views admin who can insert, user can fetch the data related to and officer who can insert some relevant data. The project mainly aims in easily fetching the data.

**Contents**

1. Introduction

* Aim/Objective
* Features

2. ER diagram

3. Schema diagram

* Introduction to 1NF 2NF 3NF and BCNF

5. Frontend

6. Backend

7. Conclusion

**Introduction**

An Aadhaar card is a unique number issued to every citizen in India and is a centralised and universal identification number. The [Aadhar card](https://www.bankbazaar.com/aadhar-card.html) is a biometric document that stores an individual's personal details in a government database, and is fast becoming the government's base for public welfare and citizen services.

While the various uses of Aadhar card are publicised by the government through awareness campaigns, there are a few uses that many users would not be aware of. We have highlighted those uses.

For example, an Aadhar card can be used as proof of identity, proof of address as well as proof of age when applying for any government service.

In this way, it is a very versatile card, as it can be used for all government related services and programm

**Things to know about Aadhaar card :**

* One can enrol for Aadhaar at any age including infants. Once attaining five years of age, the biometrics details - through iris and fingertips scans - of the children can be updated on Aadhaar. "When your child reaches the age of 5, it is important to get his/her biometrics updated to Aadhaar. Not doing so will make the Bal Aadhaar invalid," the UIDAI said.
* Aadhaar is "equally important for the elderly".  
    
   The Aadhaar enrolment system - the issuer of Unique Identity Number and Aadhaar card - also provisions for enrolment of individuals with "ill-defined fingerprints or missing biometrics".  
  Also, persons suffering from eye ailments -cataract, retina damage or partial/complete blindness - can also enrol for Aadhaar card.  
  Aadhaar enrolment can be done even if someone in a family doesn't have individual valid docs," the Authority said in another Tweet.

Who can get Aadhaar card

* Aadhaar card is available to residents only. Therefore, NRIs (non-resident Indians) and OCIs (Overseas Citizens of India) are not eligible for Aadhaar card.
* The enables Aadhaar card holders to download a digital copy of the biometrics-based identity card. "Please ensure that you delete the local copy of your Aadhaar downloaded on any public machine to avoid its misuse," the UIDAI advised Aadhaar holders.
* Aadhaar is to be mandatorily quoted for a number of important tasks. These range from investment in PPF (Public Provident Fund), NSC (National Savings Certificate) to bank account to applying for a PAN (Permanent Account Number) and filing income tax return (ITR). The UIDAI has advised Aadhaar card holders not to share their Aadhaar number or Aadhaar OTP with anyone.

**1.1 Aim/Objective**

Aadhaar program was launched with a main objective to provide universal identity to every Indian resident.As its acceptance as a mandatory document for various initiatives has been officially made. In addition to this it will help in reducing the corruption since every individual carries only one unique number. Believed linking documents in further be given more importance. As more and more government schemes are being launched by using data linked to Aadhaar.

**1.2 Features**

**Pan card**

The problem of individuals having multiple PAN cards still exists. However, since Aadhaar is linked to your unique biometric identification (unique fingerprint), the problem of multiple PAN cards will be automatically resolved. Hence, the government will be able to track all the transactions, asset purchases and expenses incurred by an individual through this linking. This will largely plug evasion of taxes as most of your key transactions will be tracked through this mechanism. That is expected to reduce the volume of black money in the economy. This linking will help the government to set up a meticulous and foolproof surveillance system, which will largely be a technology-driven. Since demonetisation has already underscored the move from a cash economy to a digital economy, the government can now track volumes of transactions through the effective use of mapping technology.

**Ration card**

A very important official document, ration card entitles you to avail fuel and food grains that are offered by the government at a subsidized rate. The card is very important in India. It also allows people to have a proof of identity in the country and establish a connection with the database of the government. Another important identity proof issued by the Government of India is an Aadhaar card. You also need to link your Aadhaar number with the ration card so as to receive maximum benefits of social welfare schemes and any cases of fraud can be prevented. A very important official document, ration card entitles you to avail fuel and food grains that are offered by the government at a subsidized rate. Aadhaar card will ensure that the right people get the benefit of PDS.  Aadhaar will manage to create an audit trail in the distribution system by identifying and eliminating the corrupt middlemen.

**Health card**

Patients Aadhaar numbers will be linked to a health ID and these will be used in electronic health records.The health records will contain all the information related to the patient including health. records produced during his or her visit to the hospital such as X-ray reports, blood test reports. This will allow continuity of care when patients go to a facility and the records can be shared electronically. This will avoid duplication.

**Travel ID**

As per the current policy of railway travel concession, male senior citizen of the age of 60 years and above, women senior citizen of the age of 58 and above are granted 40% and 50% concession . There are many such government schemes so, these schemes can be easily implemented if Aadhaar is linked with Travel IDs it becoms easy and essential for implementing such benefits. And can avoid corrupt middlemen.

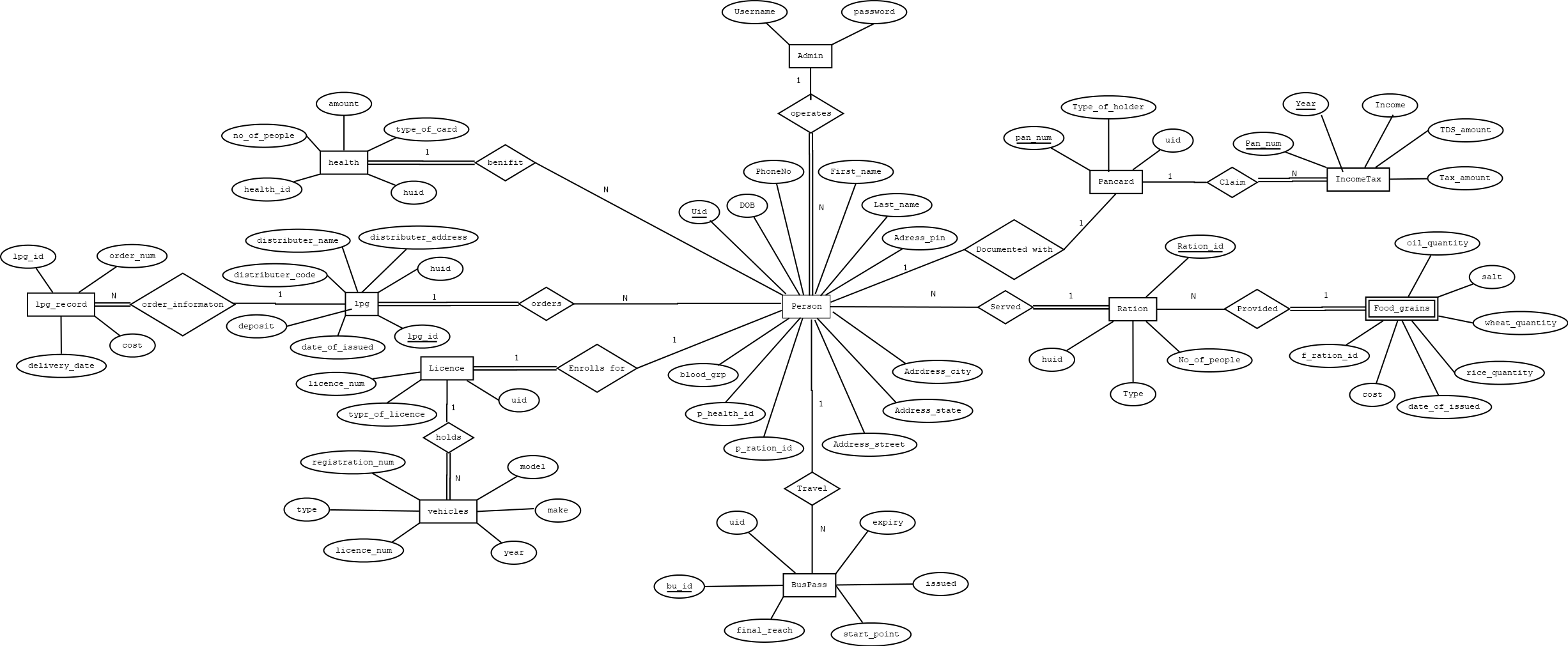
**LPG Connection**

He Indian government is providing subsidy on domestic liquefied petroleum gas (LPG) for all the households across India. To receive the LPG subsidy one has to link their Aadhaar card to the LPG gas connection as the subsidy amount back to people. provide a photocopy of Aadhar card to the respective gas agency dealer to link it to the consumer number or LPG gas connection. The prices of LPG in India is decided based on crude prices in the international market added with the currency exchange rate. With an intention to help the common man to afford the LPG cylinders, the central government is providing subsidy on domestic cylinders up to 12 cylinders per year. The subsidy amount will be directly credited to the customer's .Based on income of the people so, that corruption reduces in providing subsidy.

**Lisence**

The Government has come across many instances where a single person has more than one drivinglicense. As per the acts related to road transport, a person cannot possess multiple drivers’ licenses. Driving license issued in one state is valid in all other states and union territories of the country. Thus, a person need not have multiple driving licenses.To curb this menace, the central government is planning to link Aadhaar with driving licenses.

**Entity Representation Diagram(ER diagram)**

****

**Schema**

Health Card

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Health id** | Huid | No of people | Amount | Type of card |

LPG

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lpg id** | Date of issue | Deposit | Distributer code | Distributer name | Distributer address | Huid |

LPG Record

|  |  |  |  |
| --- | --- | --- | --- |
| Cost | Delivery date | **Order number** | Lpg Id |

License

|  |  |  |
| --- | --- | --- |
| Type of license | UID | **License number** |

Food grains

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **F Ration id** | Date of issue | Cost | Rice Quantity | Wheat Quantity | Salt | Oil Quantity |

Income tax

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pan Num** | Year | Income | TDS Amount | Tax amount |

 Pan card

|  |  |  |
| --- | --- | --- |
| **Pan Num** | Type of holder | UID |

Person

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FName | LName | DoB | **UID** | Address | Phone | Blood group | Ration ID | Health ID |

Ration card

|  |  |  |  |
| --- | --- | --- | --- |
| **Ration\_id** | Huid | Type | No\_of\_people |

Transport

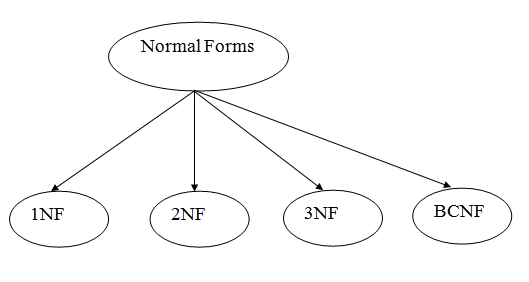
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bu id** | UID | Final reach | Start point | Issued | Expiry |

Vehicle

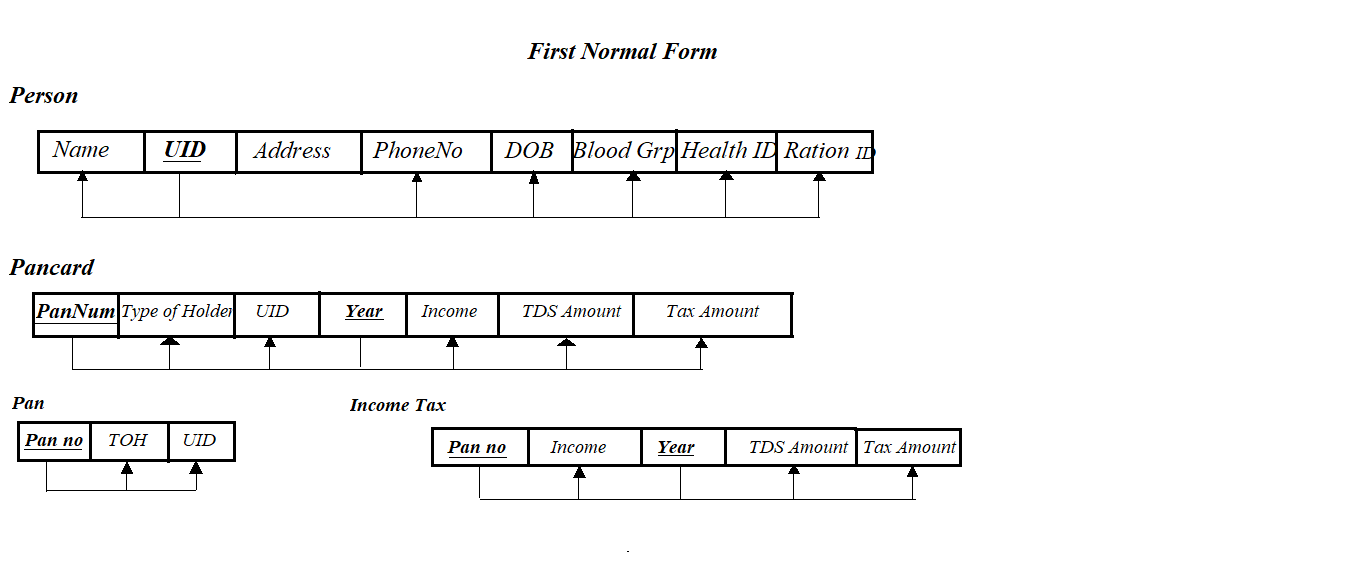
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Registration number** | Licence Number | Type | Year | make | Model |

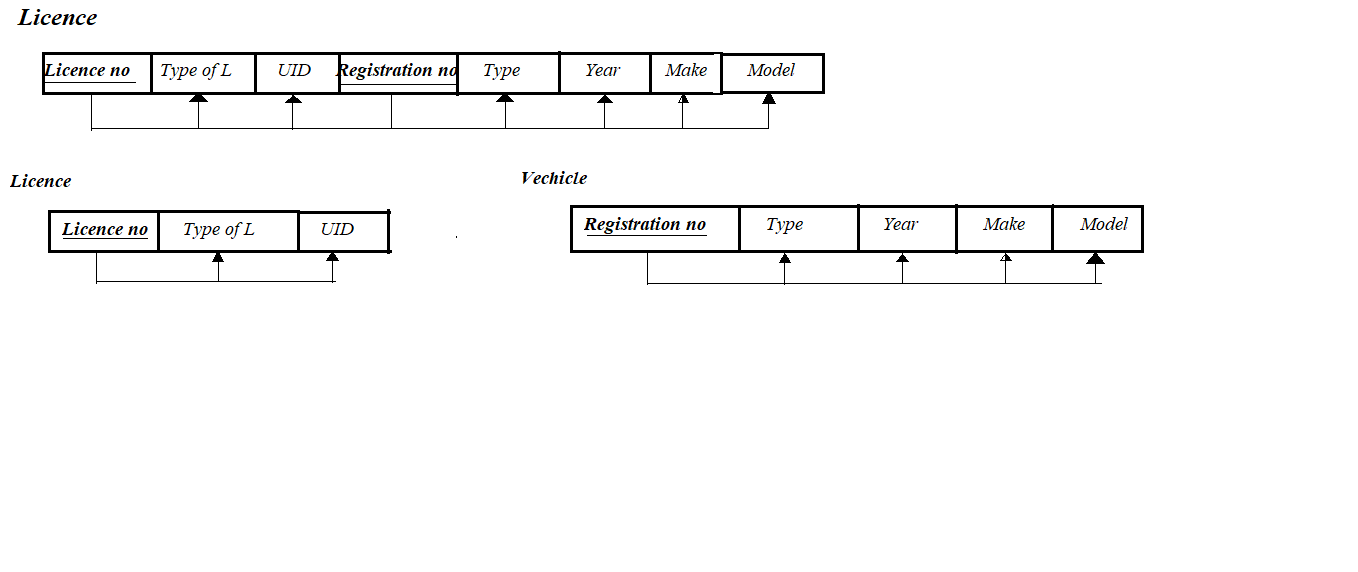
**Normalization**

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
* Normalization divides the larger table into the smaller table and links them using relationship.
* The normal form is used to reduce redundancy from the database table.



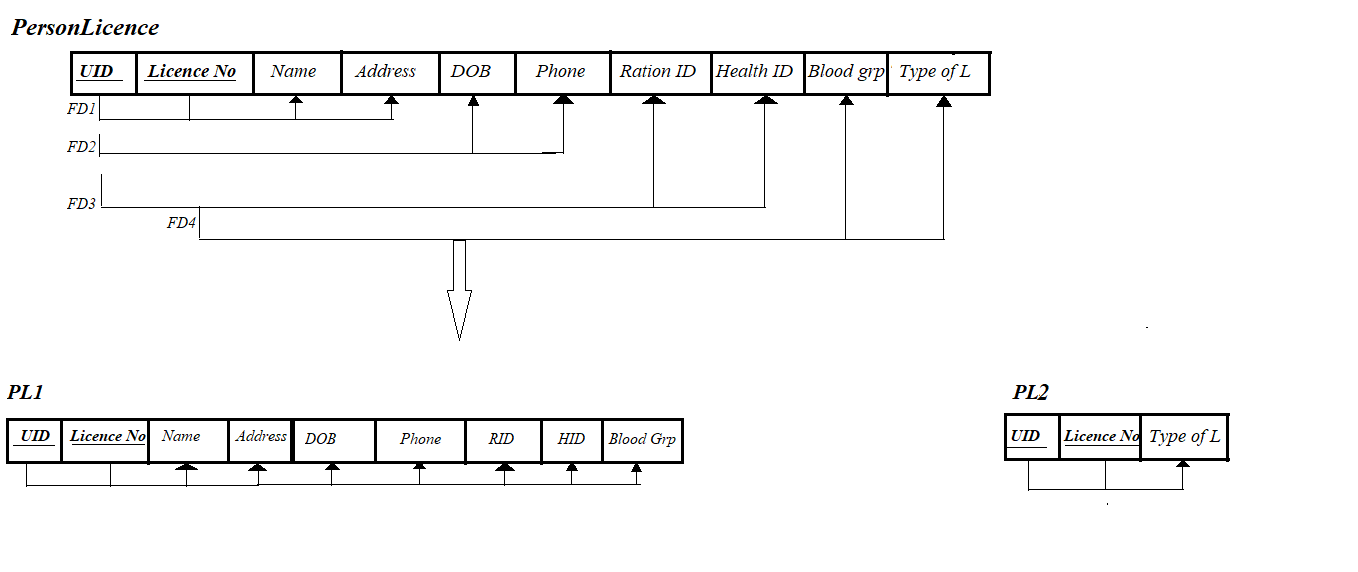
**First Normal Form:**

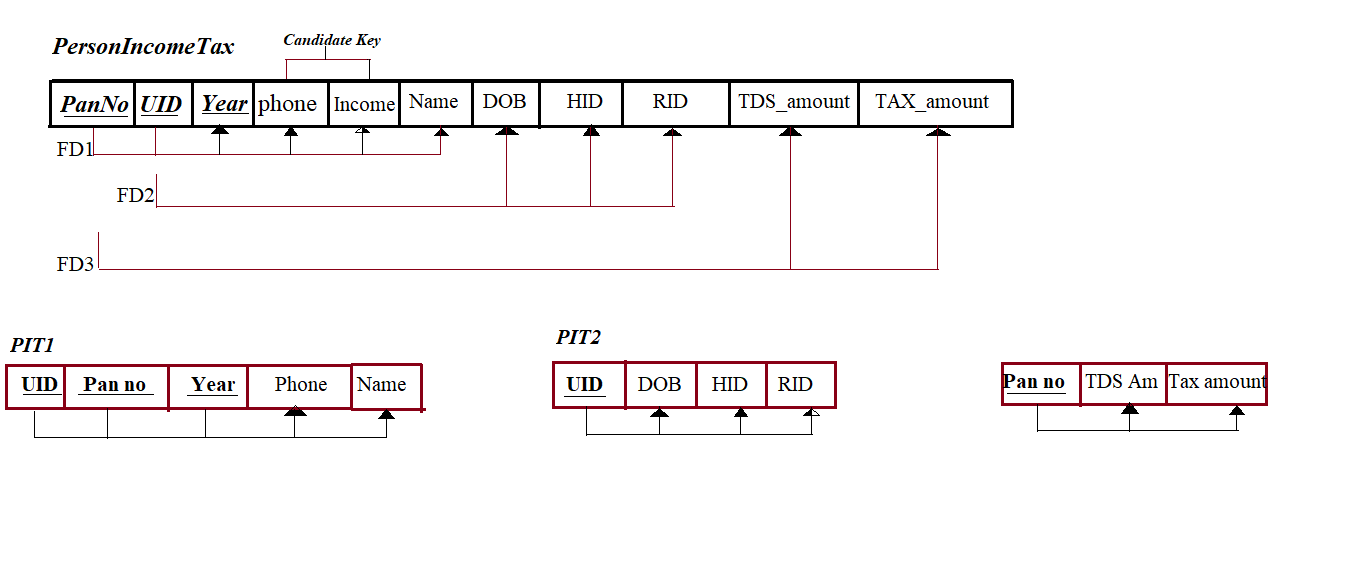
It States that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be single value from the domain of that attribute. 



Second Normal Form :

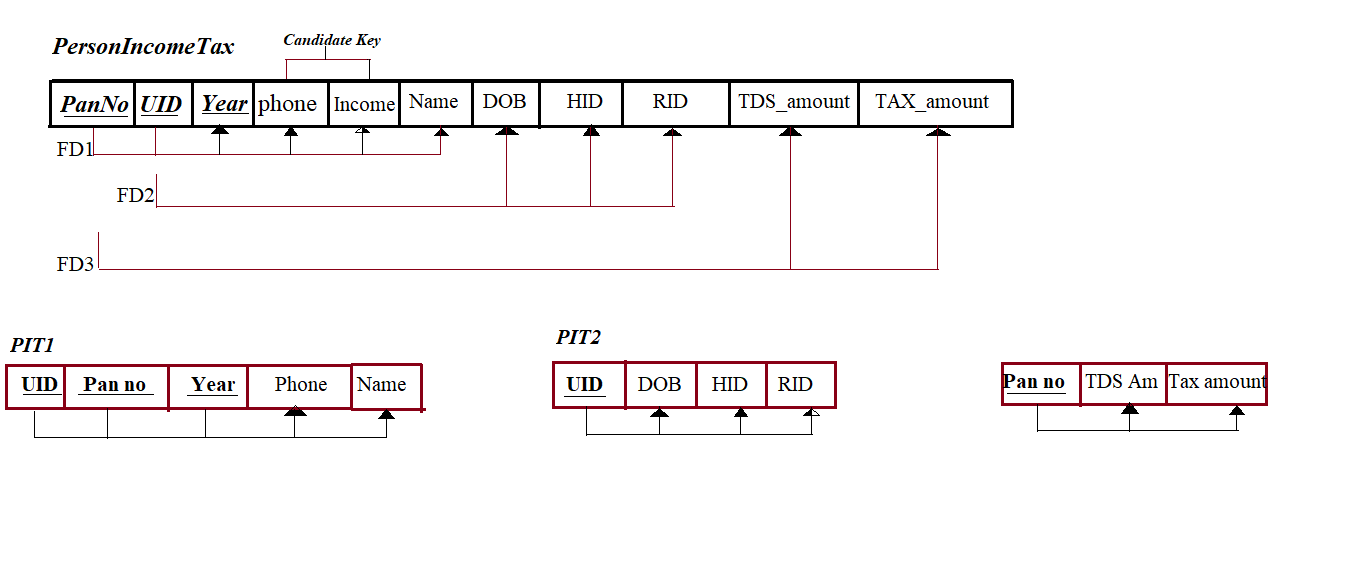
A Table is said to be in 2NF if both the following conditions hold

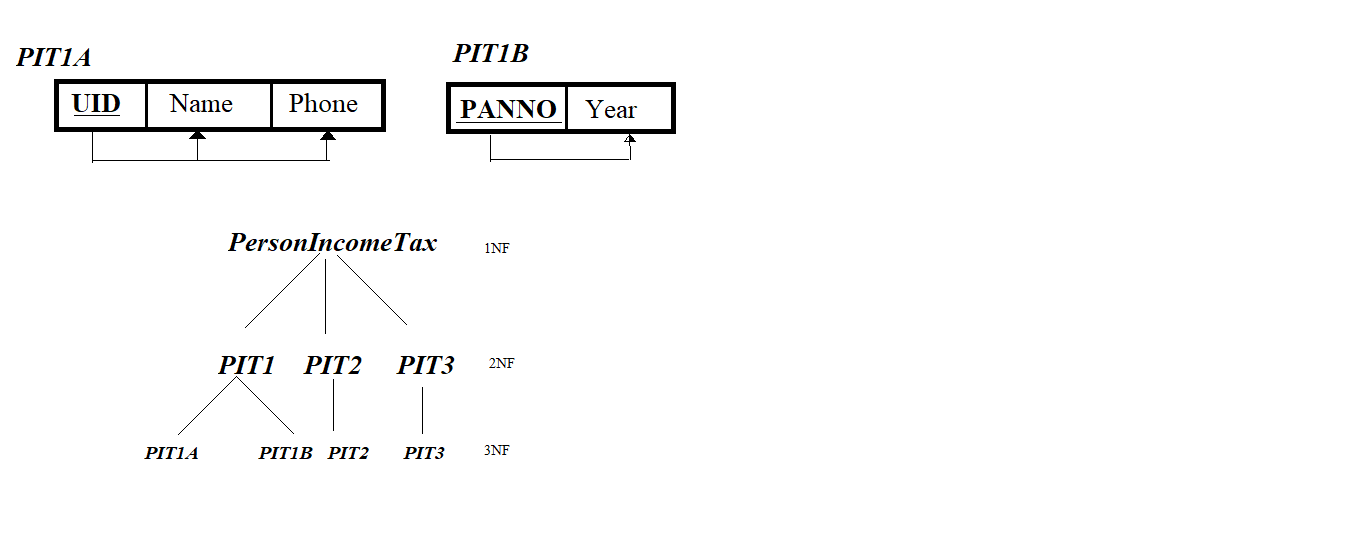
* Table is in 1NF(First Normal Form)
* No non-prime attribute is dependent on the proper subset of any candidate key of table.



Third Normal Form:

* + It should be in the 2NF(Second Normal Form)
  + Transitive functional dependency of non-prime attribute on any supper key should be removed.





BCNF

* The table should be in 3NF(Third Normal Form)
* The functional Dependency of X🡪Y where X is a supper key.

Table in BCNF

|  |  |  |
| --- | --- | --- |
| **UID** | Ration ID | Health ID |
| 546789 | 90807 | 56789 |
| 546789 | 90807 | 56789 |
| 234567 | 25874 | 23109 |
| 234567 | 25874 | 23109 |
| 987654 | 23689 | 34567 |
| 987654 | 23689 | 34567 |
| 123987 | 54216 | 45678 |
| 123987 | 54216 | 45678 |

|  |  |
| --- | --- |
| **UID** | Health ID |
| 546789 | 56789 |
| 546789 | 56789 |
| 234567 | 23109 |
| 234567 | 23109 |
| 987654 | 34567 |
| 987654 | 34567 |
| 123987 | 45678 |
| 123987 | 45678 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ration ID** | Huid | Type | No\_of\_people |
| 90807 | 08976566 | APL | 3 |
| 90807 | 08976566 | BPL | 7 |
| 25874 | 08765432 | APL | 4 |
| 25874 | 09213456 | BPL | 5 |

**code:**

*import mysql.connector as mc*

*from tkinter import \**

*import tkinter as tk*

*import sys*

*class projectcode:*

*def connectt(self):*

*mydb=mc.connect(host="localhost",user="root",passwd="",database="unicard")*

*mycursor=mydb.cursor()*

*return mydb,mycursor*

*def close(self,mydb,mycursor):*

*mydb.close()*

*mycursor.close()*

*def food\_grains\_retrive(self,huid):*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select \* from food\_grains where f\_ration\_id=(select ration\_id from ration where huid=%s)",(huid,))*

*record=list(mycursor.fetchall())*

*return record*

*def licence\_retrive(self,uid):*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select \* from licence where uid=%s",(uid,))*

*record=mycursor.fetchall()*

*mycursor.execute("select \* from vehicle where licence\_num=(select licence\_num from licence where uid=%s)",(uid,))*

*record1=list(mycursor.fetchall())*

*return record,record1*

*def lpg\_record\_retrive(self,huid):*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select \* from lpg\_record where lpg\_id=(select lpg\_id from lpg where huid=%s)",(huid,))*

*record=list(mycursor.fetchall())*

*return record*

*def income\_tax\_retrive(self,uid):*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select \* from income\_tax where pan\_num=(select pan\_num from pan\_card where uid=%s)",(uid,))*

*record=list(mycursor.fetchall() return record*

*1.delete person,2.delete head of a ration card or*

*3.health card or 4.lpg card and make user defined head'''*

*'''ob = projectcode()*

*record=ob.person\_retrive(13)*

*for i in range(len(record[0])):*

*print(record[0][i])'''*

*def quit(self):*

*self.Canvas1.destroy()*

*self.root.quit()*

*self.root.destroy()*

*exit() def start(self):*

*self.root=Tk()*

*self.root.geometry("600x450+352+160")*

*self.root.configure(background="#d9d9d9")*

*self.initial()self.first\_page()*

*self.root.mainloop()*

*def initial(self):*

*self.Canvas1 = tk.Canvas(self.root, width=1200, height=700)*

*#filename=PhotoImage('C:\\Users\\Spoorthi S\\Desktop\\uni.png')*

*#background\_l=Label(self.Canvas1,image=filename)*

*#Label\_0=Label(self.root,text='lkdjlk')*

*#background\_l.pack()*

*self.Canvas1.place(relx=0.0, rely=0.0, relheight=0.984, relwidth=0.988 self.Canvas1.configure(background="#45d8ac")*

*self.Canvas1.configure(borderwidth="2")self.Canvas1.configure(insertbackground="black")*

*self.Canvas1.configure(relief="ridge") self.Canvas1.configure(selectbackground="#c4c4c4")*

*self.Canvas1.configure(selectforeground="black")self.root.title("UNICARD SYSTEM")*

*def first\_page(self): self.Canvas1.destroy()*

*self.initial() self.root.title("MAIN PAGE")*

*def officerr(self):self.Canvas1.destroy()*

*self.initial() self.root.title("OFFICER PAGE")*

*self.Button1 = tk.Button(self.Canvas1)*

*self.Button1.place(relx=0.202, rely=0.339, height=24, width=70)self.Button1.configure(activebackground="#ececec")*

*self.Button1.configure(activeforeground="#000000") def lpg(self):*

*self.root.title("LPG INSERT PAGE")*

*self.Canvas1.destroy()*

*self.initial() self.Entry7 = tk.Entry(self.Canvas1)*

*self.Entry7.place(relx=0.464, rely=0.105,height=20, relwidth=0.272)*

*self.Entry7.configure(background="white")*

*self.Entry7.configure(disabledforeground="#a3a3a3")*

*self.Entry7.configure(font="TkFixedFont")*

*self.Entry7.configure(foreground="#000000")*

*self.Entry7.configure(insertbackground="black")*

*self.uid=StringVar()*

*self.Entry7.configure(textvariable=self.uid)*

*def lpg\_record\_insert(self):*

*self.uidd=self.uid.get()self.n1=self.ct.get()self.n2=self.dd.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into lpg\_record values(%s,%s,default,(select lpg\_id from lpg where huid=%s))",(self.n1,self.n2,self.uidd))*

*mydb.commit() if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor)*

*def income(self):*

*self.Canvas1.destroy()*

*self.initial()self.root.title("INCOMETAX PAGE")*

*self.Entry7 = tk.Entry(self.Canvas1)*

*self.Entry7.place(relx=0.464, rely=0.105,height=20, relwidth=0.272)*

*self.Entry7.configure(background="white")*

*self.Entry7.configure(disabledforeground="#a3a3a3")*

*self.Entry7.configure(font="TkFixedFont")*

*self.Entry7.configure(foreground="#000000")*

*self.Entry7.configure(insertbackground="black")*

*self.uid=StringVar()*

*self.Entry7.configure(textvariable=self.uid)*

*self.Label7 = tk.Label(self.Canvas1)*

*self.Label7.place(relx=0.182, rely=0.105, height=21, width=100)*

*self.Label7.configure(background="#45d8ac")*

*self.Label7.configure(disabledforeground="#a3a3a3")*

*self.Label7.configure(foreground="#000000")*

*self.Label7.configure(text='''Uid:''')*

*self.Label2 = tk.Label(self.Canvas1)*

*self.Label2.place(relx=0.149, rely=0.226, height=21, width=70)*

*self.Label2.configure(background="#45d8ac")*

*self.Label2.configure(disabledforeground="#a3a3a3")*

*self.Label2.configure(foreground="#000000")*

*self.Label2.configure(text='''TDS:''')*

*self.uidd=self.uid.get() self.n1=self.y.get() self.n2=self.inc.get() self.n3=self.tds.get()self.n4=self.ta.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into income\_tax values((select pan\_num from pan\_card where uid=%s),%s,%s,%s,%s)",(self.uidd,self.y,self.inc,self.tds,self.ta))*

*if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else:self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor) def vehicle(self):*

*self.Canvas1.destroy()*

*self.initial() self.root.title("VEHICAL INSERT PAGE")*

*self.Entry7 = tk.Entry(self.Canvas1)*

*self.Entry7.place(relx=0.464, rely=0.105,height=20, relwidth=0.272)*

*self.Entry7.configure(background="white")*

*self.Entry7.configure(disabledforeground="#a3a3a3")*

*self.Entry7.configure(font="TkFixedFont")*

*self.Entry7.configure(foreground="#000000")*

*self.Entry7.configure(insertbackground="black")*

*self.uid=StringVar()*

*self.Entry7.configure(textvariable=self.uid)*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.149, rely=0.01, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following information:''')*

*self.Message1.configure(width=1000)*

*self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.415, rely=0.158,height=20, relwidth=0.388)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.rn=StringVar()*

*self.Entry1.configure(textvariable=self.rn) self.Entry2 = tk.Entry(self.Canvas1)*

*self.Entry2.place(relx=0.415, rely=0.226,height=20, relwidth=0.388)*

*self.Entry2.configure(background="white")*

*self.Entry2.configure(disabledforeground="#a3a3a3")*

*self.Entry2.configure(font="TkFixedFont")*

*self.Entry2.configure(foreground="#000000")*

*self.Entry2.configure(insertbackground="black")*

*self.ty=StringVar()self.Entry2.configure(textvariable=self.ty)*

*self.Entry3 = tk.Entry(self.Canvas1)*

*self.Entry3.place(relx=0.415, rely=0.293,height=20, relwidth=0.388)*

*self.Entry3.configure(background="white")*

*self.Entry3.configure(disabledforeground="#a3a3a3")*

*self.Entry3.configure(font="TkFixedFont")*

*self.Entry3.configure(foreground="#000000")*

*self.Entry3.configure(insertbackground="black")*

*self.y=StringVar() self.Entry3.configure(textvariable=self.y)*

*self.Entry4 = tk.Entry(self.Canvas1)*

*self.Entry4.place(relx=0.415, rely=0.361,height=20, relwidth=0.388)*

*self.Entry4.configure(background="white")*

*self.Entry4.configure(disabledforeground="#a3a3a3")*

*self.Entry4.configure(font="TkFixedFont")*

*self.Entry4.configure(foreground="#000000")*

*self.Entry4.configure(insertbackground="black")*

*self.ma=StringVar()self.Entry4.configure(textvariable=self.ma)*

*self.Entry5 = tk.Entry(self.Canvas1) self.Entry5.place(relx=0.415, rely=0.429,height=20, relwidth=0.388)*

*self.Entry5.configure(background="white")*

*self.Entry5.configure(disabledforeground="#a3a3a3")*

*self.Entry5.configure(font="TkFixedFont")*

*self.Entry5.configure(foreground="#000000")*

*self.Entry5.configure(insertbackground="black")*

*self.mo=StringVar() self.Entry5.configure(textvariable=self.mo)def vehicle\_insert(self):*

*self.uidd=self.uid.get()self.n1=self.rn.get() self.n2=self.ty.get()self.n3=self.y.get() self.n4=self.ma.get() self.n5=self.mo.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into vehicle(registration\_num,licence\_num,type,year,make,model)values(%s,(select licence\_num from licence where uid=%s),%s,%s,%s,%s)",(self.n1,self.uidd,self.n2,self.n3,self.n4,self.n5))*

*if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9") self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor)*

*self.Canvas1.destroy()self.initial()*

*self.root.title("FOOD\_GRAIN INSERT PAGE") self.Entry7 = tk.Entry(self.Canvas1)*

*self.Entry7.place(relx=0.464, rely=0.105,height=20, relwidth=0.272)*

*self.Entry7.configure(background="white")*

*self.Entry7.configure(disabledforeground="#a3a3a3")*

*self.Entry7.configure(font="TkFixedFont")*

*self.Entry7.configure(foreground="#000000")*

*self.Entry7.configure(insertbackground="black")*

*self.uid=StringVar()*

*self.Entry7.configure(textvariable=self.uid)*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.149, rely=0.01, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following information:''')*

*self.Message1.configure(width=1000)*

*self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.415, rely=0.158,height=20, relwidth=0.388)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.co=StringVar()*

*self.Entry1.configure(textvariable=self.co)*

*self.Entry2 = tk.Entry(self.Canvas1)*

*self.Entry2.place(relx=0.415, rely=0.226,height=20, relwidth=0.388)*

*self.Entry2.configure(background="white")*

*self.Entry2.configure(disabledforeground="#a3a3a3")*

*self.Entry2.configure(font="TkFixedFont")*

*self.Entry2.configure(foreground="#000000") self.Entry2.configure(insertbackground="black")*

*self.doi=StringVar() self.Entry2.configure(textvariable=self.doi)*

*self.Entry3 = tk.Entry(self.Canvas1)*

*self.Entry3.place(relx=0.415, rely=0.293,height=20, relwidth=0.388)*

*self.Entry3.configure(background="white")*

*self.Entry3.configure(disabledforeground="#a3a3a3")*

*self.Entry3.configure(font="TkFixedFont")*

*self.Entry3.configure(foreground="#000000")*

*self.Entry3.configure(insertbackground="black")*

*self.rq=StringVar()self.Entry3.configure(textvariable=self.rq)self.Entry4 = tk.Entry(self.Canvas1)*

*self.Entry4.place(relx=0.415, rely=0.361,height=20, relwidth=0.388)*

*self.Entry4.configure(background="white")*

*self.Entry4.configure(disabledforeground="#a3a3a3")*

*self.Entry4.configure(font="TkFixedFont")*

*self.Entry4.configure(foreground="#000000")*

*self.Entry4.configure(insertbackground="black")*

*self.wq=StringVar()self.Entry4.configure(textvariable=self.wq)self.Entry5 = tk.Entry(self.Canvas1)*

*self.Entry5.place(relx=0.415, rely=0.429,height=20, relwidth=0.388)*

*self.Entry5.configure(background="white")*

*self.Entry5.configure(disabledforeground="#a3a3a3")*

*self.Entry5.configure(font="TkFixedFont")*

*self.Entry5.configure(foreground="#000000")*

*self.Entry5.configure(insertbackground="black")*

*self.sa=StringVar() self.Entry5.configure(textvariable=self.sa)*

*self.Entry7 = tk.Entry(self.Canvas1)*

*self.Entry7.place(relx=0.415, rely=0.500,height=20, relwidth=0.388)*

*self.Entry7.configure(background="white")*

*self.Entry7.configure(disabledforeground="#a3a3a3")*

*self.Entry7.configure(font="TkFixedFont")*

*self.Entry7.configure(foreground="#000000")*

*self.Entry7.configure(insertbackground="black")*

*self.oi=StringVar() self.Entry7.configure(textvariable=self.oi)*

*def food\_insert(self):*

*self.uidd=self.uid.get() self.n1=self.co.get() self.n2=self.doi.get()self.n3=self.rq.get()self.n4=self.wq.get() self.n5=self.sa.get()self.n6=self.oi.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into food\_grains values((select ration\_id from ration where huid=%s),%s,%s,%s,%s,%s,%s)",(self.uidd,self.n1,self.n2,self.n3,self.n4,self.n5,self.n6)) if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1) self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor)*

*def userr(self):self.Canvas1.destroy()*

*self.initial()self.root.title("USER PAGE") self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.05, rely=0.15, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")self.Message1.configure(text='''Enter Uid:''')*

*self.Message1.configure(width=1000)*

*self.uid=StringVar()*

*self.Entry1.configure(textvariable=self.uid)*

*def userdata\_fetch(self):*

*self.Canvas1.destroy()*

*self.initial() self.root.title("USERDATA\_FETCH PAGE")*

*def health\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select huid,type\_of\_card,amount,no\_of\_people from health where huid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy()*

*self.initial()*

*if(len(record)==0):*

*headlist=['Huid:','Type of Card:','Amount','Number of People:']*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])):*

*#mycursor.execute("select \* from person p join health h on p.p\_health\_id=h.health\_id where h.huid=%s",(huid,))*

*#record1=list(mycursor.fetchall())*

*def lpg\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select huid,date\_of\_issue,deposit,distributer\_code,distributer\_name,distributer\_address from lpg where huid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy()*

*self.initial()*

*if(len(record)==0):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.6 , relwidth=0.500)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Information does not exsits''')*

*self.Message1.configure(width=2000)*

*headlist=['Huid','Date of issue:','Deposit','Distributer Code','Distributer Name','Distributer Address']*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])):*

*self.Label1 = tk.Label(self.Frame1)*

*self.Label1.grid(row=j+2,column=i)*

*self.Label1.configure(background="#d9d9d9")*

*self.Label1.configure(disabledforeground="#a3a3a3")*

*self.Label1.configure(foreground="#000000")*

*self.Label1.configure(text=record[i][j]) #mycursor.execute("select \* from person p join lpg l on p.p\_lpg\_id=l.lpg\_id where l.huid=%s",(huid,))*

*#record1=list(mycursor.fetchall() def licence\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select licence\_num,type\_of\_licence from licence where uid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy()self.initial()*

*self.root.title("LICENSE PAGE")*

*if(len(record)==0):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.6 , relwidth=0.500)*

*self.Message1.configure(background="#45d8ac")self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Information does not exsits''')*

*self.Message1.configure(width=2000) headlist=['Licence Num:','Type of Licence:']*

*record.insert(0,headlist) for i in range(len(record)):*

*for j in range(len(record[i])):*

*self.Label1 = tk.Label(self.Frame1) self.Label1.grid(row=j+2,column=i)*

*self.Label1.configure(background="#d9d9d9")self.Label1.configure(disabledforeground="#a3a3a3")*

*self.Label1.configure(foreground="#000000")self.Label1.configure(text=record[i][j]) #mycursor.execute("select \* from vehicle where licence\_num=(select licence\_num from licence where uid=%s)",(uid,))*

*#record1=list(mycursor.fetchall() def transport\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select start\_point,final\_reach,issued,expiry from transport where uid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy() self.initial()*

*self.root.title("BUSPASS PAGE")if(len(record)==0):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.6 , relwidth=0.500)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Information does not exsits''')*

*self.Message1.configure(width=2000)*

*headlist=['Start Point:','Final Reach:','Date of Issue:','Date of Expiry:']*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])):*

*def pan\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select type\_of\_holder from pan\_card where uid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy()*

*self.initial()*

*self.root.title("PANCARD PAGE")*

*if(len(record)==0):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.6 , relwidth=0.500)self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000") self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Information does not exsits''')*

*self.Message1.configure(width=2000)self.Button2 = tk.Button(self.Canvas1)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.051 , relwidth=0.212)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Pan infromation:''')*

*self.Message1.configure(width=1000)self.Frame1 = tk.Frame(self.Canvas1)*

*self.Frame1.place(relx=0.066, rely=0.130, relheight=0.800 , relwidth=0.871)*

*self.Frame1.configure(relief='groove')*

*self.Frame1.configure(borderwidth="2")*

*self.Frame1.configure(relief="groove")*

*self.Frame1.configure(background="#d9d9d9")*

*headlist=['Type of holder:',]*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])) def ration\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select huid,type,no\_of\_people from ration where huid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())*

*self.Canvas1.destroy()*

*self.initial()self.root.title("RATIONCARD PAGE")*

*if(len(record)==0):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.6 , relwidth=0.500)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Information does not exsits''')*

*self.Message1.configure(width=2000)self.Frame1 = tk.Frame(self.Canvas1)*

*self.Frame1.place(relx=0.066, rely=0.130, relheight=0.800 , relwidth=0.871)*

*self.Frame1.configure(relief='groove')*

*self.Frame1.configure(borderwidth="2")*

*self.Frame1.configure(relief="groove")*

*self.Frame1.configure(background="#d9d9d9")*

*headlist=['Huid:','type of Card(APL or BPL):','no of people:']*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])):*

*self.Label1 = tk.Label(self.Frame1)*

*self.Label1.grid(row=j+2,column=i)*

*self.Label1.configure(background="#d9d9d9")*

*self.Label1.configure(disabledforeground="#a3a3a3")*

*self.Label1.configure(foreground="#000000")*

*self.Label1.configure(text=record[i][j])'''mycursor.execute("select uid ,first\_name,last\_name from person p join ration r on p.p\_ration\_id=r.ration\_id where r.huid=%s",(self.uidd,))*

*record1=list(mycursor.fetchall())*

*headlist=['uid','Fname Name','Last Name']*

*record1.insert(0,headlist)*

*for k in range(len(record)):*

*for l in range(len(record[k])):*

*def person\_retrive(self):*

*self.uidd=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("select first\_name,last\_name,Dob,phone,blood\_grp,address\_street,address\_city,address\_state,address\_pin from person where uid=%s",(self.uidd,))*

*record=list(mycursor.fetchall())self.Canvas1.destroy() self.initial()*

*self.root.title("PERSON\_RETRIVE INSERT PAGE")*

*if(len(record)==0):*

*else:self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.083, rely=0.088, relheight=0.051 , relwidth=0.212)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Personal infromation:''')*

*self.Message1.configure(width=1000)self.Button1 = tk.Button(self.Frame1)*

*headlist=['First Name:','Last Name:','Dob:','Phone:','Blood Grp:','Address:','City:','State:','Pin:']*

*record.insert(0,headlist)*

*for i in range(len(record)):*

*for j in range(len(record[i])):*

*self.Label1 = tk.Label(self.Frame1)*

*self.Label1.grid(row=j+2,column=i)*

*self.Label1.configure(background="#d9d9d9")*

*self.Label1.configure(disabledforeground="#a3a3a3")*

*self.Label1.configure(foreground="#000000")*

*self.Label1.configure(text=record[i][j])*

*def admin\_login(self):*

*self.Canvas1.destroy()*

*self.initial()*

*self.root.title("INSERT PAGE")*

*def login\_db(self):*

*admin\_id=self.e3.get()*

*password=self.e4.get()*

*self.login\_admin(admin\_id,password)*

*def login\_admin(self,admin\_id,password):*

*conn,cursor=self.connectt()*

*cursor.execute("SELECT username FROM admin WHERE username= %s",(admin\_id,))*

*for row in cursor:*

*i=i+1*

*if(i==1):*

*cursor.execute("SELECT password FROM admin WHERE password= %s AND username =%s",(password,admin\_id))*

*for row in cursor:*

*j=j+1*

*if(j==1):*

*self.Canvas1.destroy()*

*self.initial()*

*self.admin\_page()*

*if(i==0):*

*self.admin\_login()*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.381, rely=0.632, relheight=0.086 , relwidth=0.23)*

*self.Message1.configure(background="#d9d9d9")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Invalid Username''')*

*self.Message1.configure(width=300)*

*elif(i==1 and j==0):*

*self.admin\_login()*

*self.Message2 = tk.Message(self.Canvas1)*

*self.Message2.place(relx=0.381, rely=0.632, relheight=0.086 , relwidth=0.23)*

*self.Message2.configure(background="#d9d9d9")*

*self.Message2.configure(foreground="#ed3d11")*

*self.Message2.configure(highlightbackground="#d9d9d9")*

*self.Message2.configure(highlightcolor="black")*

*self.Message2.configure(text='''Invalid Password''')*

*self.Message2.configure(width=300)*

*self.close(cursor,conn)def admin\_page(self):*

*self.Canvas1.destroy()*

*self.initial()*

*self.root.title("ADMIN\_PAGE INSERT")*

*def insert\_options(self):*

*self.Canvas1.destroy()*

*self.initial()*

*self.root.title("INSERT\_OPTIONS INSERT")*

*self.initial()*

*self.root.title("ADHAR INSERT")*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.149, rely=0.09, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")*

*self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.415, rely=0.158,height=20, relwidth=0.388)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.fn=StringVar() self.Entry1.configure(textvariable=self.fn) self.Entry2 = tk.Entry(self.Canvas1)*

*self.Entry2.place(relx=0.415, rely=0.226,height=20, relwidth=0.388)*

*self.Entry2.configure(background="white")*

*self.Entry2.configure(disabledforeground="#a3a3a3")*

*self.Entry2.configure(font="TkFixedFont")*

*self.Entry2.configure(foreground="#000000")*

*self.Entry2.configure(insertbackground="black")*

*self.ln=StringVar()*

*self.Entry2.configure(textvariable=self.ln) self.Label1 = tk.Label(self.Canvas1)*

*self.Label1.place(relx=0.182, rely=0.158, height=21, width=64)*

*self.Label1.configure(background="#45d8ac")*

*self.Label1.configure(disabledforeground="#a3a3a3")*

*self.Label1.configure(foreground="#000000")*

*self.Label1.configure(text='''First name:''')*

*def perop(self):self.fna=self.fn.get()*

*self.lna=self.ln.get()self.addss=self.adds.get() self.addcc=self.addc.get()*

*self.addstt=self.addst.get()self.pno=self.pn.get()*

*self.dobb=self.dob.get() self.bgg=self.bg.get()self.pii=self.pi.get()*

*self.person\_insert(self.dobb,self.fna,self.lna,self.addss,self.addcc,self.addstt,self.pii,self.pno,self.bgg)*

*def person\_insert(self,dob,fn,ln,adds,addc,addst,addp,ph,bg):*

*mydb,mycursor=self.connectt()mycursor.execute("SET FOREIGN\_KEY\_CHECKS=0")*

*if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''') self.Message1.configure(width=1000)*

*mydb.commit()*

*self.close(mydb,mycursor)*

*def ratin(self): self.Canvas1.destroy()*

*self.initial()*

*self.root.title("RATIONCARD INSERT")*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.133, rely=0.066, relheight=0.051 , relwidth=0.255)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following details:''')*

*self.Message1.configure(width=1000)*

*def ration\_insert(self):*

*self.n1=self.huid.get() self.n2=self.toc.get() self.n3=self.nop.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into ration values(%s,%s,%s,%s)",("default",self.n1,self.n2,self.n3))*

*mydb.commit() mycursor.execute("update person set person.p\_ration\_id=(select ration\_id from ration where huid=%s)where uid=%s",(self.n1,self.n1))*

*mydb.commit() if(mycursor.rowcount==1):*

*def aptec(self):*

*self.Canvas1.destroy()*

*self.initial()self.root.title("UPDATE PERSON TO RATION CARD")*

*def aptecc(self):*

*self.n1=self.huid.get()*

*self.n2=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("update person set person.p\_ration\_id=(select ration\_id from ration where huid=%s)where uid=%s",(self.n1,self.n2))*

*mydb.commit()if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1) self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9") self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: mydb.commit()*

*self.close(mydb,mycursor) def bpin(self):*

*self.Canvas1.destroy()*

*self.initial() self.root.title("BUSPASS INSERT")*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.116, rely=0.088, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following information:''')*

*self.Message1.configure(width=1000) self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.415, rely=0.177,height=20, relwidth=0.355)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.uid=StringVar()*

*self.Entry1.configure(textvariable=self.uid)*

*def transport\_insert(self): self.uidd=self.uid.get() self.stt=self.st.get() self.fpp=self.fp.get() self.doii=self.doi.get() self.doee=self.doe.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into transport values(default,%s,%s,%s,%s,%s)",(self.uidd,self.stt,self.fpp,self.doii,self.doee))*

*mydb.commit() if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Value Inserted''') self.Message1.configure(width=1000)*

*else:self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor)*

*def licence\_insert(self):*

*self.n1=self.uid.get() self.n2=self.tol.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into licence values(default,%s,%s)",(self.n1,self.n2))*

*mydb.commit()*

*if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.100, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")self.Message1.configure(text='''Error in inserting the value''')*

*self.Message1.configure(width=1000)*

*self.close(mydb,mycursor)*

*def lpg\_insert(self):*

*self.n1=self.doi.get()self.n2=self.dp.get()self.n3=self.dc.get()*

*self.n4=self.dn.get() self.n5=self.da.get()self.n6=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into lpg values(default,%s,%s,%s,%s,%s,%s)",(self.n1,self.n2,self.n3,self.n4,self.n5,self.n6))*

*mydb.commit() if(mycursor.rowcount==1):*

*self.close(mydb,mycursor) def aptecl(self):*

*self.Canvas1.destroy() self.initial()*

*self.root.title("UPDATE PERSON UID TO LPG")*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.133, rely=0.066, relheight=0.051 , relwidth=0.255)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following details:''')*

*self.Message1.configure(width=1000)self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.448, rely=0.155,height=20, relwidth=0.272)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.huid=StringVar() self.Entry1.configure(insertbackground="black")*

*self.Entry1.configure(textvariable=self.huid)*

*def apteccl(self):*

*self.n1=self.huid.get() self.n2=self.uid.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("update person set person.p\_lpg\_id=(select lpg\_id from lpg where huid=%s)where uid=%s",(self.n1,self.n2))*

*mydb.commit()if(mycursor.rowcount==1):*

*def health\_insert(self):*

*self.n1=self.uid.get()self.n2=self.np.get()self.n3=self.amount.get()self.n4=self.ty.get()*

*mydb,mycursor=self.connectt()*

*mycursor.execute("insert into health values(default,%s,%s,%s,%s)",(self.n1,self.n2,self.n3,self.n4))*

*mydb.commit()mycursor.execute("update person set person.p\_health\_id=(select health\_id from health where huid=%s)where uid=%s",(self.n1,self.n1))*

*mydb.commit() if(mycursor.rowcount==1):*

*def aptecclh(self):*

*self.Canvas1.destroy()*

*self.initial() self.root.title("UPDATE\_PERSON\_UID\_TO\_HEALTHCARD")*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.133, rely=0.066, relheight=0.051 , relwidth=0.255)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following details:''')*

*self.Message1.configure(width=1000)*

*self.Label2 = tk.Label(self.Canvas1)*

*self.Label2.place(relx=0.133, rely=0.243, height=21, width=146)*

*self.Label2.configure(background="#45d8ac")*

*self.Label2.configure(disabledforeground="#a3a3a3")*

*self.Label2.configure(foreground="#000000")*

*self.Label2.configure(text='''Uid:''')*

*self.Entry2 = tk.Entry(self.Canvas1)*

*self.Entry2.place(relx=0.448, rely=0.243,height=20, relwidth=0.272)*

*self.Entry2.configure(background="white")*

*self.Entry2.configure(disabledforeground="#a3a3a3")*

*self.Entry2.configure(font="TkFixedFont")*

*self.Entry2.configure(foreground="#000000")*

*self.uid=StringVar()*

*self.Entry2.configure(insertbackground="black")*

*self.Entry2.configure(textvariable=self.uid)*

*def update\_options(self):*

*self.Canvas1.destroy()*

*self.initial()self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.149, rely=0.068, relheight=0.052, relwidth=0.187)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Select the Options:''')*

*self.Message1.configure(width=1000) def upa(self):*

*self.Canvas1.destroy()*

*self.initial() self.root.title("UPDATE\_ADDRESS")self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.133, rely=0.135, relheight=0.052, relwidth=0.303)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black") self.Message1.configure(text='''Enter the following information:''')*

*self.Message1.configure(width=1000)self.Entry1 = tk.Entry(self.Canvas1)*

*self.e3=StringVar()*

*self.Entry1.configure(textvariable=self.e3) def updateadd(self):*

*self.n1=self.e3.get()self.n2=self.e4.get() self.n3=self.e5.get() self.n4=self.e6.get()self.n5=self.e7.get()*

*self.update\_person\_add(self.n2,self.n3,self.n4,self.n5,self.n1)*

*def update\_person\_add(self,adds,addc,addst,addp,uid):*

*mydb,mycursor=self.connectt()*

*mycursor.execute("update person set address\_street=%s , address\_city=%s , address\_state=%s , address\_pin=%s where uid=%s",(adds,addc,addst,addp,uid))*

*if(mycursor.rowcount==1):*

*self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.200, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black") self.Message1.configure(text='''Value Inserted''')*

*self.Message1.configure(width=1000)*

*else: self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.200, rely=0.850, relheight=0.086 , relwidth=0.25)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#ed3d11")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Error in inserting the value''') self.Message1.configure(width=1000)*

*mydb.commit() self.close(mydb,mycursor) def upp(self):*

*self.Canvas1.destroy()*

*self.initial()*

*self.root.title("UPDATE\_PHONE\_NUMBER")self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.133, rely=0.135, relheight=0.052 , relwidth=0.303)*

*self.Message1.configure(background="#45d8ac")*

*self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black")*

*self.Message1.configure(text='''Enter the following information:''')*

*self.Message1.configure(width=1000)*

*self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.415, rely=0.248,height=20, relwidth=0.388)*

*self.Entry1.configure(background="white")*

*self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")*

*self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.uid=StringVar() self.Entry1.configure(textvariable=self.uid)*

*def Delete(self): self.Canvas1.destroy()self.initial()*

*self.root.title("DELETE")*

*self.Entry1 = tk.Entry(self.Canvas1)*

*self.Entry1.place(relx=0.367, rely=0.271,height=20, relwidth=0.273)*

*self.Entry1.configure(background="white")self.Entry1.configure(disabledforeground="#a3a3a3")*

*self.Entry1.configure(font="TkFixedFont")self.Entry1.configure(foreground="#000000")*

*self.Entry1.configure(insertbackground="black")*

*self.e3=StringVar()self.Entry1.configure(textvariable=self.e3)self.Message1 = tk.Message(self.Canvas1)*

*self.Message1.place(relx=0.167, rely=0.158, relheight=0.052, relwidth=0.197)*

*self.Message1.configure(background="#45d8ac") self.Message1.configure(foreground="#000000")*

*self.Message1.configure(highlightbackground="#d9d9d9")*

*self.Message1.configure(highlightcolor="black") self.Message1.configure(text='''Enter the Unique Id:''')*

*self.Message1.configure(width=1000) #def dell(self):*

*#self.uidd=self.e3.get() #self.delete(self.uidd)*

*def delete(self):*

*uidd=self.e3.get()*

*mydb,mycursor=self.connectt()*

*'''mycursor.execute("select licence\_num from licence where uid=%s",(uidd,))*

*ln=mycursor.fetchall()*

*mycursor.execute("select \* from licence where licence\_num=%s",(ln[0][0],))*

*vehcount=mycursor.fetchall()*

*if(int(vehcount[0][0])>0):*

*vo=int(input("Enter the owner for other vehicles:\n"))*

*mycursor.execute("update vehicle set licence\_num=%s where licence\_num=%s",(vo,ln[0][0]))*

*mydb.commit()'''*

*mycursor.execute("delete from licence where uid=%s",(uidd,))*

*mydb.commit() mycursor.execute("delete from pan\_card where uid=%s",(uidd,))*

*mydb.commit() mycursor.execute("delete from transport where uid=%s",(uidd,))*

*mydb.commit()mycursor.execute("select ration\_id from ration where huid=%s",(uidd,))*

*ration\_id=mycursor.fetchall()mycursor.execute("select type from ration where huid=%s",(uidd,))*

*typee=mycursor.fetchall()mycursor.execute("select no\_of\_people from ration where huid=%s",(uidd,))*

*nop=mycursor.fetchall()mycursor.execute("delete from ration where huid=%s",(uidd,))*

*mydb.commit() d=mycursor.rowcountmycursor.execute("delete from person where uid=%s",(uidd,))*

*else: #n=int(input("Enter other huid:\n"))*

*mycursor.execute("select uid from person where p\_ration\_id=%s limit 1",(ration\_id[0][0],))*

*n=mycursor.fetchall() d1=mycursor.rowcount*

*if(d1!=0):*

*mycursor.execute("insert into ration values(%s,%s,%s,%s)",(ration\_id[0][0],n[0][0],typee[0][0],nop[0][0]-1))*

*mydb.commit() mycursor.execute("select health\_id from health where huid=%s",(uidd,))*

*health\_id=mycursor.fetchall()mycursor.execute("select amount from health where huid=%s",(uidd,))*

*amount=mycursor.fetchall()mycursor.execute("select no\_of\_people from health where huid=%s",(uidd,))*

*nop=mycursor.fetchall()mycursor.execute("select type\_of\_card from health where huid=%s",(uidd,))*

*toc=mycursor.fetchall() mycursor.execute("delete from health where huid=%s",(uidd,))*

*mydb.commit()d=mycursor.rowcount mycursor.execute("select lpg\_id from lpg where huid=%s",(uidd,))*

*lpg\_id=mycursor.fetchall() mycursor.execute("select date\_of\_issue from lpg where huid=%s",(uidd,))*

*doi=mycursor.fetchall() mycursor.execute("select deposit from lpg where huid=%s",(uidd,))*

*dep=mycursor.fetchall()mycursor.execute("select distributer\_code from lpg where huid=%s",(uidd,))*

*dc=mycursor.fetchall() mycursor.execute("select distributer\_name from lpg where huid=%s",(uidd,))*

*dn=mycursor.fetchall() mycursor.execute("select distributer\_address from lpg where huid=%s",(uidd,))*

*da=mycursor.fetchall()mycursor.execute("delete from lpg where huid=%s",(uidd,))*

*mydb.commit() d=mycursor.rowcount*

*if d==0:*

*mycursor.execute("delete from person where uid=%s",(uidd,))*

*mydb.commit()*

*else:*

*#n=int(input("Enter other huid:\n"))*

*mycursor.execute("select uid from person where p\_lpg\_id=%s limit 1",(lpg\_id[0][0],))*

*n=mycursor.fetchall()*

*d1=mycursor.rowcount*

*if(d1!=0):*

*mycursor.execute("insert into lpg values(%s,%s,%s,%s,%s,%s,%s)",(lpg\_id[0][0],doi[0][0],dep[0][0],dc[0][0],dn[0][0],da[0][0],n[0][0])) mydb.commit()*

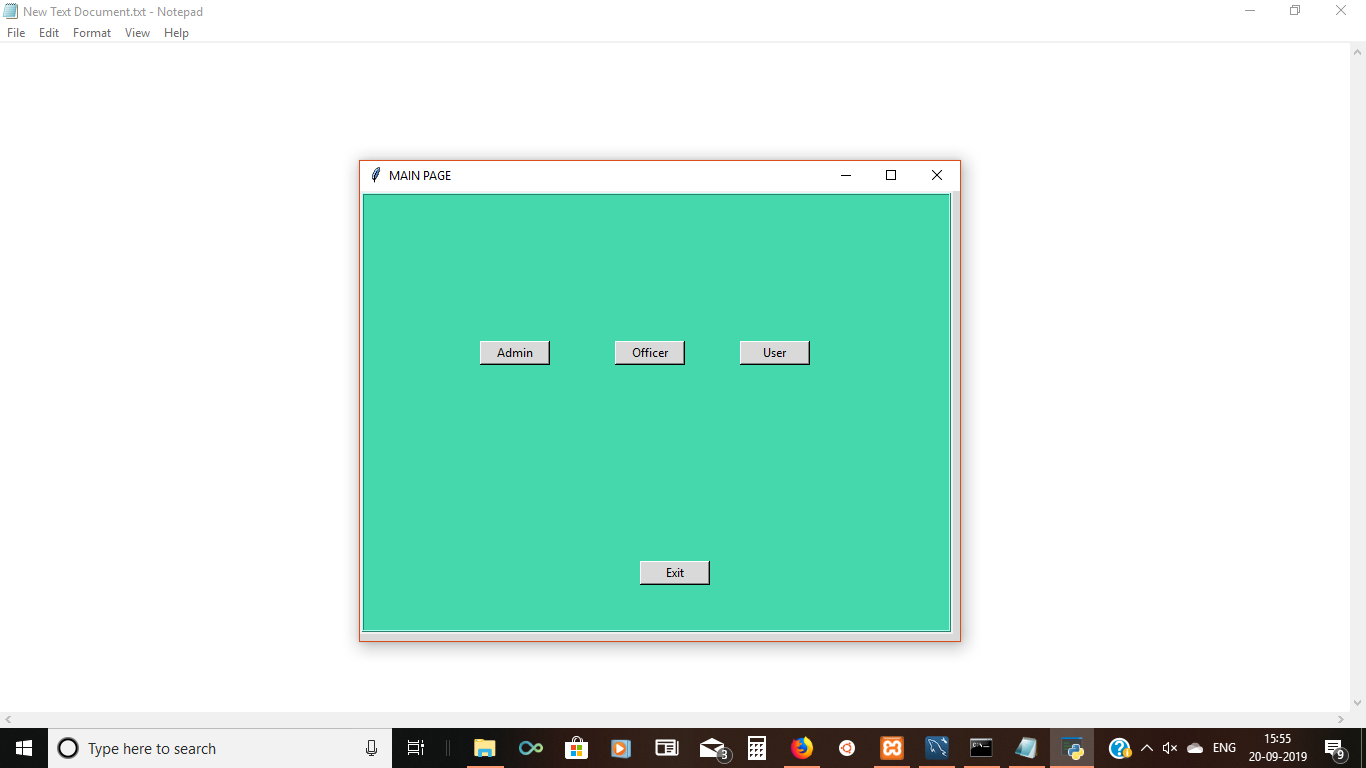
*self.close(mydb,mycursor)*

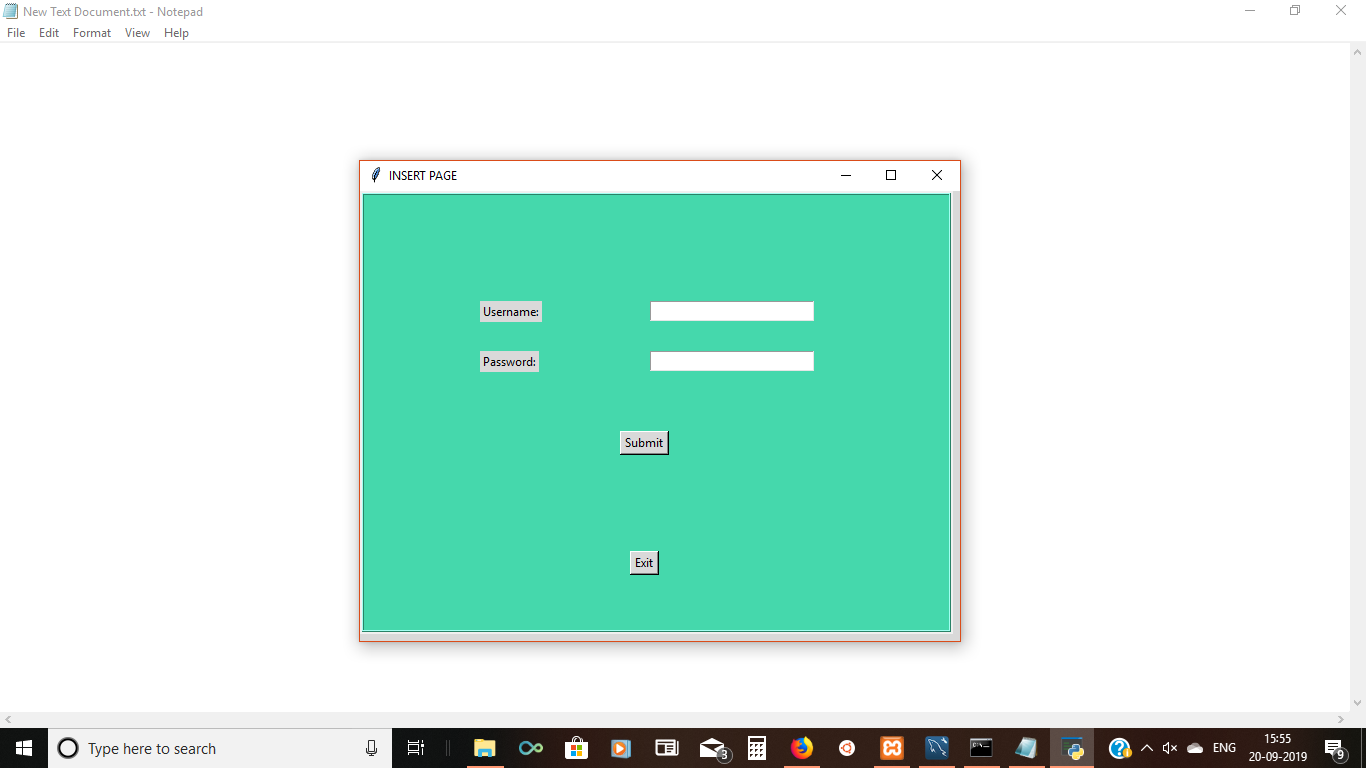
*ob=projectcode()*

*ob.start()*

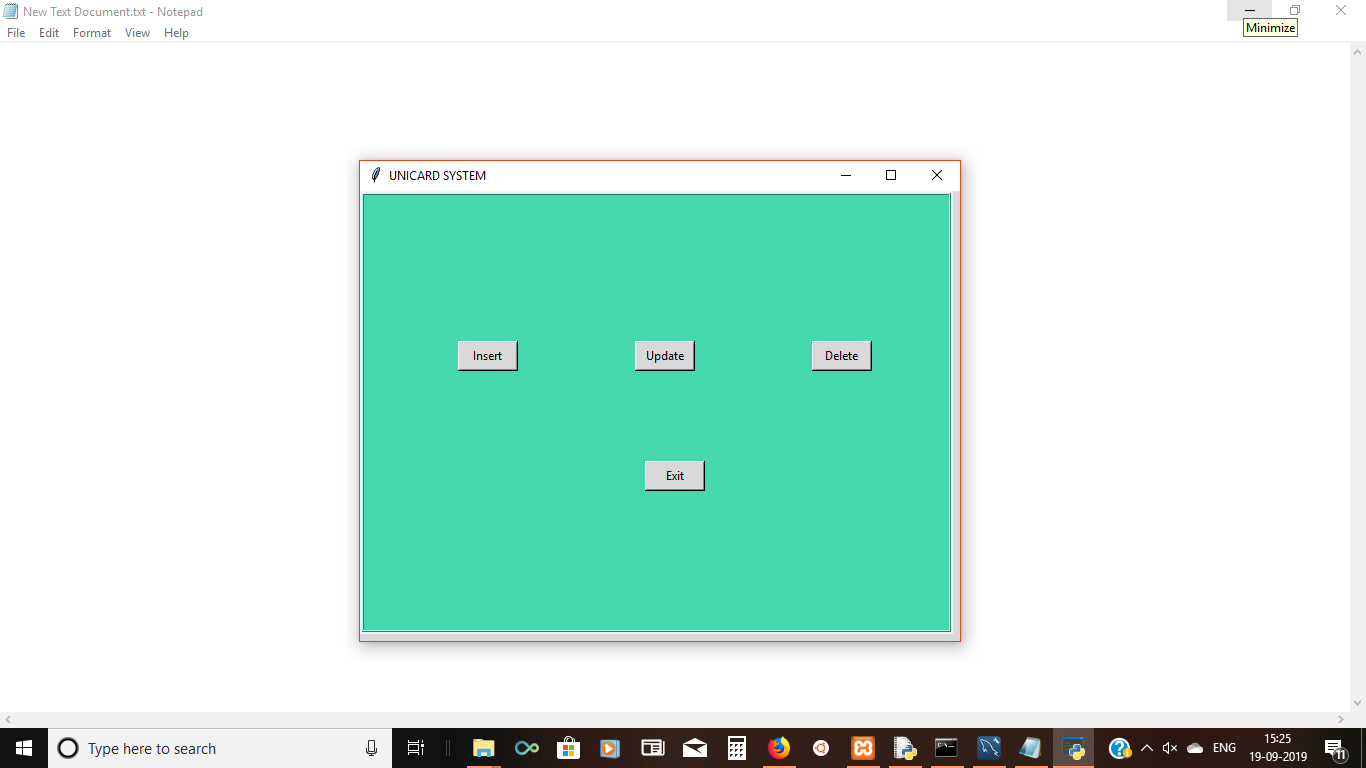
**Project demo:**

***Main login page***

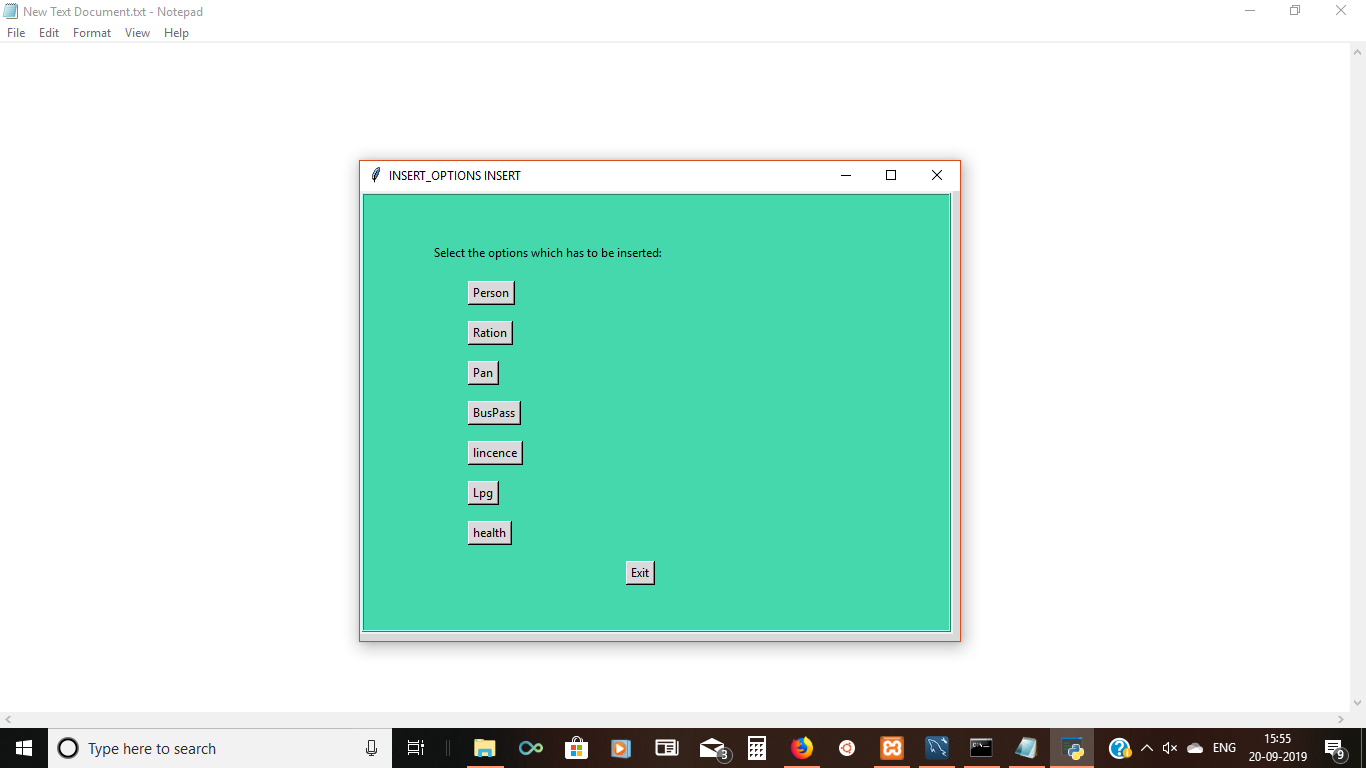


***Admin login page***

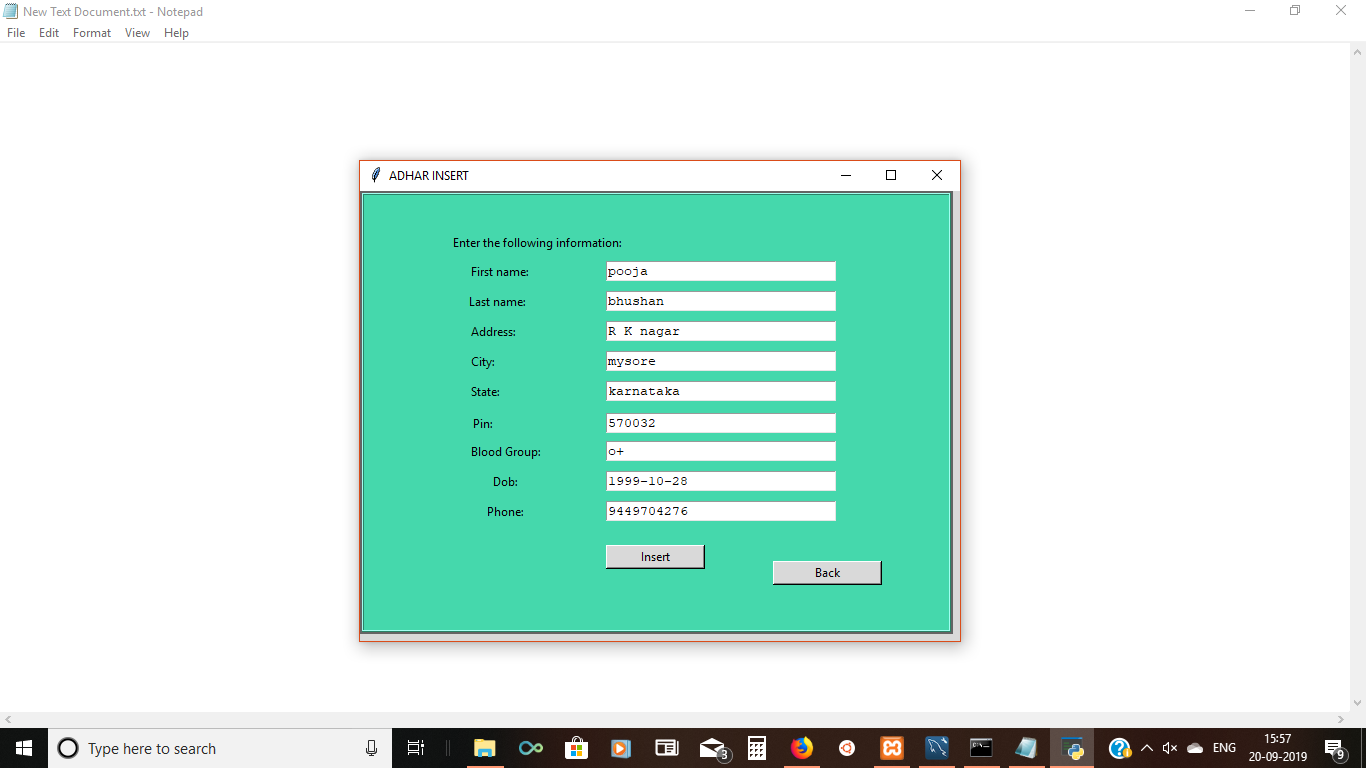
***Admin page***



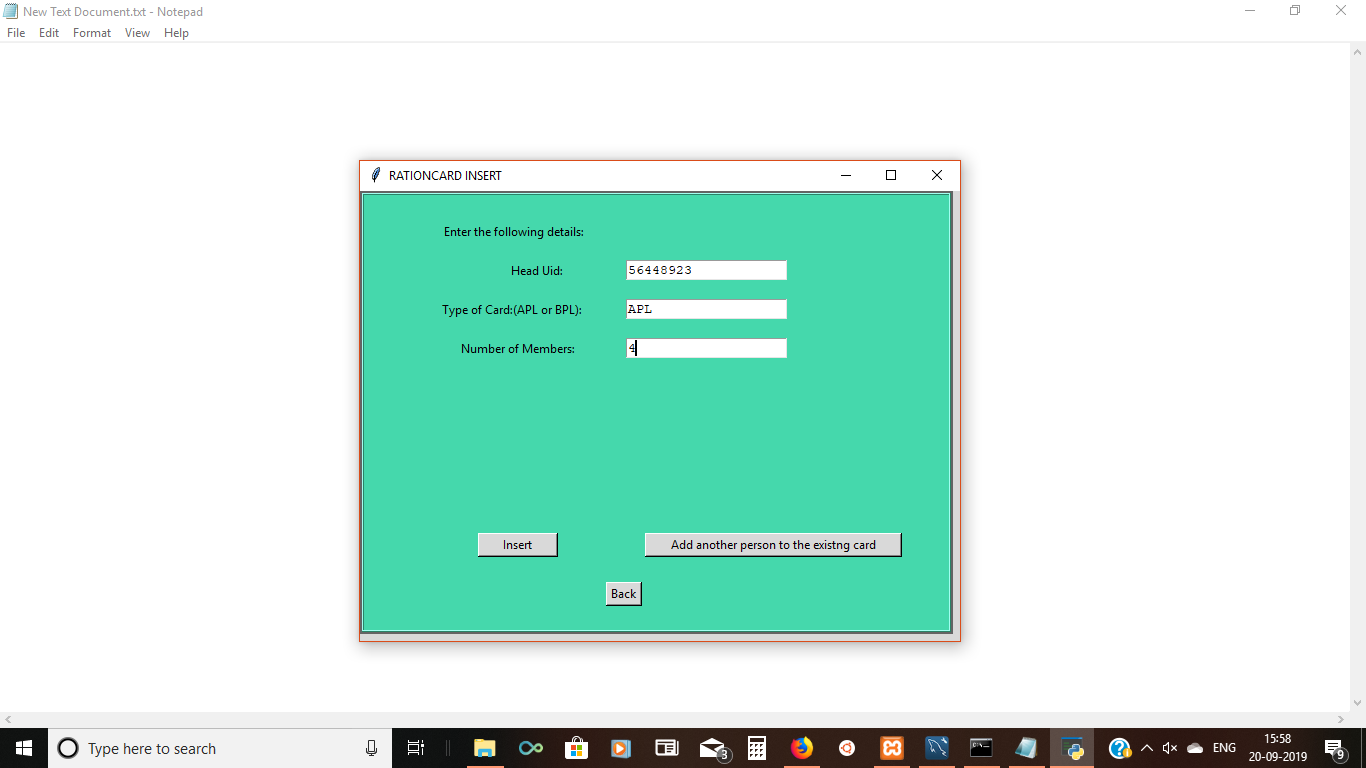
***Admin insert Option page***

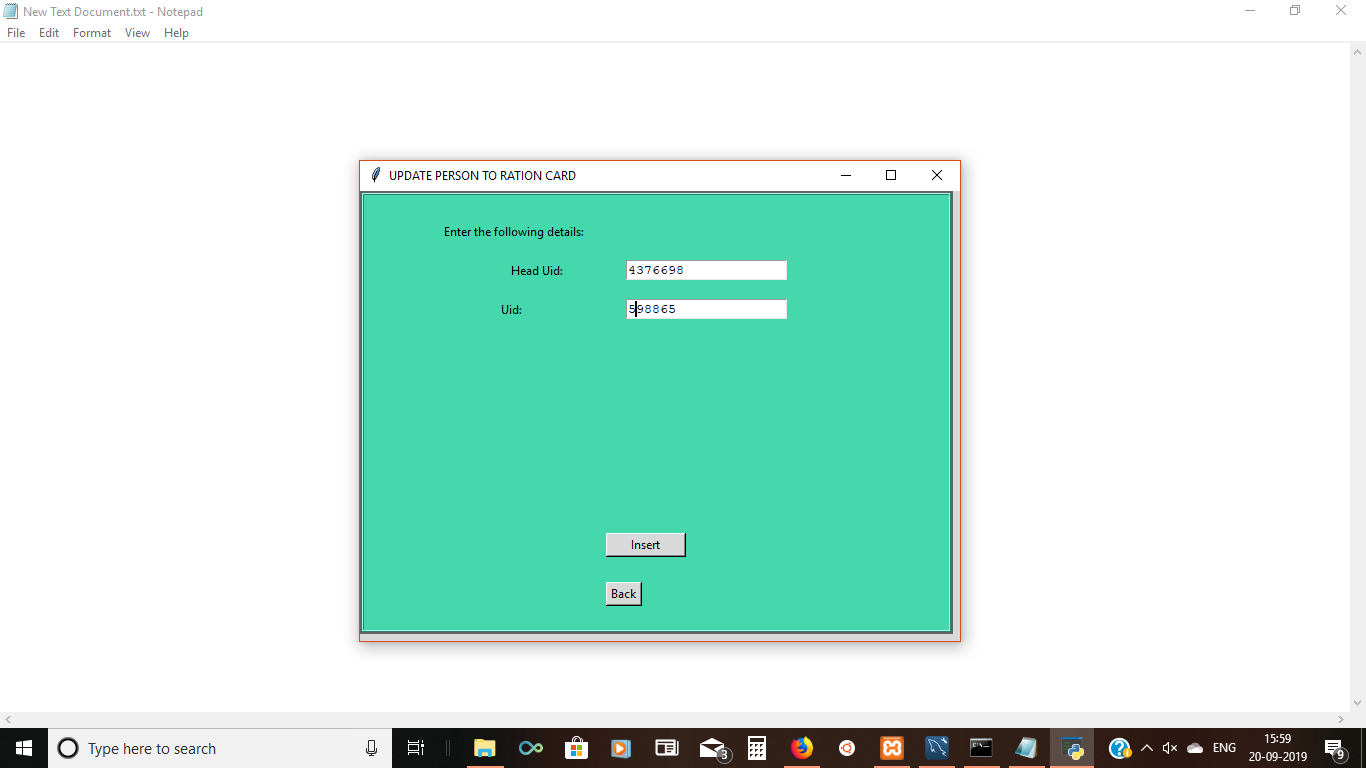


***Person insert page***

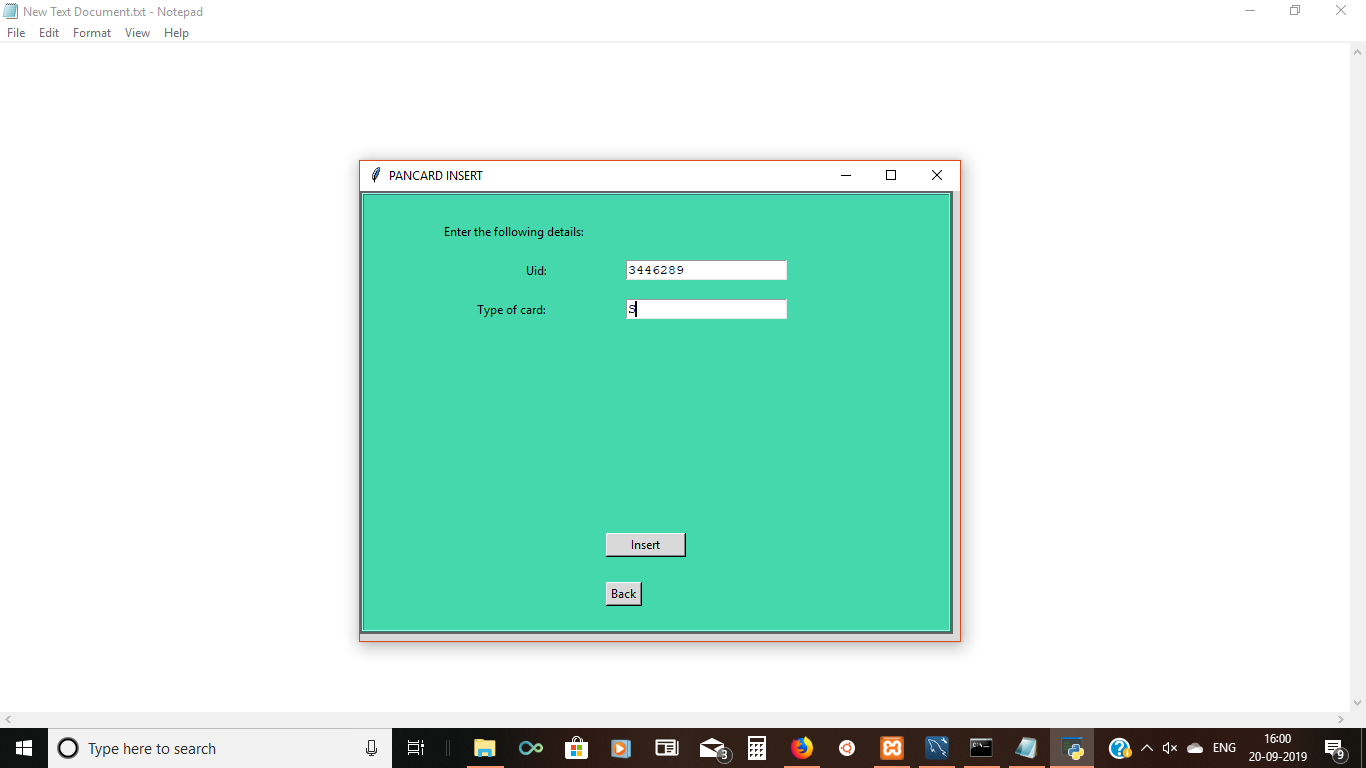


***Ration card insert page***

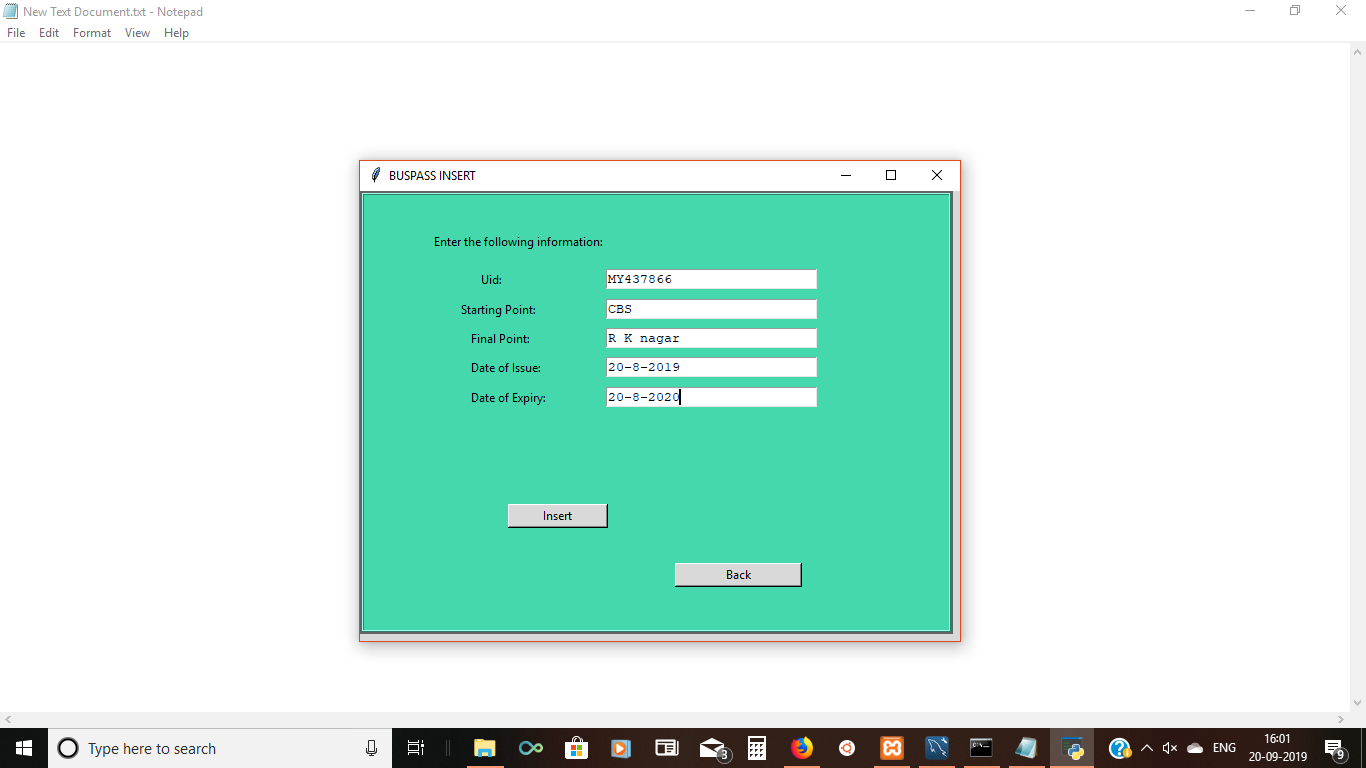


***Update Ration card***

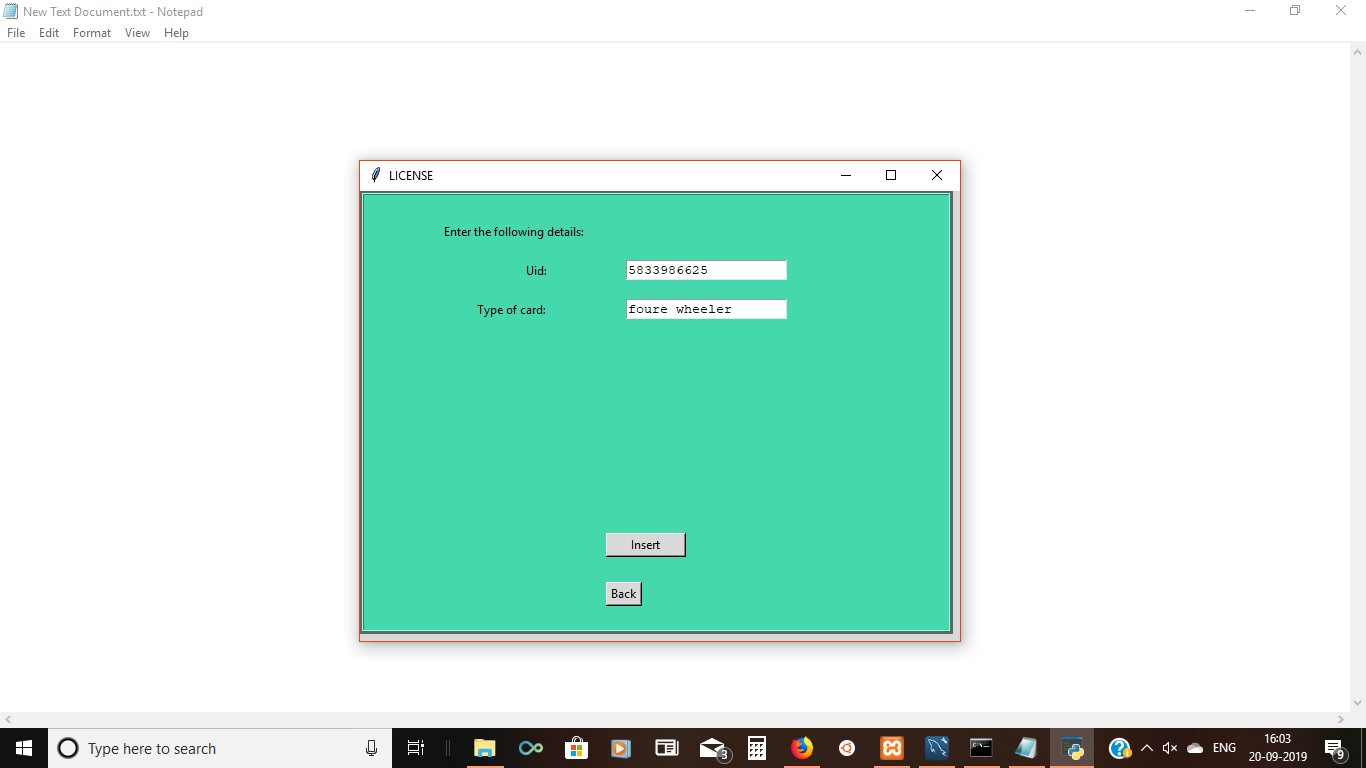
***Pan card insert page***



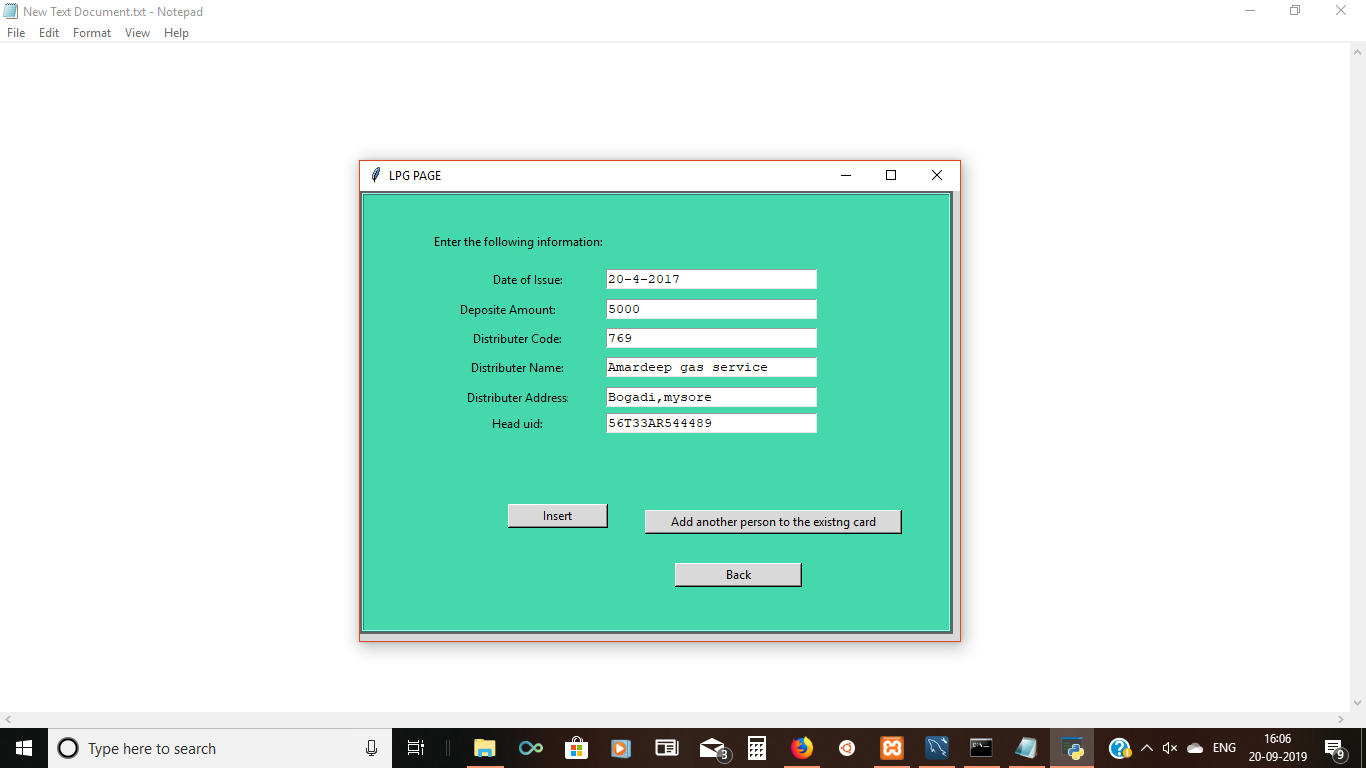
***BusPass insert page***



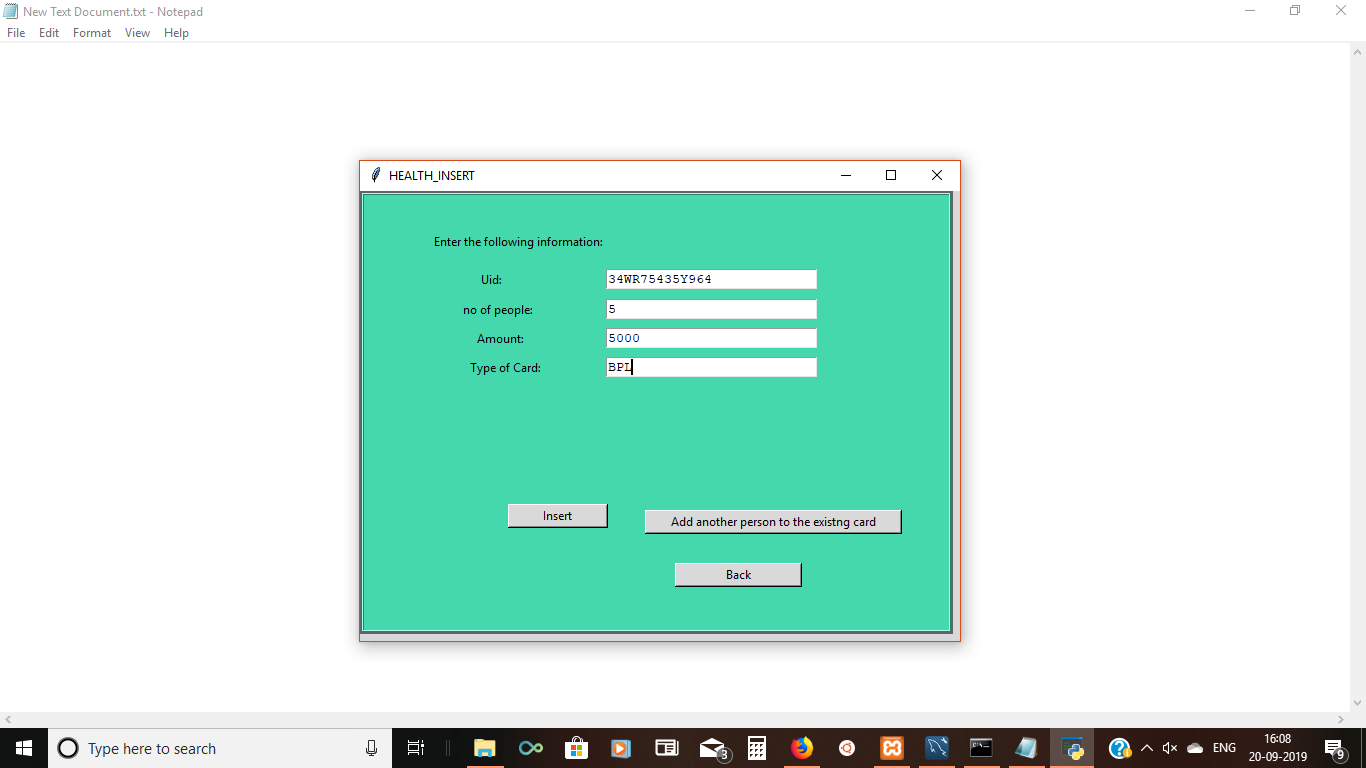
***License insert page***



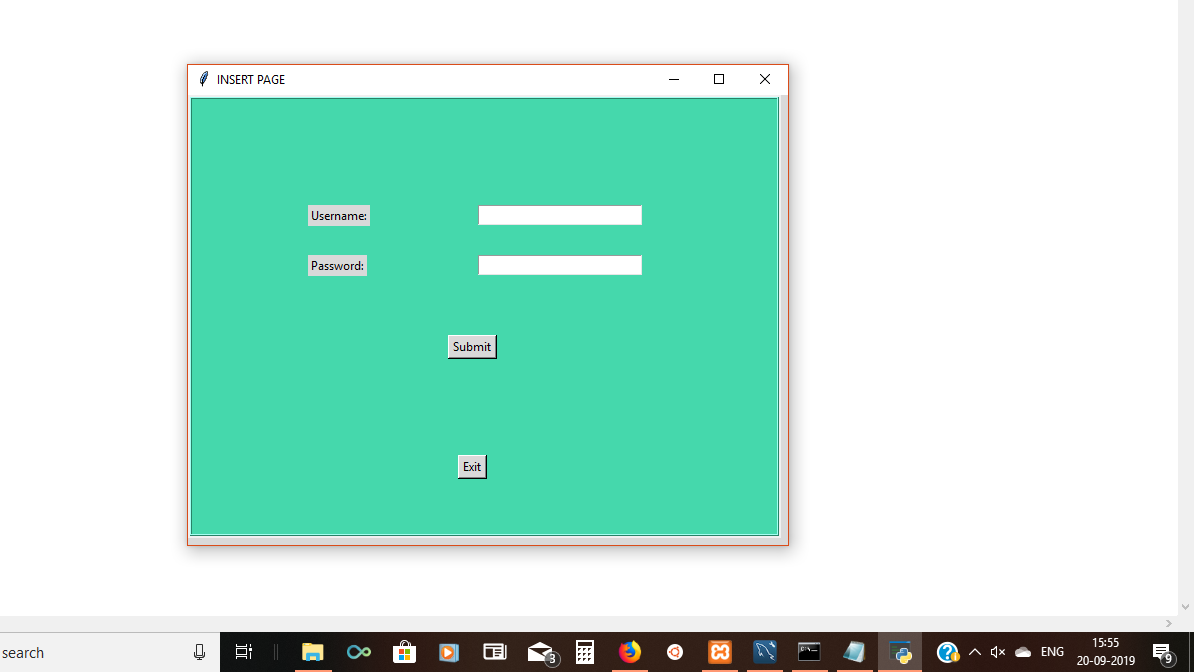
***Lpg insert page***

******

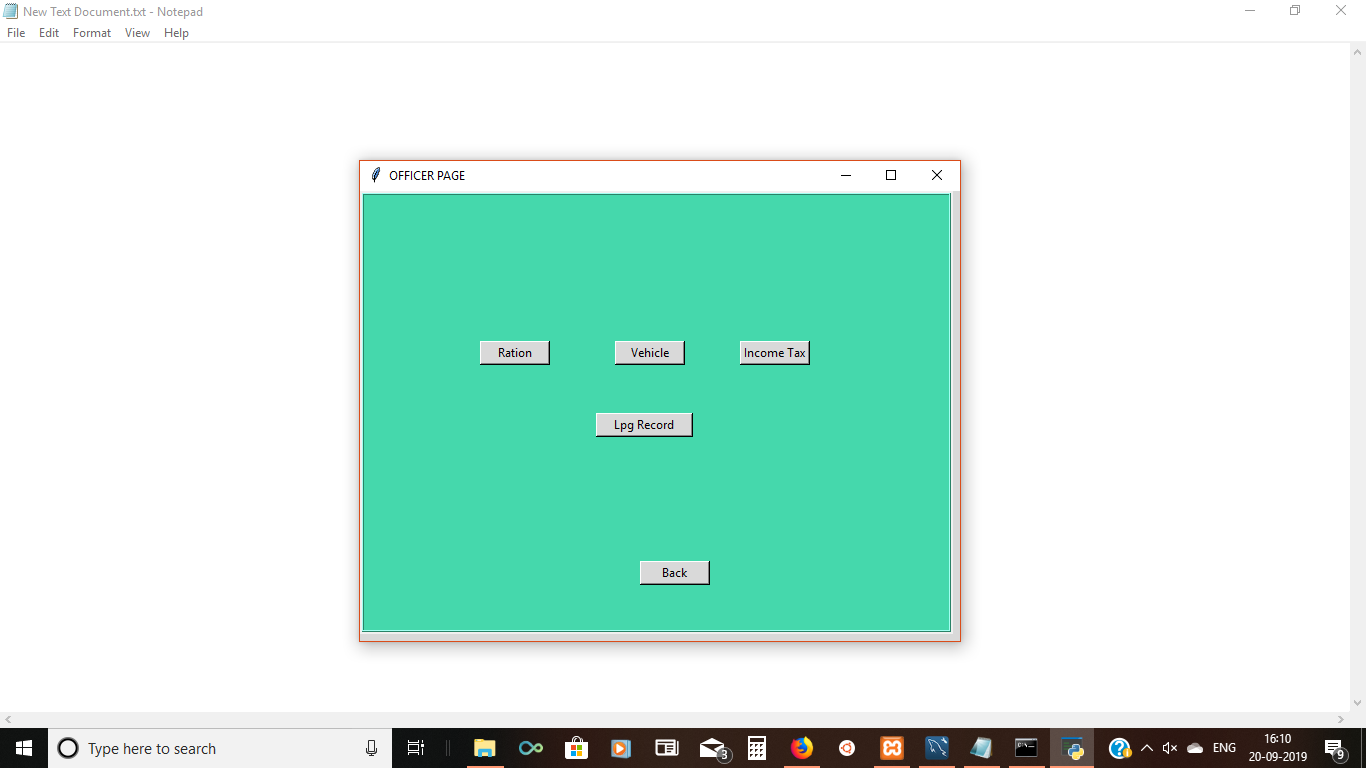
***Health card insert page***

******

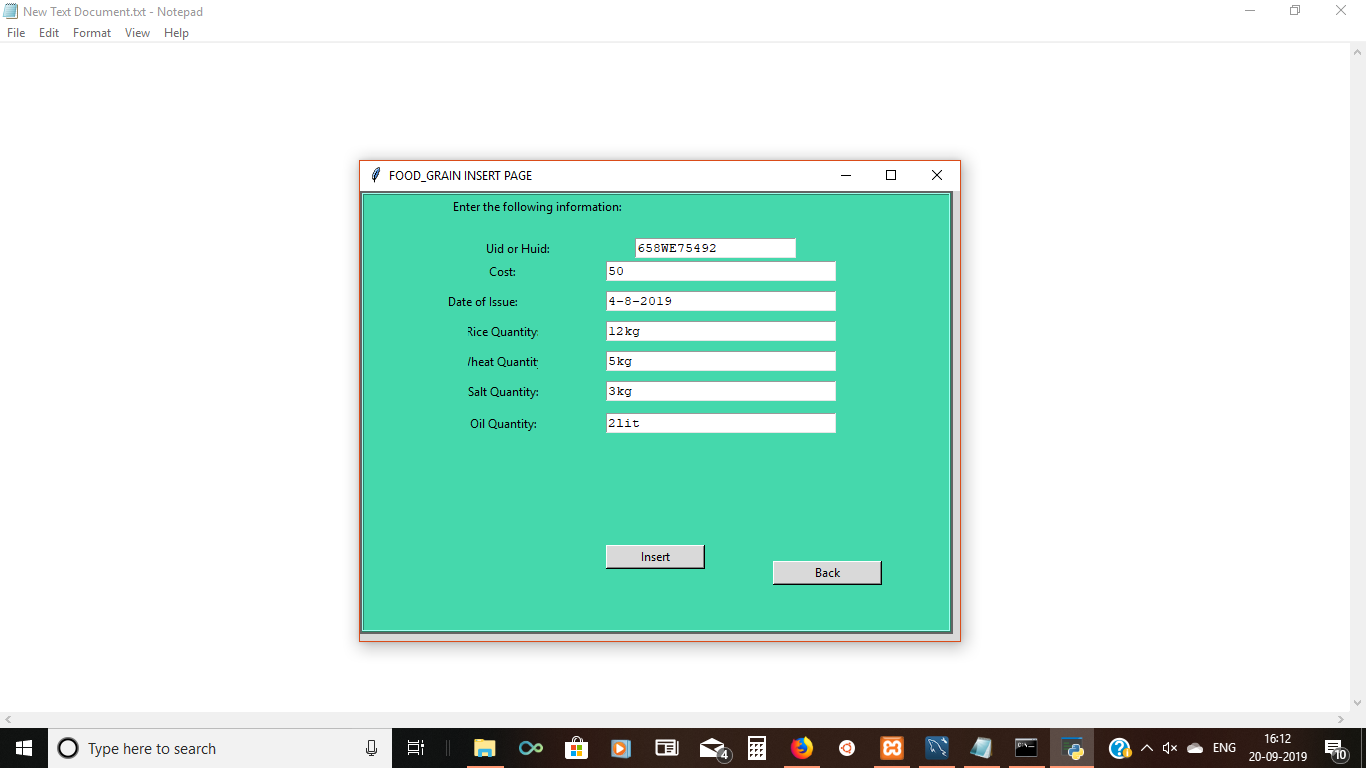
***Officer login page***

******

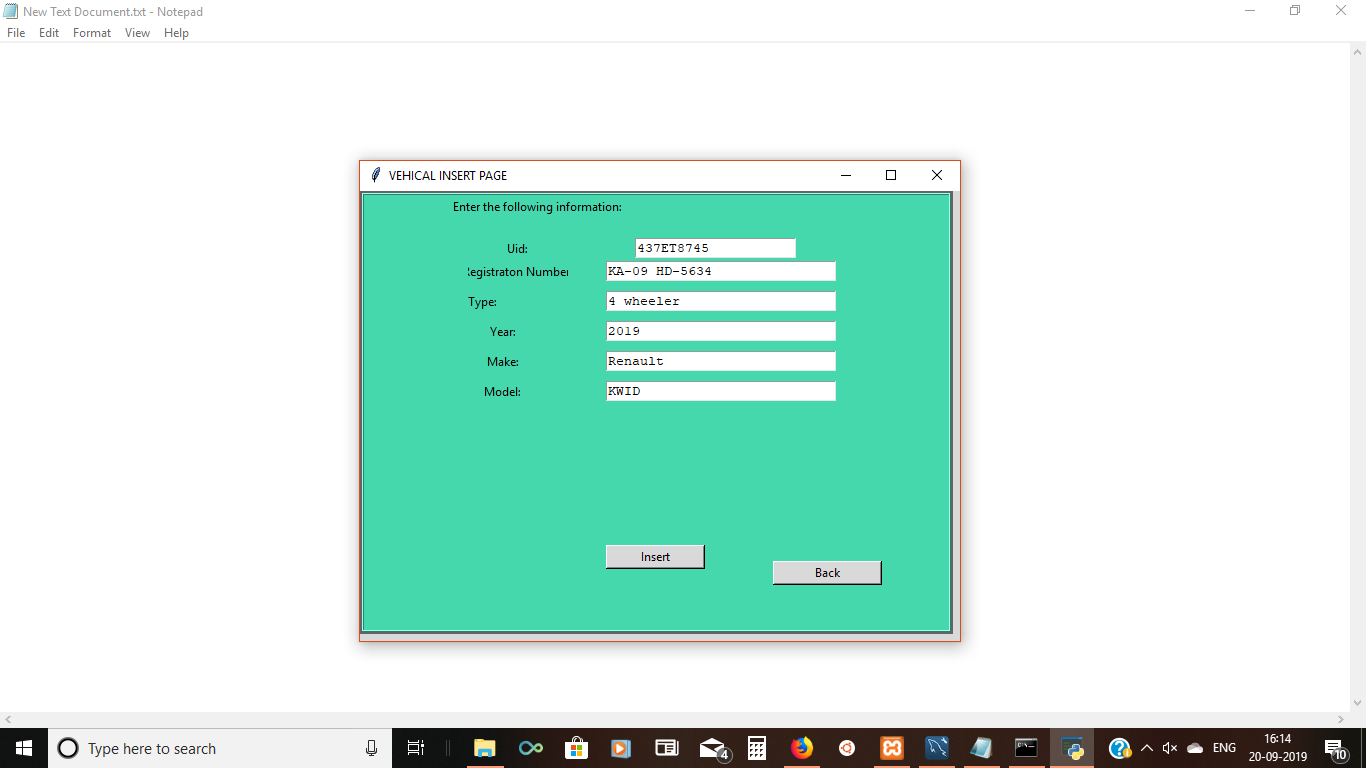
***Officer update page***

******

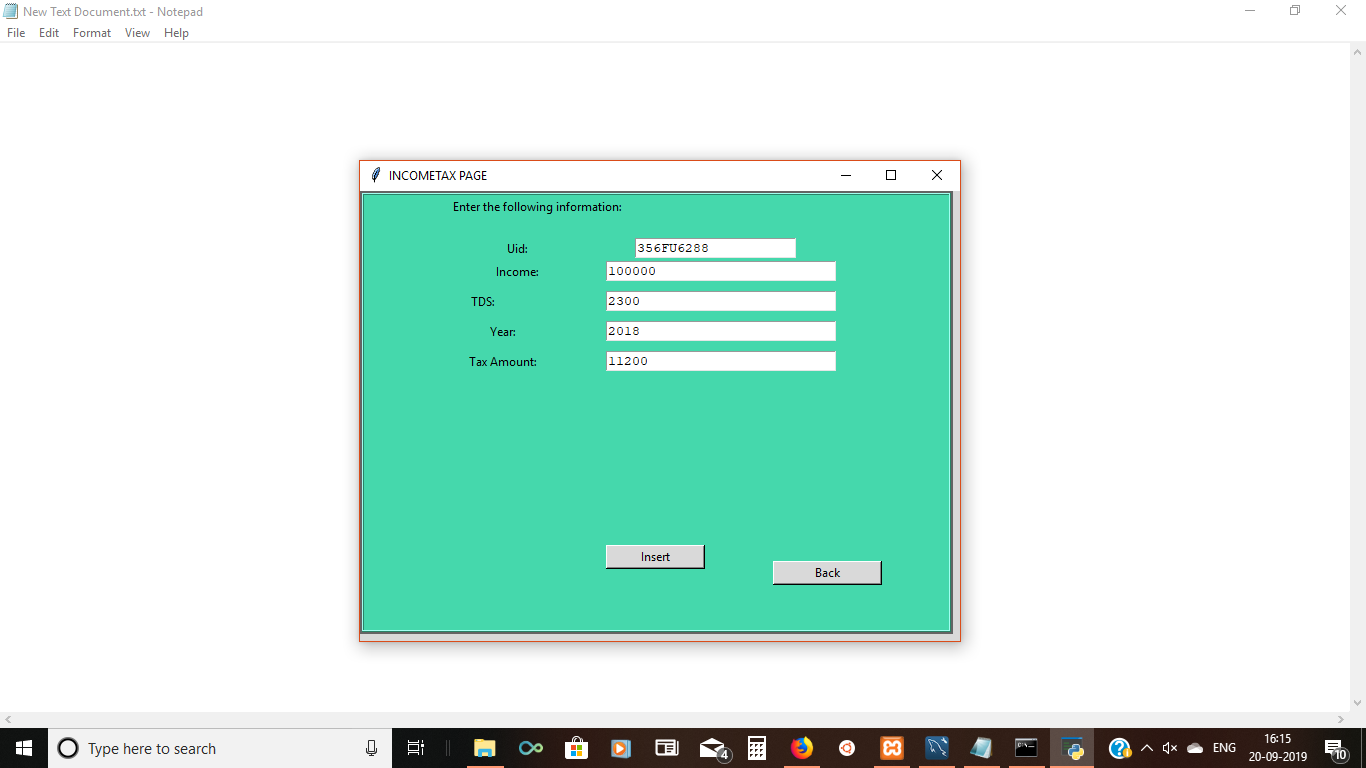
***Officer Food grains update page***

******

***Officer Vehicle update page***

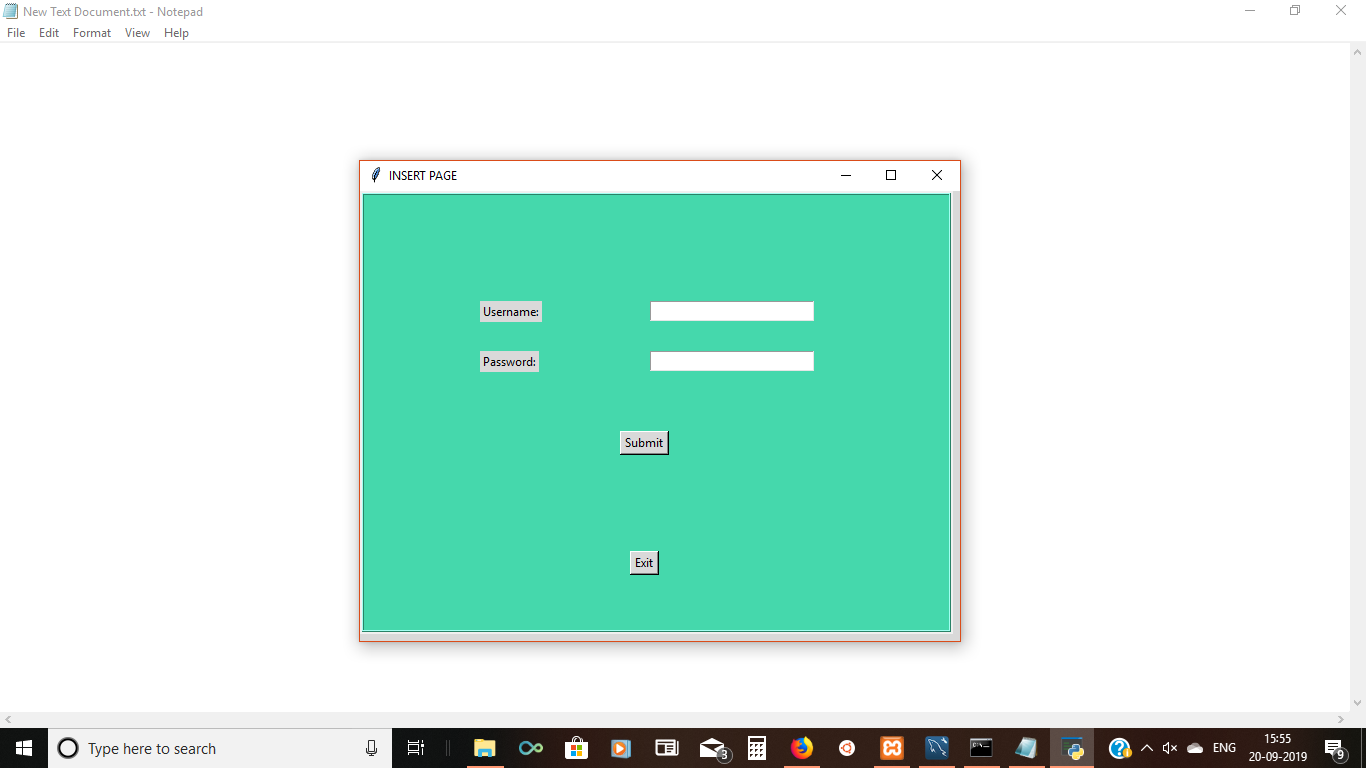
******

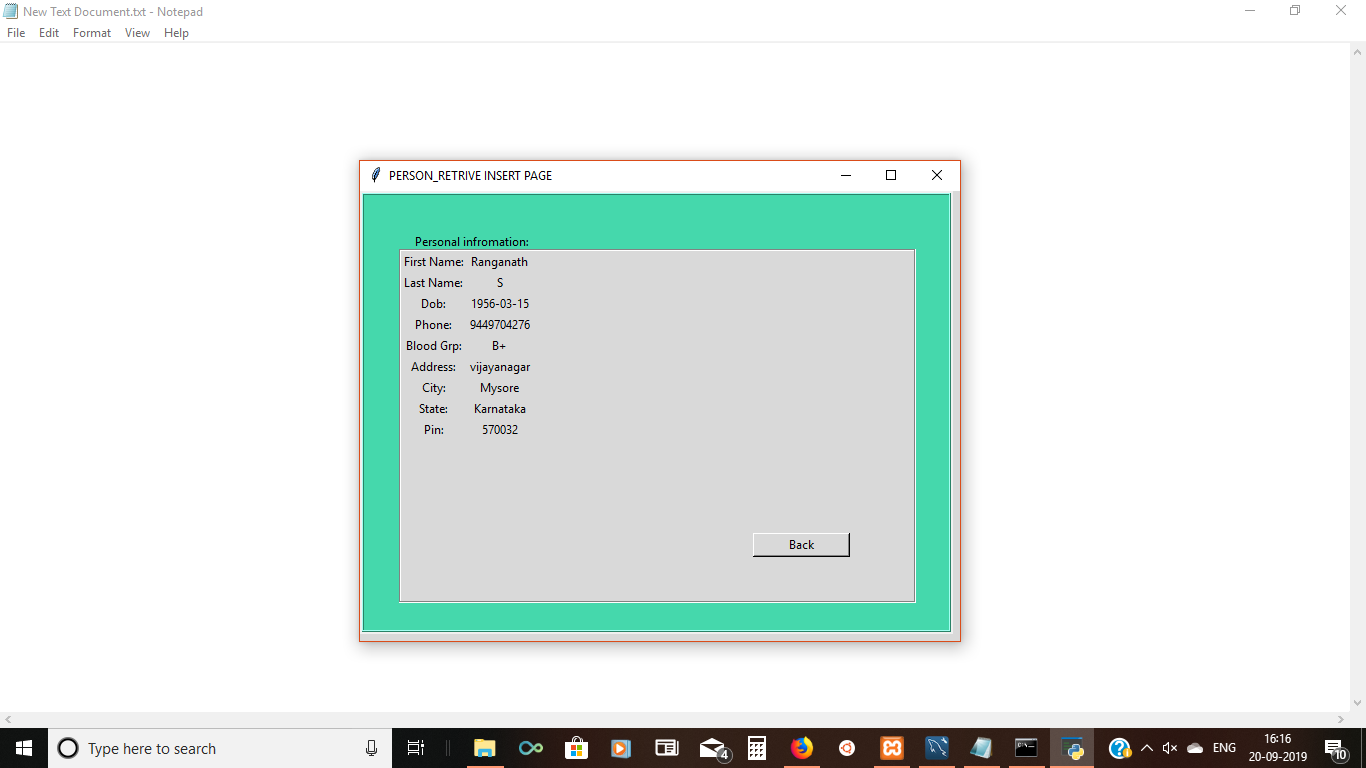
***Officer Income Tax update page***

******

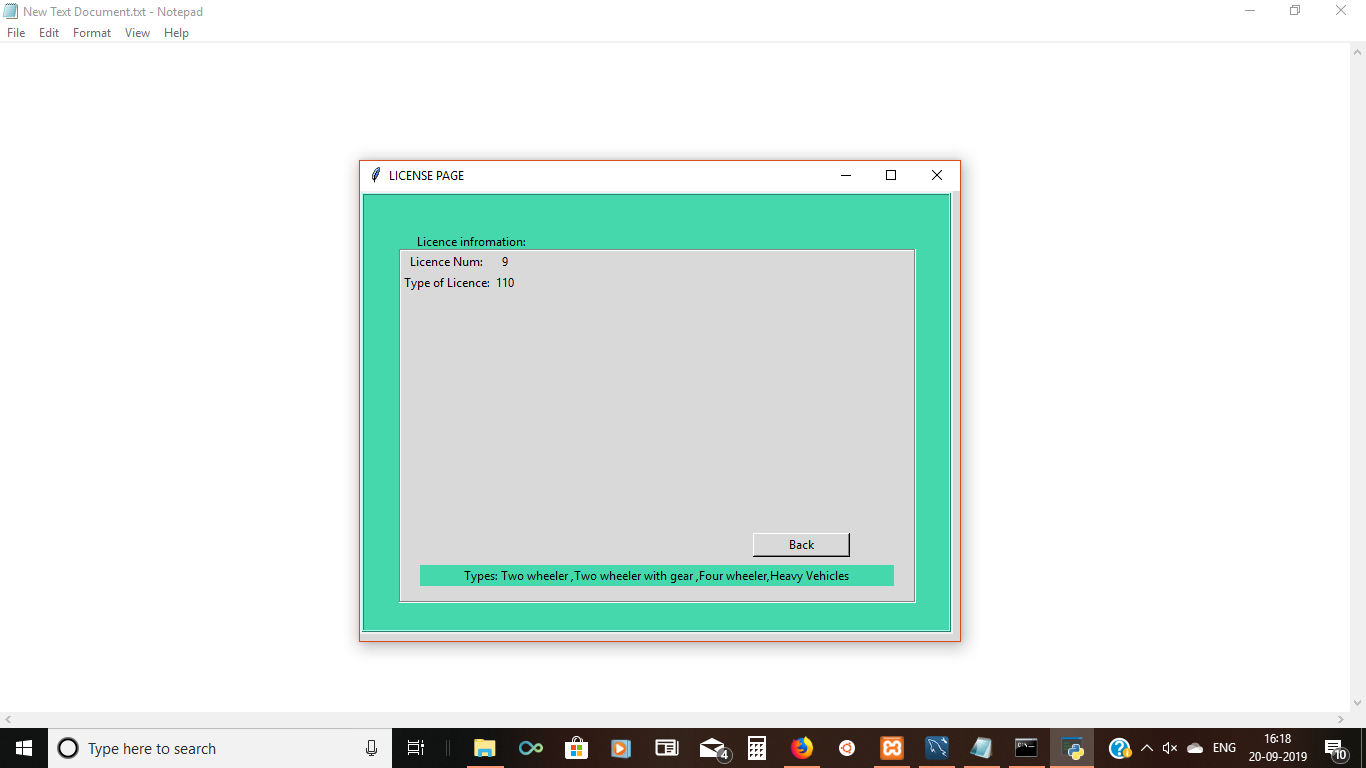
***Officer LPG Record Update page***

***User login page***

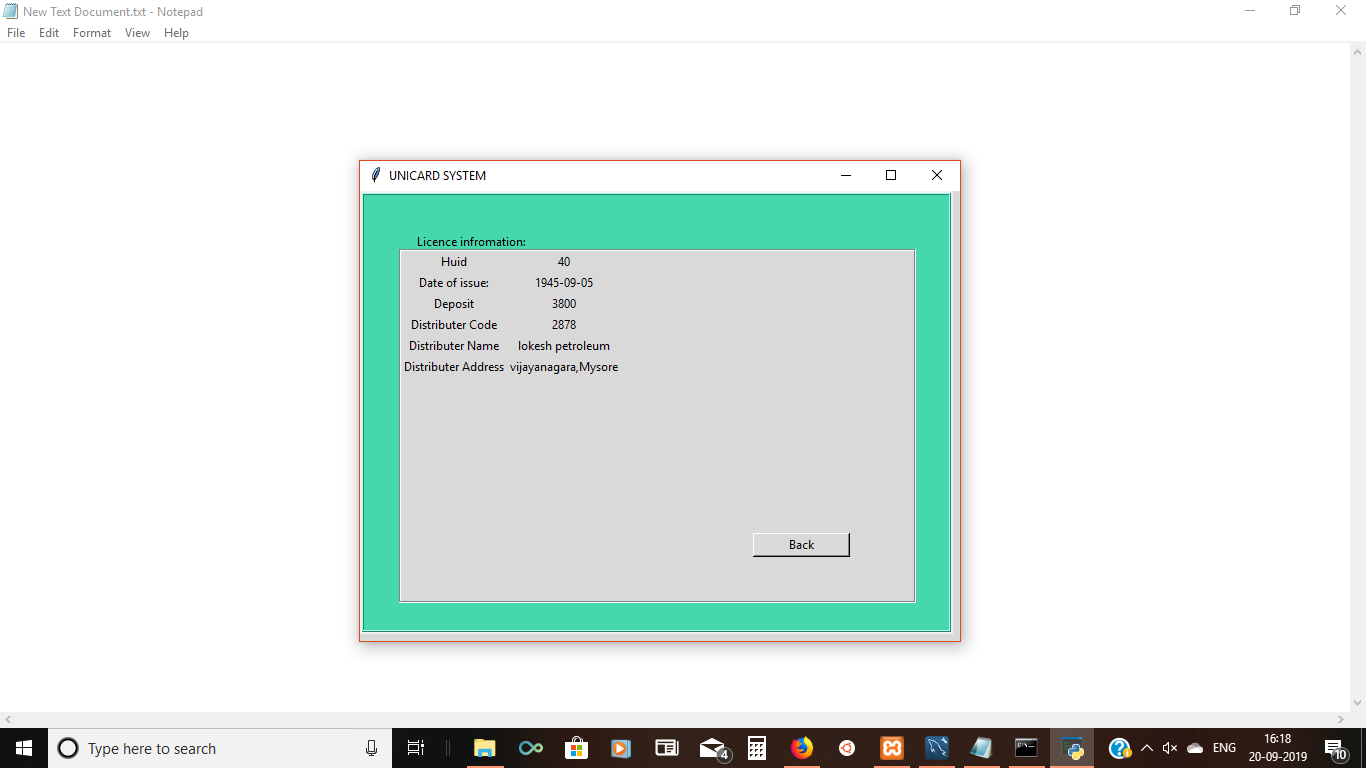
******

***Person information page ***

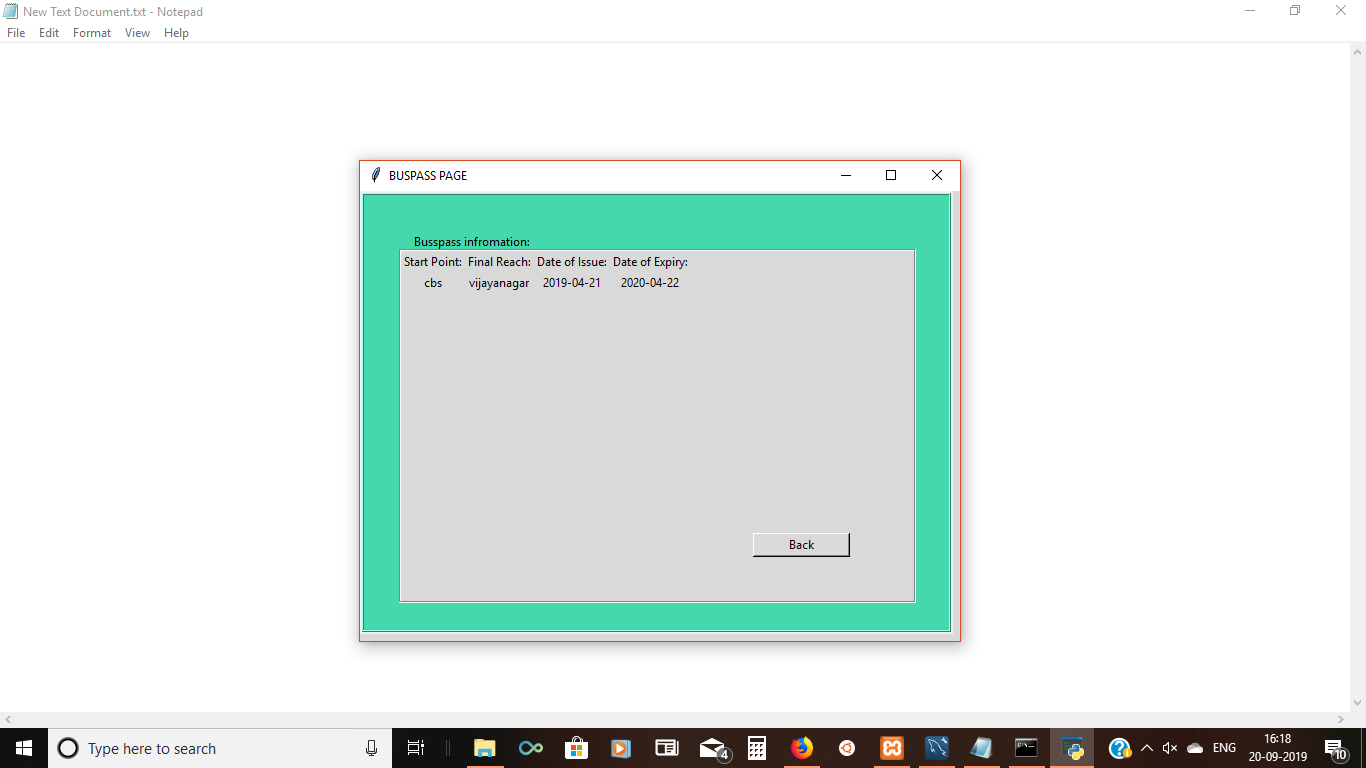
***Person License information page***

******

***Person LPG information page***

******

***Person Bus pass information retrieve page***



**Conclusion**

By using this Documents linking and Management system :

* The rate of corruption will reduce.
* Implemented to reduce the manul work and save storage space.
* Prevents documets loss.
* finds documents quickly so, that it saves time.
* Track volumes of transactions through the effective use of mapping technology.
* Identifying and eliminating the middlemen (misuse).
* Will avoid duplication of Documents.