Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

DONE BY

NAYANI NAMRATHA – RA1911028010143

GOWTHAMI PRIYA – RA1911028010140

BHARATH REDDY – RA1911028010130

PROBLEM STATEMENTS

Aim: To build an ONLINE BUS RESERVATION SYSTEM software.

Project Description: Currently, the type of system being used at the counter is an internal system which is manually used in selling the bus tickets. The problems facing by the companies are that customers have to go to the counter to buy bus tickets or ask for bus schedule, customers will also have to queue up for a long time in order to secure a bus ticket and will also need to pay cash when they buy the bus ticket.

Objectives: The main purpose of this study is to automate the manual procedures of reserving a bus ticket for any journey. This system is said to be an automatic system and customers can select seats by themselves. Specifically, objectives of this project will consist of:

i) Providing a web-based bus ticket reservation function where a customer can buy bus ticket through the online system without a need to queue up at the counter to purchase a bus ticket.

ii) Enabling customers to check the availability and types of busses online. Customer can check the time of arrival & departure for every bus through the system.

iii) Easing bus ticket payment by obtaining a bank pin after payments is made to the various designated banks.

iv) Ability of customers to cancel their reservation.

v) Admin user privileges in updating and canceling payment, route and vehicle records.

Problem statement:

Currently, the type of system being used at the counter is an internal system which is manually used in selling the bus tickets. The problems facing by the companies are that customers have to go to the counter to buy bus tickets or ask for bus schedule, customers will also have to queue up for a long time in order to secure a bus ticket and will also need to pay cash when they buy the bus ticket.

BUSINESS CASE STUDY

Business Case:

In this ongoing busy world, everyone will be thinking of going in a smart way of travelling. Booking of tickets in a online mode will be of a great use of travelling in Bus

DETAILED BUSINESS PROBLEM:

Problem statement:

System that are using by the verisity at the counter currently is an analog system and just used to sell the bus ticket at the counter Officials and the student has to go to the counter to buy bus ticket or ask for bus schedule .Further more, customer need to pay cash when they buy the bus ticket and sometimes needs to queue up long time to get the ticket.

Assumptions:

Since application is internet based , the user need a basic knowledge of choosing correct destination and should have the knowledge of making payment. Updating of data may take some time. Recovery of previous data depends on user.

REQUIREMENTS ELICITATION AND USER STORIES

SCOPE:

The system product to be produced is a Bus Booking Management System which will automate the major Bus operations. The first subsystem is a Bus Reservation and as well as advance Booking System to keep tracks of reservations and seats availability. The second subsystem is the Tracking and Selling tickets System that charges the current journey. The third subsystem is a General Management Services and Automated Tasks System which generates reports to audit all bus operations and allows modification of subsystem information. There are two accounts for the BBMS. The end users are the Bus staff (customer service representative) and administrator the Bus managers. Both user types can access the Reservation and Booking System and the ticket Tracking and Selling System. The General Management System will be restricted to management users.

The objective of this Bus Management System is to provide a system that can manage a bus that has increased in size to a total of 60 seats. Without automation the management of the bus has become an unwieldy task. The end users' day-to-day jobs of managing a bus will be simplified by a considerable amount through the automated system. The system will be able to handle many services to take care of all passengers in a quick manner. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction**.**

SPECIFIC REQUIREMENTS

This section contains all the software requirements in a more detailed manner, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies all the requirements.

Functional Requirements

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into three main categories, Reservation/Booking, travel, and Management. For further details, refer to the use cases.

Nonfunctional Requirements

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and as well as portability.

Performance Requirements

Performance requirements define the acceptable response times for system functionality.

- The load time for user interface screens shall take no longer than two seconds.
- The log in information shall be verified within five seconds.
- Queries shall return results within five seconds.

Logical Database Requirements

The logical database requirements include the retention of the following data elements. The list below is not complete but only designed as a starting point for development.

COST AND RISK ANALYSIS

WBS With Project Schedule

Module (#)	Activity (#)	Sub-Task(#)	Assignee(s)	Planned Start Date	Planned End Date	Actual Start Date	Actual End Date	Status
Product Research	Surveys	-	Team members	1st Feb	5th Feb	1st Feb	5th Feb	done
Product Research	Analysis	-	Team members	6th Feb	8th Feb	6th Feb	9th Feb	done
Product Design	Front-end development	Design based on research and documentation	Creatives Team (web-dev team)	11th Feb	15 March	11 Feb	20 March	In progress
Product Design	Design selection	Testing	Design Creatives team lead (web-dev team)	16 March	20 March	21 march	27 march	In progress

Product Development	Initial prototype	Integration of front-end and back-end	Technical team	16 March	10 April	21 March	15 April	In progress
Product Development	Testing	Working analysis	Technical team lead	11 April	20 April	16 April	30 April	In progress
Marketing	Strategy	Plan	Logistics team	16 April	20 April	-	-	Yet to start
Marketing	Collateral	Brochures, Advertising and Commercials	Content and Creatives team	21 April	30 April	-	-	Yet to start
Management	Database Management	Bus and booking data	Database management team	15 April	15 May	-	-	Yet to start
Management	Database Management	User data	Database management team	10 May	20 May	-	-	Yet to start
Management	Final analysis	Bug fixes	Team members	21 May	30 May	-	-	Yet to start

Risk Identification

Financial: this could mean loss of funding, insurance costs, fraud, theft, fees etc.

Physical: this involves physical assets of the organization, personal injuries and environmental.

Ethical or moral: involves a perpetuated, actual or potential harm to the reputation or beliefs of an individual or organization.

Legal: this includes responsibilities and adherence to the law, rules and regulations of governing bodies such as the federal, state or local governments

List (Describe) Register

Risk ID (#)	Risk Description	Impact Description
R01	Software failure or Database/System Crash	May result in loss of data and difficulty in booking for the customers.
R02	Virus Attacks	Risk of losing user data and corruption of system files.
R03	Poor resource management and documentation	Unnecessary Time waste and causes chaos. Creates a bad work environment and demotivates stakeholders.
R04	Failure to deliver the end product on time	Creates a sense of dissatisfaction among all clients and stakeholders and also a bad work environment.
R05	Constantly altering project requirements and priorities	Leads to not having a proper objective, randomly abandoning certain tasks and causes sudden change in schedules.

Managing Risk

Risk ID (#)	Status [Open / Closed]	Risk Appetite [Accept/ Mitigate/ Transfer/ Avoid]	Action	Action Owner	Target Date	Remarks
R01	Open	Averse	Actions taken during normal operation to ensure system can recover from failure	Front-end/Back-end Developer	1 -5 Feb	Database Recovery processes vary depending on the type of failure that occurred, the structures affected.
R02	Open	Averse	Actions taken after a virus attack to restore database to consistent state	Front-end/Back-end Developer	-	To counter hackers' damaging influence, administrators use various software and hardware solutions that should detect, avert,register, and deflect their attacks.
R03	Closed	Flexible	Hold scheduling workshops with the project team so they understand the plan and reduce the likelihood of missed tasks	Project Lead or UI/UX Designer	6 -9 Feb	Share the plan and go through upcoming tasks at each weekly project progress meeting
R04	Open	Cautious	Must be prepared to hit an unexpected roadblock along the way. Must have a plan B in case plan A falls through.	Project Lead or UI/UX Designer	11th Feb to 20th March	Prevent lack of communication, poor planning or a lack of discipline
R05	Closed	Cautious	Ask all your stakeholders to	Tester/Quality		List all the requirements you

	.		review the list and confirm that it presents the current view of what they want the project to deliver	Assurance Lead	6 -9 Feb	currently have on the project and make sure they all tie back to the project's objectives.

PROJECT PLANNING

Project Management Plan

Key issues driving the project-

- Lack of well-designed Online Bus Reservation system
- Huge consumer base dissatisfied with the current state
- Lack of competitors
- Help improve with the current situation

Focus Area	Details
Integration Management	Our Project integration management touches all five phases of a project. This ensures smoothly run and integrated project processes
Scope Management	helps to book our bus tickets online we will design, develop, test and implement our project plan pitch our project try to get more sponsorship
Schedule Management	we scheduled our project in order to complete it within the due date
Cost Management	Estimate Effort (hr) =200 Assign Team - 3 members have control over budget
Quality Management	we look into the time complexity of the result to get it within seconds and we assure its availability 24/7
Resource Management	People: People & Skills Required

	Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Roles and responsibility are decided by the team
Communication Management	conference calls , video conference and texting for communication
Risk Management	Planning is done by prioritizing project risks
Procurement Management	modify as per the user choice

Estimation

Effort and Cost Estimation

WBS	Activity	Activity Description	Sub-Task Description	Effort (in hours)	Cost in INR
E1FR1	E1R1A1	Design the user interface	Confirm the user requirements (acceptance criteria)	3	600
			Design prototype	6	1200
			Consult customers and consumer	9	1800
			Document all UI changes	1	200
E1FR2	E1R2A1	Design and develop Data Analytics Module	Develop Demand Forecasting feature	8	1600
			Testing	4	800
E1FR3	E1R3A1	Payment methods	To develop the payment methods	4	800
			Testing	2	400
E1FR4	E1R4A1	Design and develop search and database module	Develop search functionality	5	1000
			Testing	3	600

E1FR5	E1R5A1	Integrate modules	Integration	9	1800
			Testing	3	600
E1FR6	E1R6A1	Marketing	Marketing the product	3	600
E1FR7	E1R7A1	Communication	Weekly report for shareholders	1	200
			Internal Meetings	8	200
E1FR8	E1R8A1	Planning	Project Planning	3	600
			Risk Assessment	2	400

Effort (hr)	**Cost (INR)**
1	200

Infrastructure/Resource Cost [CapEx]

Infrastructure Requirement	**Qty**	**Cost per qty**	**Cost per item**
Computers(Development Machines)	3	9000	3000
IDE plugins, etc	11	7000	500-900
Configuration Management System	1	2000	2000

Maintenance and Support Cost [OpEx]

Category	**Details**	**Qty**	**Cost per qty per annum**	**Cost per item**
People	Network, System, Middleware and DB admin Developer , Support Consultant	3	2000	6,000
License	Operating System Database Middleware	10	10000	100,000

	IDE			
Infrastructures	Server, Storage and Network	20	20000	400,000

Project Team Formation

Identification Team members

Name	**Role**	**Responsibilities**
XYZ	Key Business User (Product Owner)	Provide clear business and user requirements
Namratha	Project Manager	Manage the project
Bharath	Business Analyst	Discuss and Document Requirements
Namratha	Technical Lead	Design the end-to-end architecture
Namratha	UX Designer	Design the user experience
Bharath	Frontend Developer	Develop user interface
Gowthami	Backend Developer	Design, Develop and Unit Test Services/API/DB
Namratha	Cloud Architect	Design the cost effective, highly available and scalable architecture
Gowthami	Cloud Operations	Provision required Services
Gowthami	Tester	Define Test Cases and Perform Testing

Responsibility Assignment Matrix

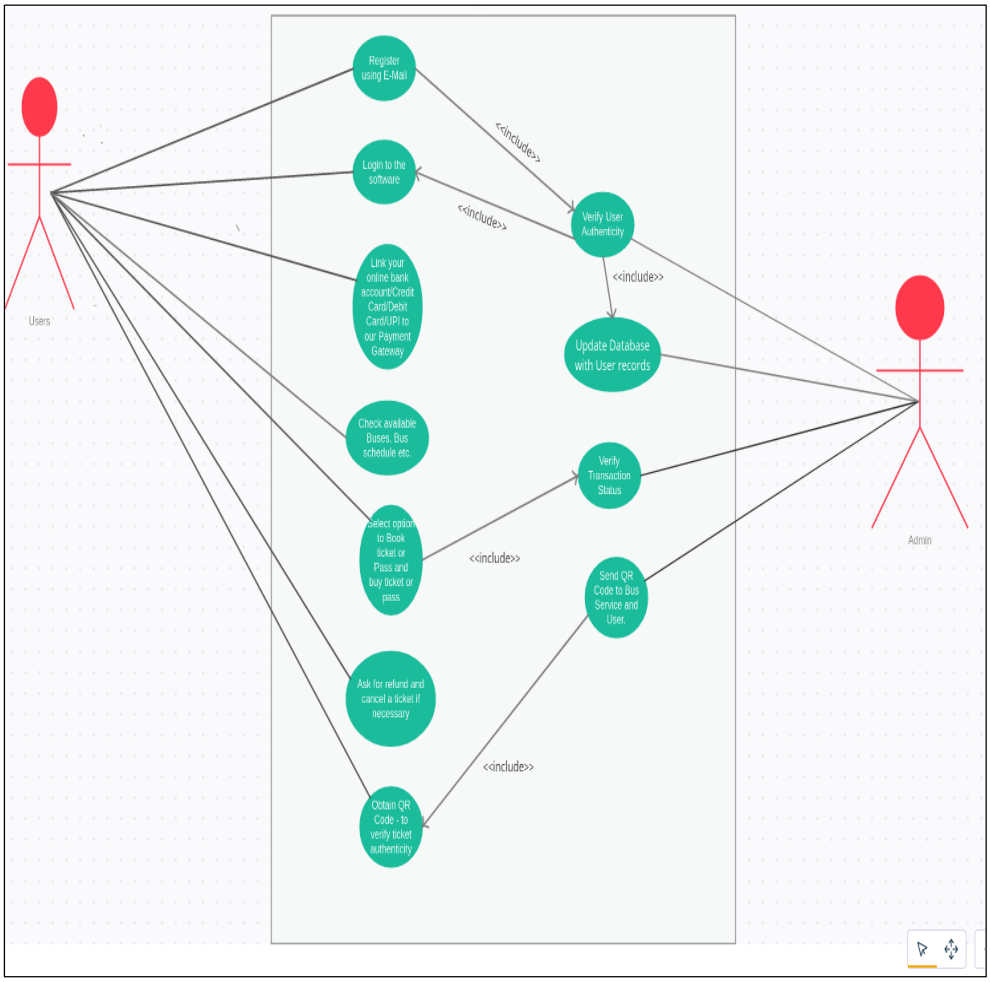
RACI Matrix	**Team Members**			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User**
User Requirement Documentation	A	C/I	I	R
Barcode scanning module		A/R	I	C/I
Data Analytics module		A/R	I	C/I
Integration		A/R	A/R/I/C	C/I

Debugging		A/R	I	C/I

A	Accountable
R	Responsible
C	Consult
I	Inform

SYSTEM DIAGRAMS

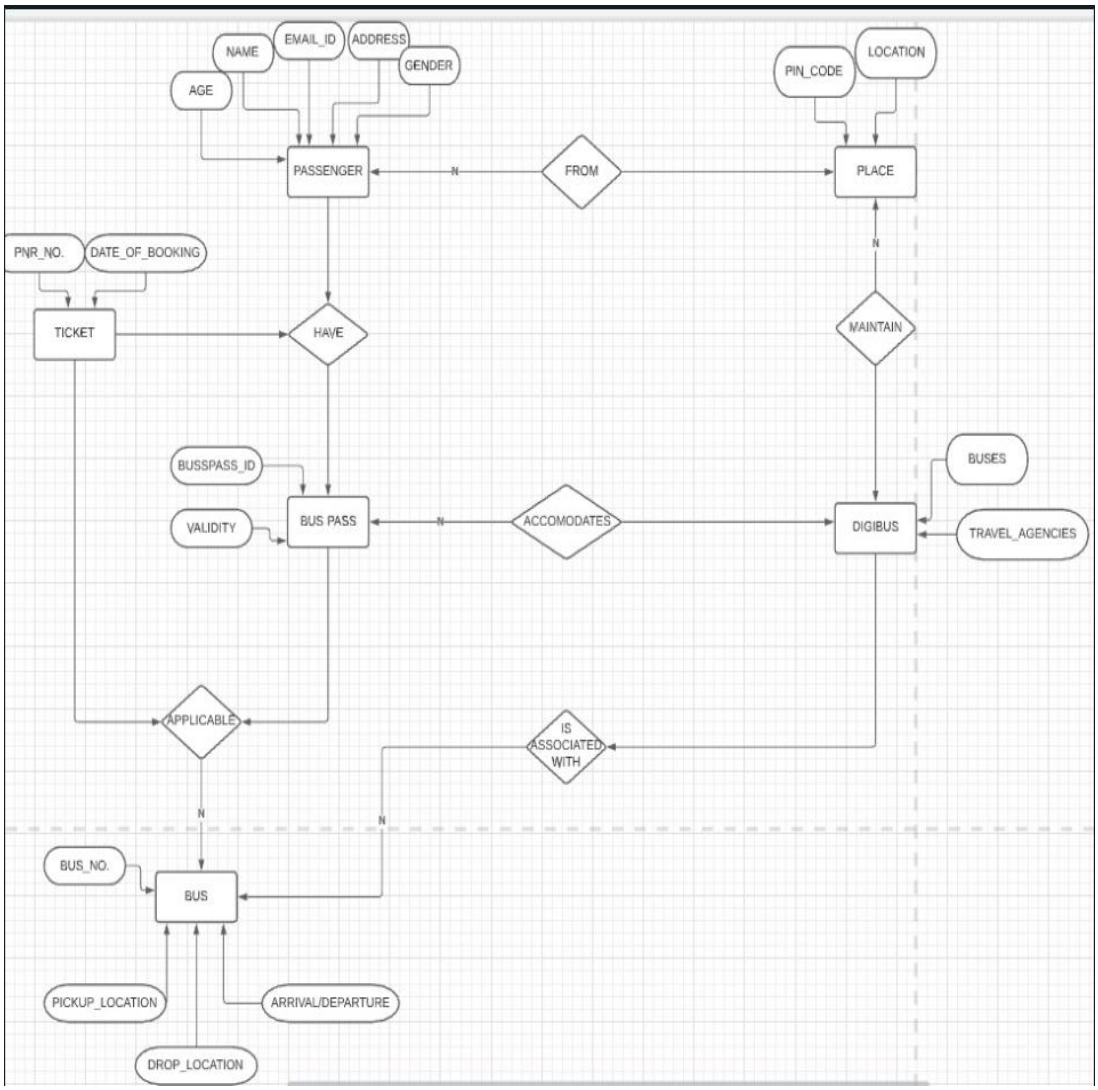
Architecture Diagram with Description:



Description of the Use Case Diagram:

- The user will first create an account and register using his/her email ID.
- Then the system administrator will check for the authenticity of the account/email and will add the user and their details to the database.
- The user can then login.
- After logging in the user will then link his online account/credit card/debit card/ UPI to either our payment gateway or E-Wallet to help in the process of seamless transactions while tickets/passes are being issued. Admin will then update the database with the user credentials.
- The user then can look at the available tickets/passes to book. He/She can also view the bus schedule.
- Once the user books their ticket, they will be sent a QR Code which will be used to check the authenticity of the ticket. The user can also cancel their pass/ticket if necessary.
- In this manner the user can seamlessly buy bus passes/tickets whenever necessary.

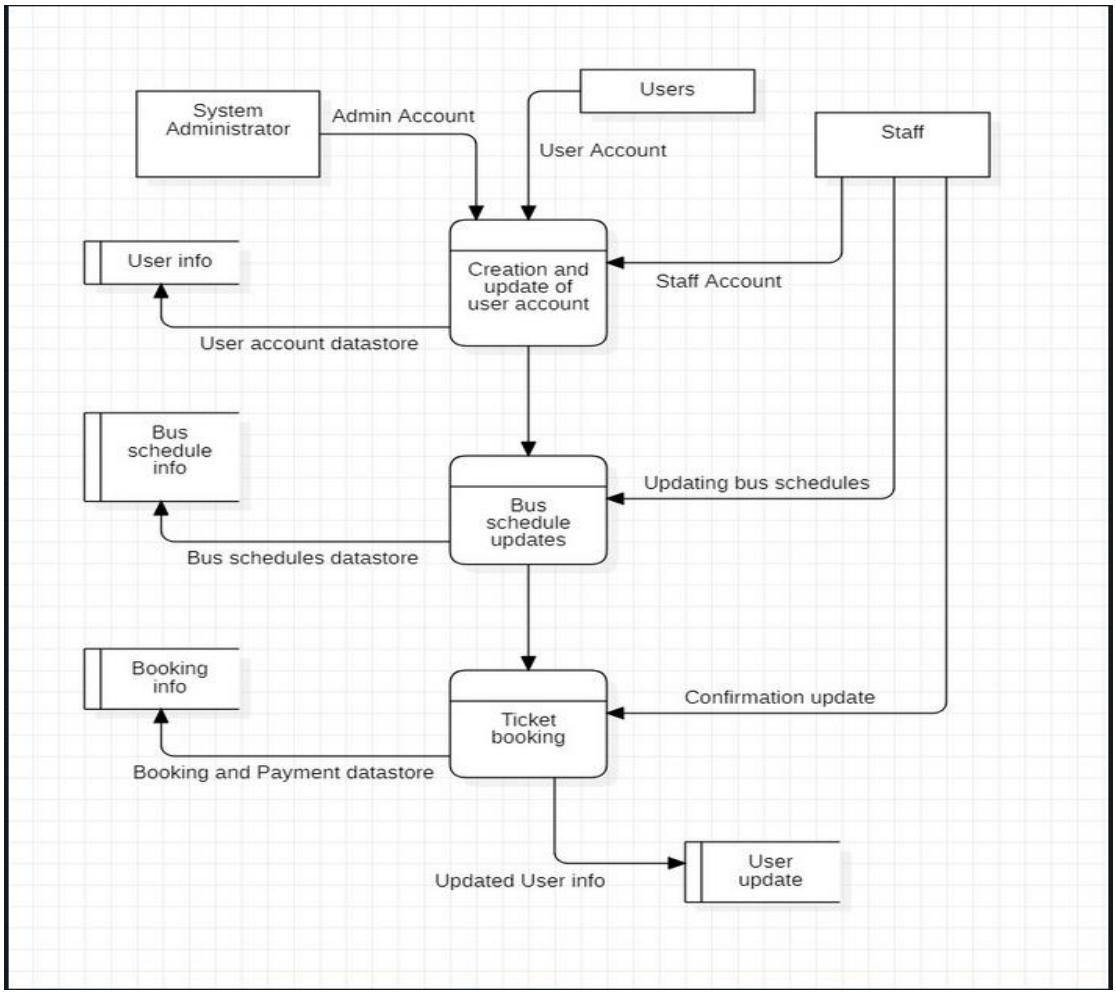
- **ER Diagram**


NAME
EMAIL_ID
ADDRESS
GENDER
AGE
PASSENGER
N
FROM
PLACE
PIN_CODE
LOCATION
N
PNR_NO.
DATE_OF_BOOKING
TICKET
HAVE
MAINTAIN
BUSSPASS_ID
BUSES
VALIDITY
BUS PASS
N
ACCOMODATES
DIGIBUS
TRAVEL_AGENCIES
APPLICABLE
IS ASSOCIATED WITH
N
N
BUS_NO.
BUS
PICKUP_LOCATION
ARRIVAL/DEPARTURE
DROP_LOCATION

➢ Description of the ER Diagram

- For the booking of a Bus pass or ticket the user must enter the details like name, age email Id, gender, Address provided by the DigiBus website
- Next, the user should enter their location details.
- From here the user can book their respective tickets or passes from the available options.
- If the user has booked a ticket he/she will be provided with a unique PNR number and booking details.
- If the user applies for a bus pass he/she will be issued a pass which carry the bus pass Id and the validity.
- They will be provided with additional information regarding bus number, pickup location, drop location, arrival/departure timings through which the passenger can plan things accordingly.
- Now the user can use the ticket/Bus pass to board their respective buses.

- **Data Flow Diagram**

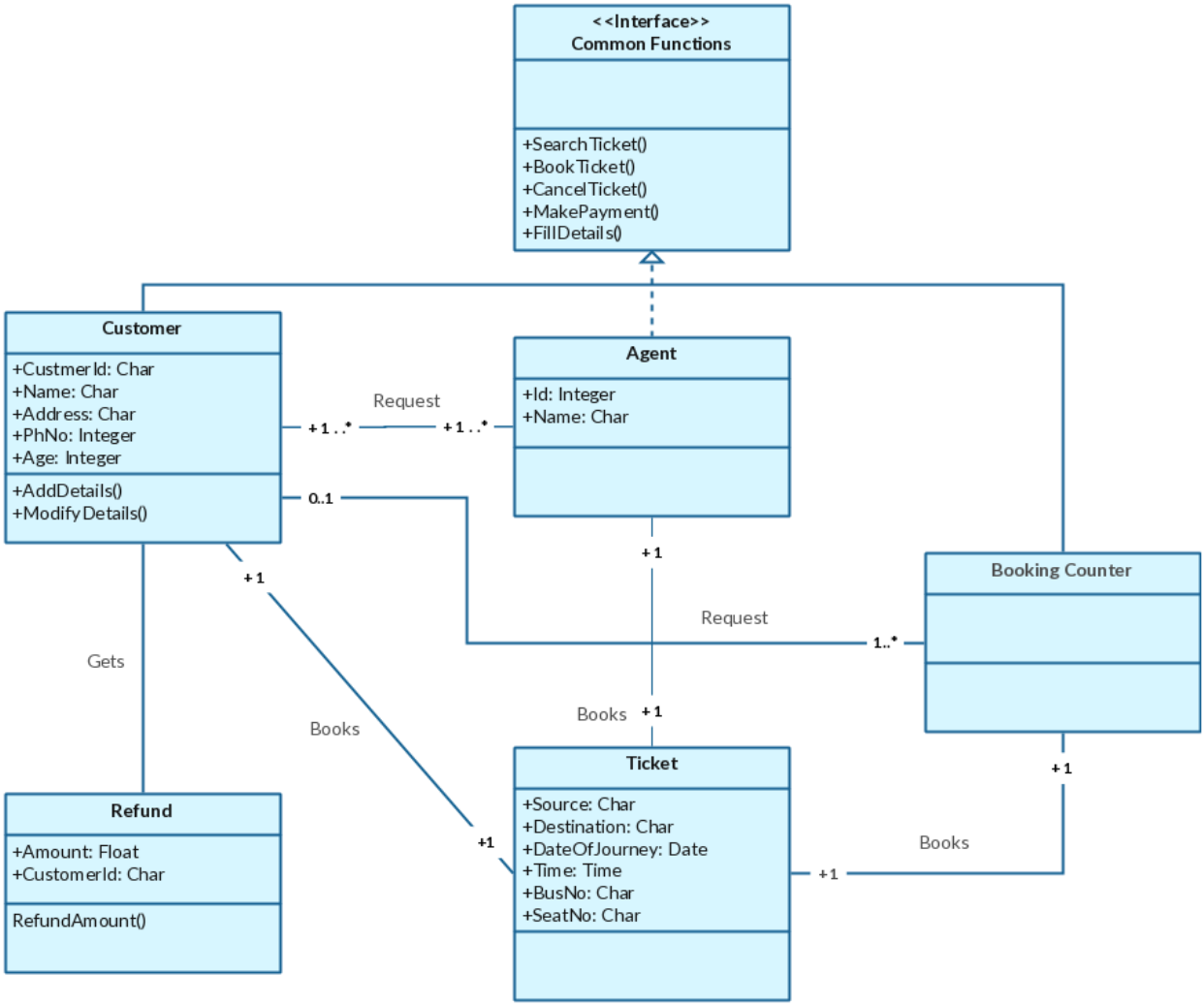


- **Description of the Data Flow Diagram**
  - There are three external entities in data flow diagram:
    - **Administrator** who has access to all the processes, who administrates all the work in website.
    - **Staff** has access to create and update their accounts, update bus schedules and to monitor and operate ticket booking and confirmation.
    - **Users** has access to create and update accounts, check bus schedules and to book tickets or pass.
  - Data regarding accounts that are created or updated, the information is stored in User info Data Store.
  - Record regarding bus schedules is stored in Bus schedule info Data Store and data regarding bookings and payments is stored in Booking info Data Store.

DESIGN STATE

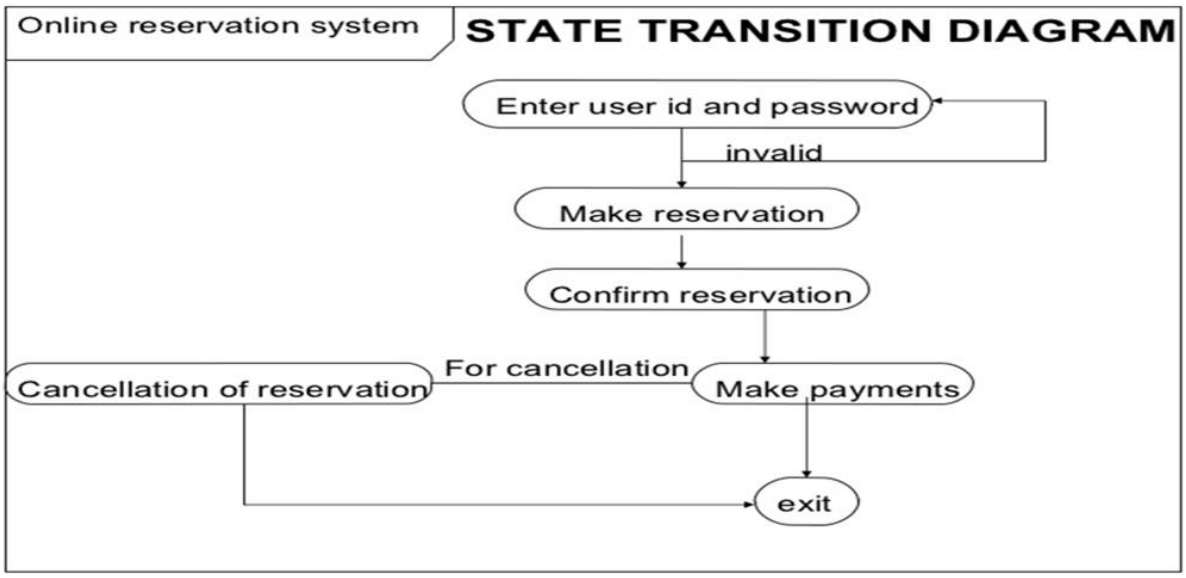
Architecture Diagram :-

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.



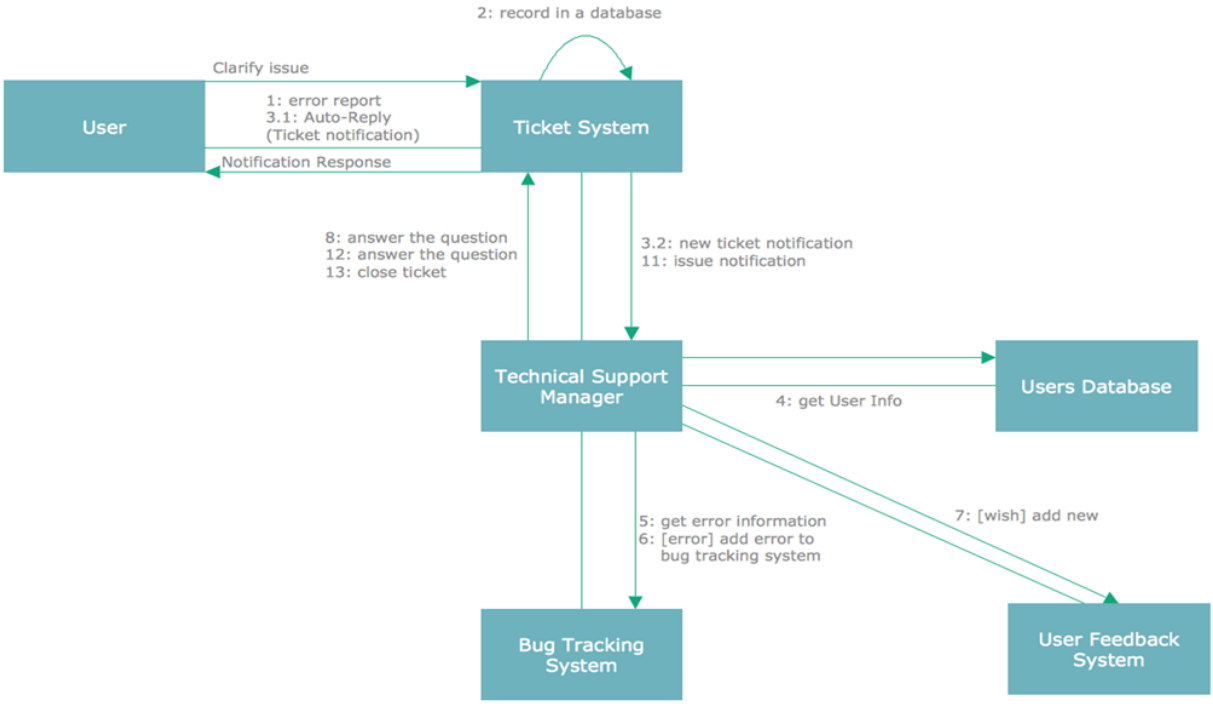
State Diagram:-

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented by a series of events that can occur in one or more possible states. Here each diagram usually represents objects of a single class and track the different states of its objects through the system



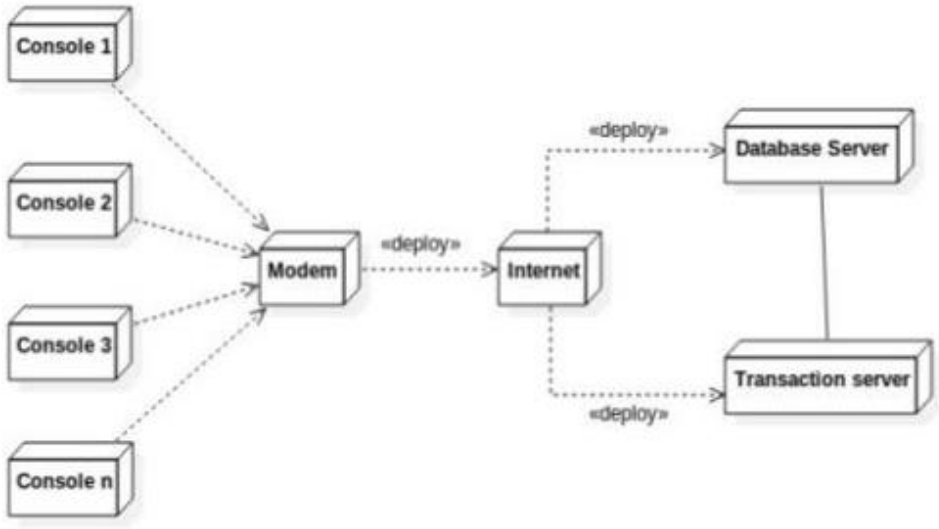
Collaboration Diagram:-

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.



Deployment Diagram:-

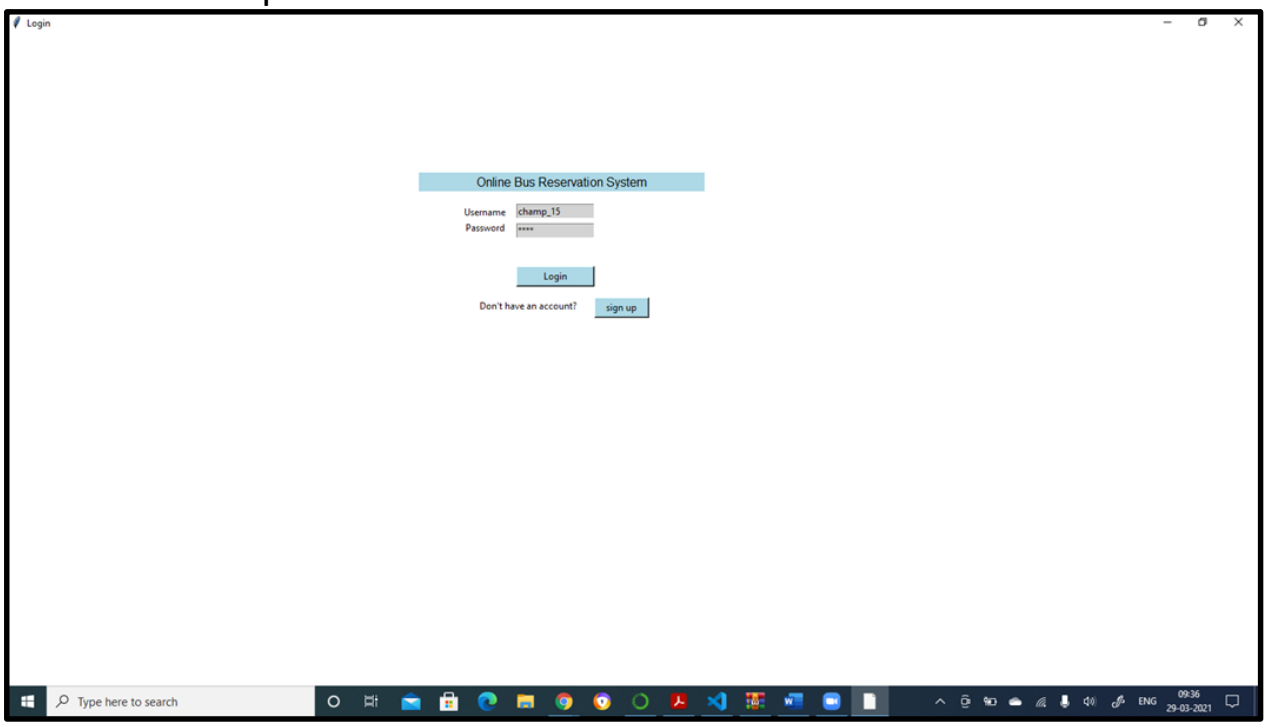
UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They capture the hardware that will be used to implement the system and the links between different items of hardware. It is a model of physical hardware elements and the communication paths between them



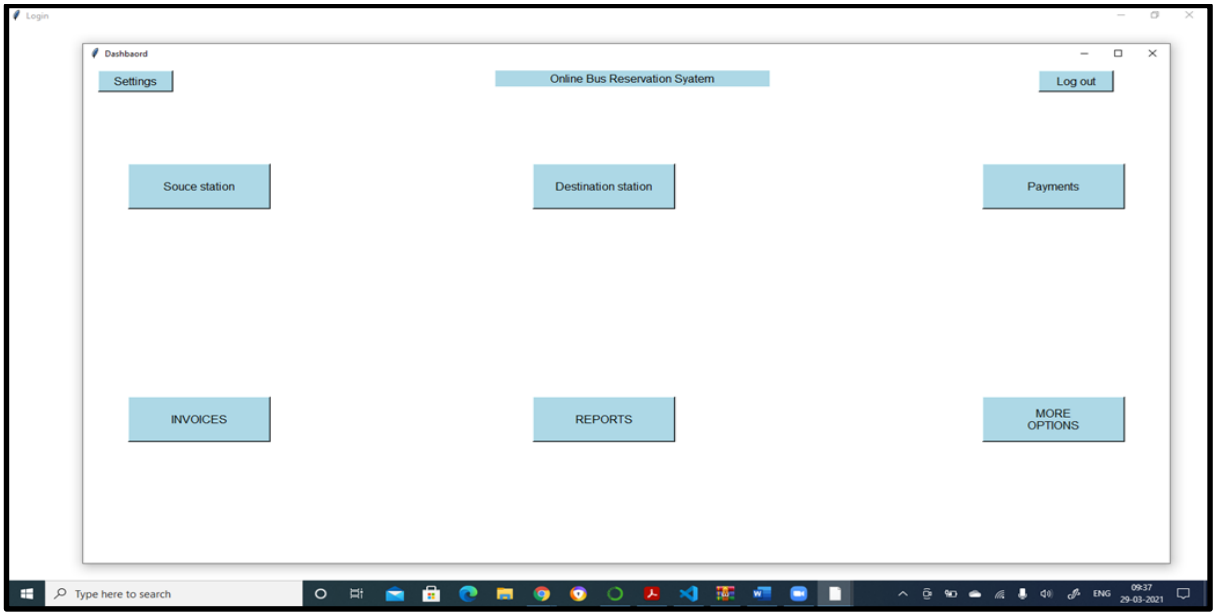
Front end design:-

The frontend of a software program or website is everything with which the user interacts. Websites must work well on multiple devices and screen sizes, which is why modern web development typically involves responsive design

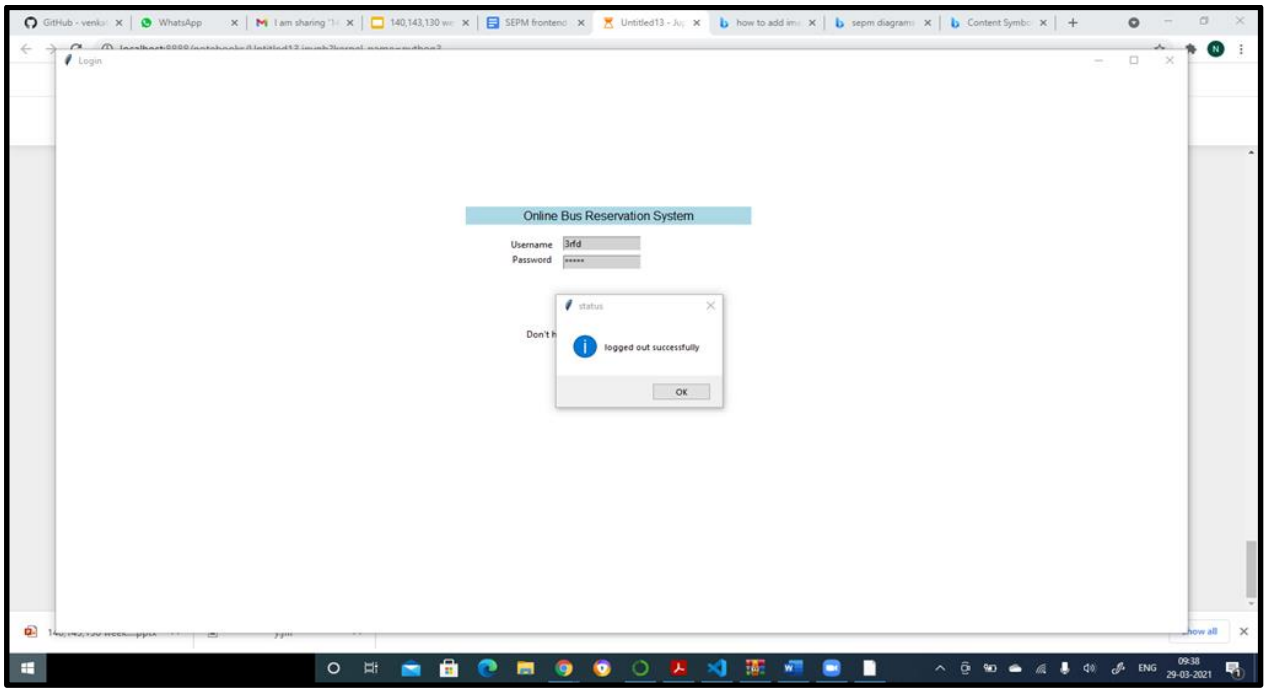
This is the login page ,where we enter the basic details such as username and password.



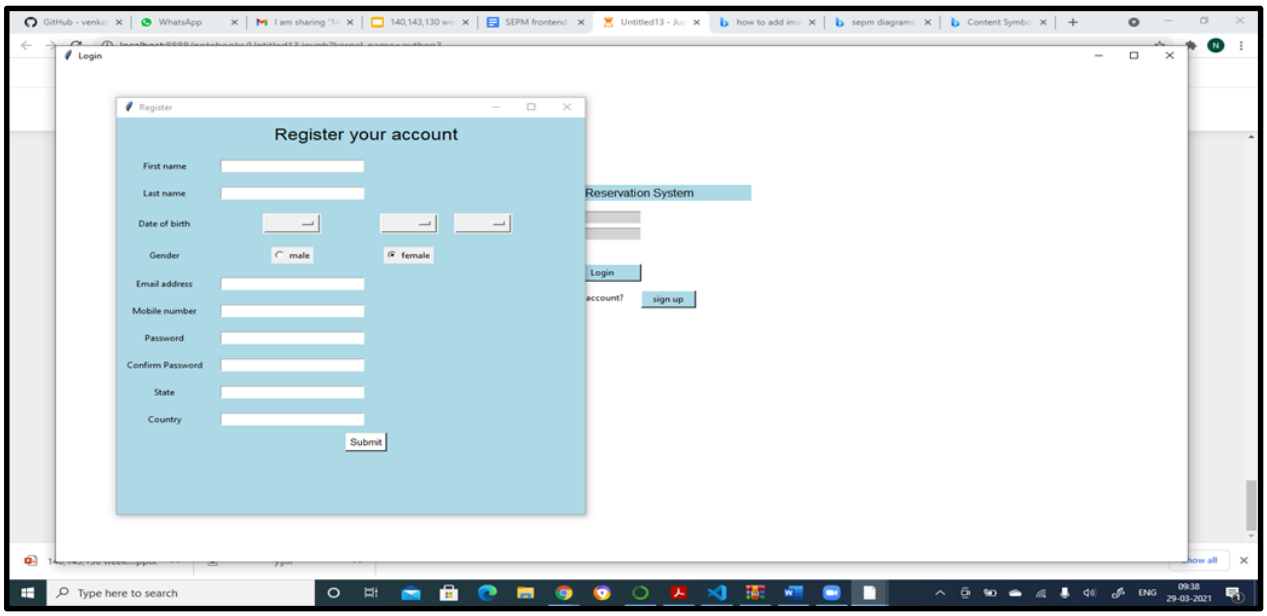
After successfully login,u will enter into one more page where u have to do select the source and destination stations and have to do the payment to get the ticket.



After doing payment,u have to click the logout button,then a message will be displayed as u have logout successfully



If u don’t have an account ,u have to click the signup option and u have to register by filling the basic personal details such as name,password,mobile number,state,password….etc,then u can successfully login.



CODING MODULE – 1

Code:-

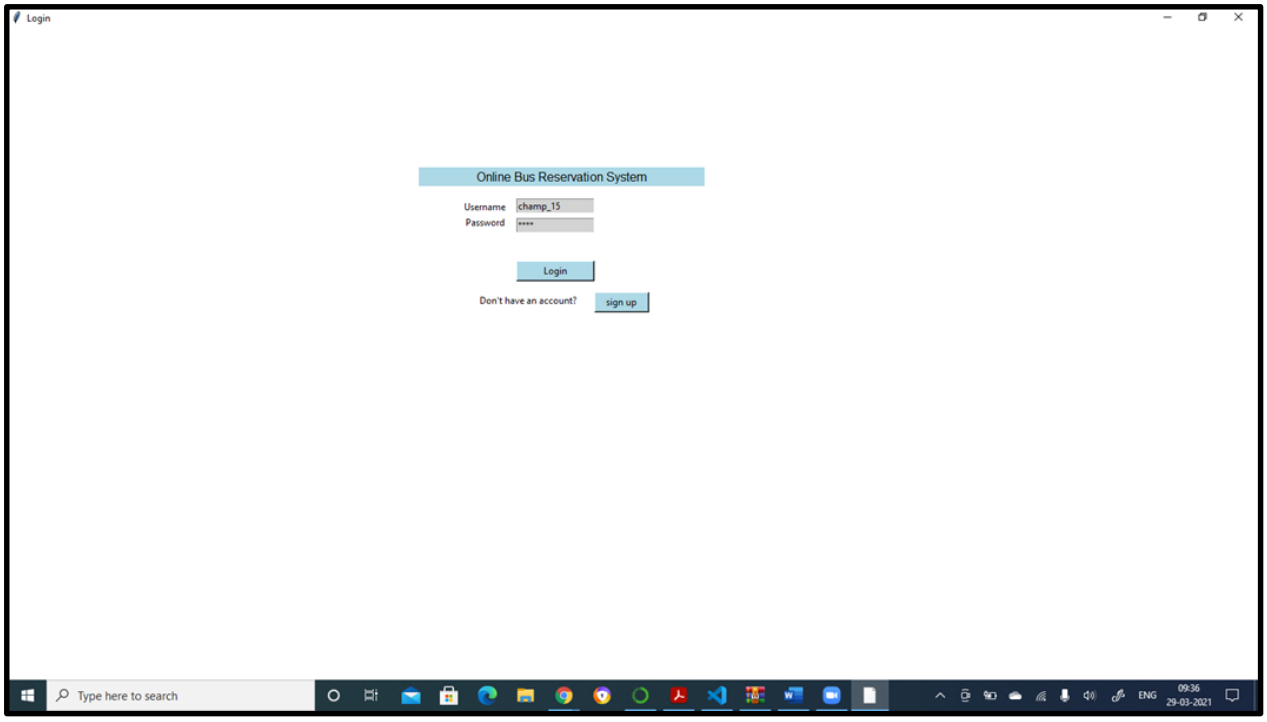
```
from tkinter import *
from tkinter import messagebox
window=Tk()


window.title("Login")
window.geometry("1450x750")
window.config(bg="white")
def con(a):
    a.config(bg="white")
def b(j):
    j.config(bg="light grey")
login=Label(window,text="Online Bus Reservation System",font=(20),width=40)
login.place(x=525,y=180)
login.config(bg="light blue")

user=Label(window,text="Username")
user.place(x=560,y=220,width=100)
con(user)

pw=Label(window,text="Password")
pw.place(x=560,y=240,width=100)
con(pw)

u=Entry(window,width=50)
```

```
u.place(x=650,y=220,width=100)
b(u)
p=Entry(window,show="*",width=50)
p.place(x=650,y=245,width=100)
b(p)

def dashboard():
   window1=Tk()
   window1.geometry("1450x750")
   window1.title("Dashbaord")
   def quit():
      window1.destroy()
      messagebox.showinfo("status","logged out successfully")
   window1.config(bg="white")
   name=Label(window1,text="Online Bus Reservation
Syatem",font=(15),width=40,bg="light blue",fg="black").place(x=550,y=10)
   home=Button(window1,text="Settings",font=(6),width=10,height=1,bg="light
blue",fg="black").place(x=20,y=10)
   quit=Button(window1,text="Log out",font=(6),width=10,height=1,bg="light
blue",fg="black",command=quit).place(x=1275,y=10)
   sd=Button(window1,text="Souce station",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=60,y=150)
   bill=Button(window1,text="Destination station",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=600,y=150)
   cusdb=Button(window1,text="Payments",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=1200,y=150)
   invo=Button(window1,text="INVOICES",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=60,y=500)
   rep=Button(window1,text="REPORTS",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=600,y=500)
   more=Button(window1,text="MORE\nOPTIONS",font=(10),width=20,height=3,bg="light
blue",fg="black").place(x=1200,y=500)
   window1.mainloop()

log=Button(text="Login",width=100,command=dashboard)
log.place(x=650,y=300,width=100)
log.config(bg="light blue")

don=Label(window,text="Don't have an account?")
don.place(x=600,y=340,width=130)
con(don)


def sign_up():

   supage=Tk()
   supage.title("Register")
   supage.geometry("600x600")
   supage.config(bg="light blue")
```

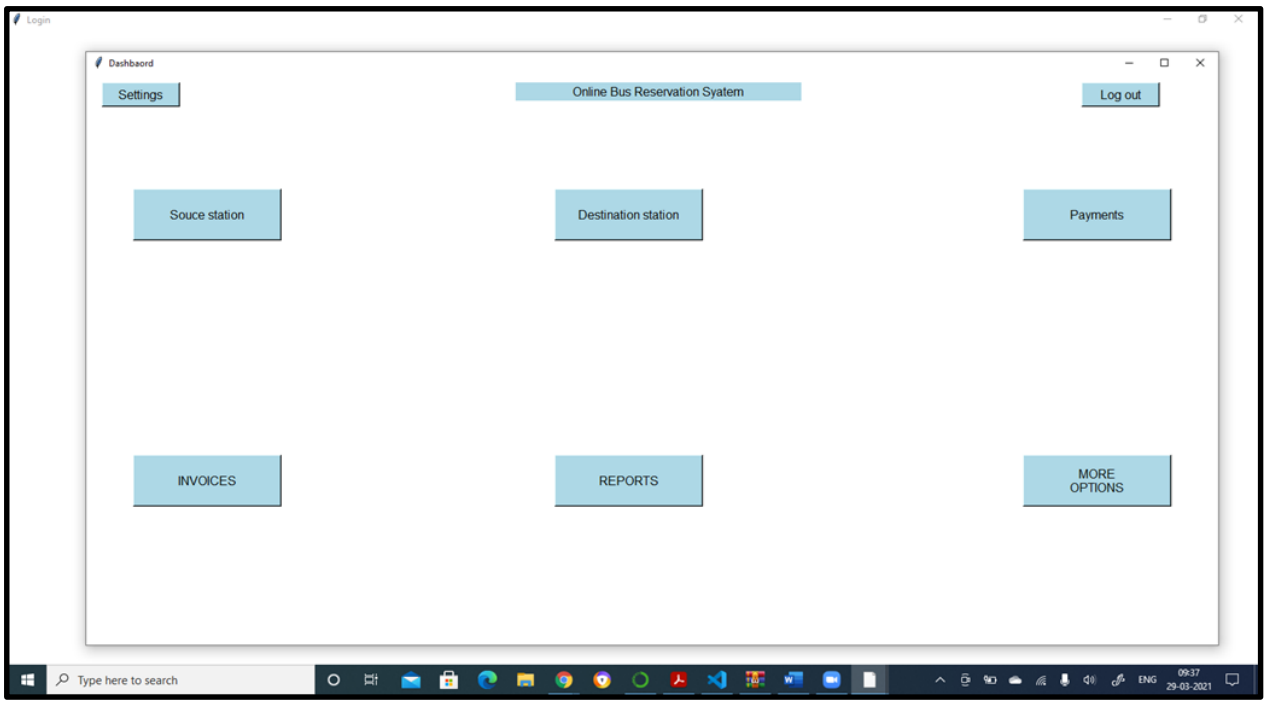
```
    head=Label(supage,text="Register your account",bg="light
blue",font=("bold",18)).grid(column=1,row=0,columnspan=3,padx=10,pady=10)

    fn=Label(supage,text="First name",bg="light
blue").grid(column=0,row=1,padx=10,pady=10)
    t2=Entry(supage,width=30,bg="white").grid(column=1,row=1,padx=10,pady=10)

    ln=Label(supage,text="Last name",bg="light
blue").grid(column=0,row=2,padx=10,pady=10)
    t3=Entry(supage,width=30).grid(column=1,row=2,padx=10,pady=10)

    dob=Label(supage,text="Date of birth",bg="light blue")
    dob.grid(column=0,row=3,padx=10,pady=10)
    dates=[
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]

months=["january","febrary","march","april","may","june","july","august","september","octo
ber","november","december"]

years=[1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2
010,2011,2012,2013,2015,2016]

    c=StringVar()
    droplist=OptionMenu(supage,c, *dates)
    droplist.config(width=5)
    c.set("date")
    droplist.grid(row=3,column=1,padx=10,pady=10)

    c=StringVar()
    droplist=OptionMenu(supage,c, *months)
    droplist.config(width=5)
    c.set("month")
    droplist.grid(row=3,column=2,padx=10,pady=10)

    c=StringVar()
    droplist=OptionMenu(supage,c, *years)
    droplist.config(width=5)
    c.set("year")
    droplist.grid(row=3,column=3,padx=10,pady=10)

    bn=Label(supage,text="Gender",bg="light blue")
    bn.grid(column=0,row=4,padx=10,pady=10)
    rad1 = Radiobutton(supage,text='male',value=1).grid(column=1,row=4,padx=10,pady=10)
    rad2 =
Radiobutton(supage,text='female',value=0).grid(column=2,row=4,pady=10,padx=10)

    email=Label(supage,text="Email address",bg="light
blue").grid(column=0,row=5,padx=10,pady=10)
    t6=Entry(supage,width=30).grid(column=1,row=5,padx=10,pady=10)
```

```
   mn=Label(supage,text="Mobile                                    number",bg="light
blue").grid(column=0,row=6,padx=10,pady=10)
   t5=Entry(supage,width=30).grid(column=1,row=6,pady=10,padx=10)

   pw=Label(supage,text="Password",bg="light
blue").grid(column=0,row=7,padx=10,pady=10)
   t7=Entry(supage,show="*",width=30).grid(column=1,row=7,padx=10,pady=10)

   cpw=Label(supage,text="Confirm                                  Password",bg="light
blue").grid(column=0,row=8,padx=10,pady=10)
   t7=Entry(supage,show="*",width=30).grid(column=1,row=8,padx=10,pady=10)

   st=Label(supage,text="State",bg="light blue").grid(column=0,row=9,padx=10,pady=10)
   t8=Entry(supage,width=30).grid(column=1,row=9,padx=10,pady=10)

   country=Label(supage,text="Country",bg="light
blue").grid(column=0,row=10,padx=10,pady=10)
   t8=Entry(supage,width=30).grid(column=1,row=10,padx=10,pady=10)


submit=Button(supage,text="Submit",font=("bold",10),fg="Black",bg="white").grid(row=11
,column=1,columnspan=3)

   window.mainloop()

sp=Button(window,text="sign up",width=60,command=sign_up)
sp.place(x=750,y=340,width=70)
sp.config(bg="light blue")

window.mainloop()
```

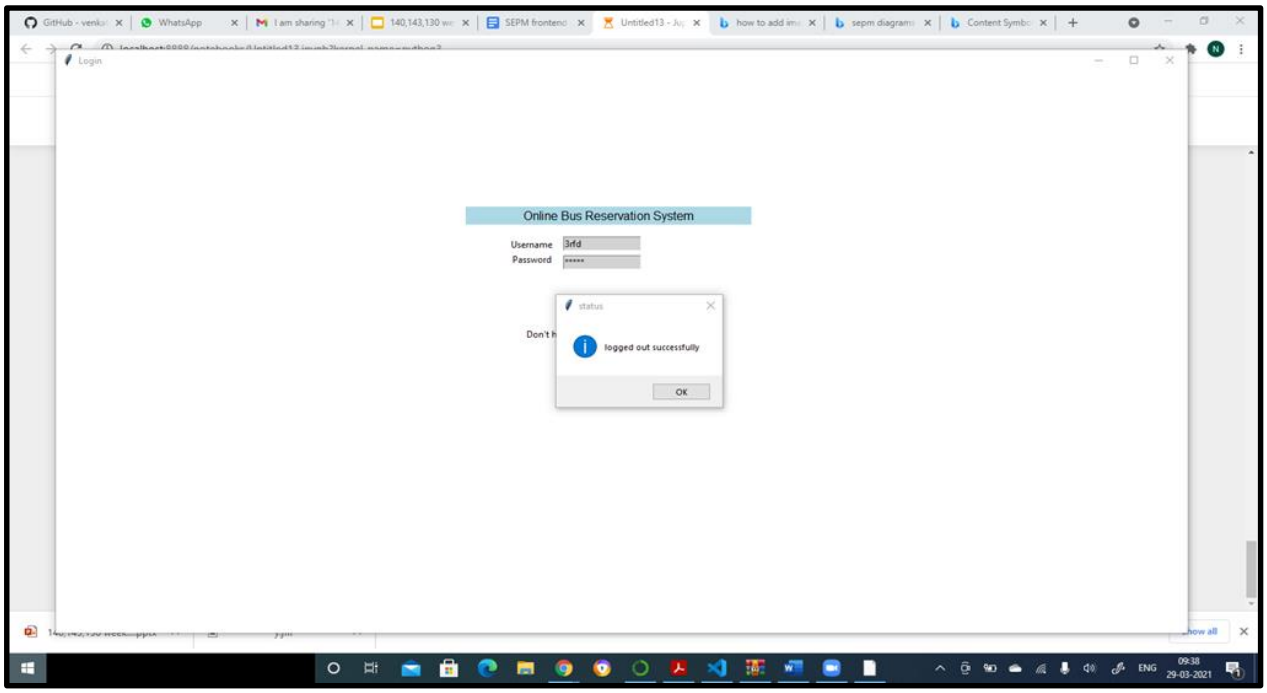
This is the login page ,where we enter the basic details such as username and password.



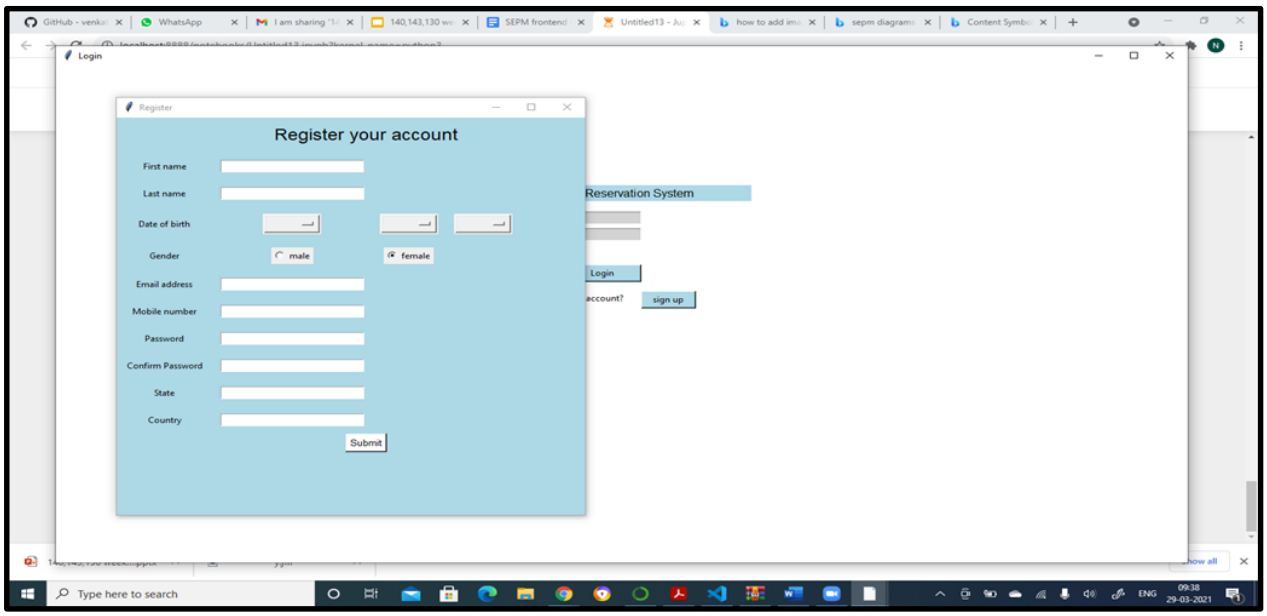
After successfully login,u will enter into one more page where u have to do select the source and destination stations and have to do the payment to get the ticket.



After doing payment,u have to click the logout button,then a message will be displayed as u have logout successfully



If u don’t have an account ,u have to click the signup option and u have to register by filling the basic personal details such as name,password,mobile number,state,password….etc,then u can successfully login.



CODING MODULE – 2

```
from tkinter import *
from tkinter import messagebox
import os
window=Tk()

window.title("Login")
window.geometry("1450x750")
window.config(bg="white")
def con(a):
    a.config(bg="white")
def b(j):
    j.config(bg="light grey")

login=Label(window,text="ONLINE BUS RESERVATION SYSTEM
",font=(20),width=40)
login.place(x=500,y=180)
login.config(bg="light blue")

user=Label(window,text="Username")
user.place(x=560,y=220,width=100)
con(user)

pw=Label(window,text="Password")
pw.place(x=560,y=240,width=100)
con(pw)

u=Entry(window,width=50)
```

```
u.place(x=650,y=220,width=100)
b(u)
p=Entry(window,show="*",width=50)
p.place(x=650,y=245,width=100)
b(p)

def check():
    user=u.get()
    passwd=p.get()
    if(user == "namratha" and passwd == "1111"):
        dashboard()
    else:
        msg=messagebox.showinfo("error","Entered username or password is wrong!")



def UPIid():
    UPIid=Tk()
    def back():
        UPIid.destroy()
        dashboard()
    UPIid.geometry("600x730")
    UPIid.config(bg="light blue")
    UPIid.title("UPI")
    dash=Button(UPIid,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(UPIid,text="UPI Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    UPI1=Label(UPIid,text="Enter@UPI",bg="coral1").place(x=100,y=150)
    t1=Entry(UPIid,width=35).place(x=170,y=150)
    Pay=Button(UPIid,text="Pay",height=2,width=10,bg="violet").place(x=240,y=220)
    UPIid.mainloop()

def Sourcestation():
    Sourcestation=Tk()
    def back():
        Sourcestation.destroy()
        dashboard()
    Sourcestation.geometry("600x650")
    Sourcestation.config(bg="light blue")
    Sourcestation.title("Sourcestation")

dash=Button(Sourcestation,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)

ti=Label(Sourcestation,text="Sourcestation",font=("bold",20),bg="magenta").place(x=180,y=
0)
    a=["Anna Nagar East","Anna Nagar Tower","Alandur","Arumbakkam","Ashok
Nagar","Chennai International Airport","Ekkattuthangal","Government
Estate","Guindy","High Court","Kaladipet","Kilpauk","Koyambedu","St. Thomas
Mount","Teynampet","Thirumangalam","Tiruvottriyur","Tondiarpet","Vadapalani","Washer
manpet","Wimco Nagar"]
    c=StringVar(Sourcestation)
```

```
    droplist=OptionMenu(Sourcestation,c, *a)
    droplist.config(width=12)
    c.set("Sourcestation")
    droplist.place(x=20,y=100)
    Sourcestation.mainloop()

def Destinationstation():
    Destinationstation=Tk()
    def back():
        Destinationstation.destroy()
        dashboard()
    Destinationstation.geometry("600x650")
    Destinationstation.config(bg="light blue")
    Destinationstation.title("Destinationstation")

dash=Button(Destinationstation,text="Dashbaord",bg="blue",command=back).place(x=475,y
=35)

ti=Label(Destinationstation,text="Destinationstation",font=("bold",20),bg="magenta").place(
x=180,y=0)
    a=["Anna Nagar East","Anna Nagar Tower","Alandur","Arumbakkam","Ashok
Nagar","Chennai International Airport","Ekkattuthangal","Government
Estate","Guindy","High Court","Kaladipet","Kilpauk","Koyambedu","St. Thomas
Mount","Teynampet","Thirumangalam","Tiruvottriyur","Tondiarpet","Vadapalani","Washer
manpet","Wimco Nagar"]
    c=StringVar(Destinationstation)
    droplist=OptionMenu(Destinationstation,c, *a)
    droplist.config(width=15)
    c.set("Destinationstation")
    droplist.place(x=20,y=100)
    Destinationstation.mainloop()

def transaction():
    transaction=Tk()
    transaction.title("Net BAnking Payment")
    transaction.geometry("600x740")
    transaction.config(bg="light green")
    def con(a):
        a.config(bg="light green")
    def b(j):
        j.config(bg="light grey")

    login=Label(transaction,text="Net Banking Payment",font=(10),width=30)
    login.place(x=50,y=180)
    login.config(bg="Violet")

    user=Label(transaction,text="Username")
    user.place(x=50,y=220,width=100)
    con(user)
```

```
    pw=Label(transaction,text="Password")
    pw.place(x=50,y=240,width=100)
    con(pw)

    u=Entry(transaction,width=50)
    u.place(x=150,y=220,width=100)
    b(u)
    p=Entry(transaction,show="*",width=50)
    p.place(x=150,y=245,width=100)
    b(p)
    Pay=Button(transaction,text="Pay",height=2,width=10,bg="violet").place(x=150,y=300)
    transaction.mainloop()

def MOREOPTIONS():
    MOREOPTIONS=Tk()
    def back():
        MOREOPTIONS.destroy()
        dashboard()
    MOREOPTIONS.geometry("600x730")
    MOREOPTIONS.config(bg="pink")
    MOREOPTIONS.title("MOREOPTIONS")

dash=Button(MOREOPTIONS,text="Dashbaord",bg="blue",command=back).place(x=475,y
=45)
    ti=Label(MOREOPTIONS,text="MORE
OPTIONS",font=("bold",20),bg="Red").place(x=200,y=0)

    MOREOPTIONS1=Label(MOREOPTIONS,text="Any
Query",bg="coral1").place(x=50,y=90)
    t1=Entry(MOREOPTIONS,width=85).place(x=50,y=120)
    MOREOPTIONS2=Label(MOREOPTIONS,text="Solutions to improve our
application",bg="coral1").place(x=50,y=150)
    t2=Entry(MOREOPTIONS,width=85).place(x=50,y=180)
    MOREOPTIONS3=Label(MOREOPTIONS,text="Give Rating(Out of
5)",bg="coral1").place(x=50,y=210)
    t3=Entry(MOREOPTIONS,width=20).place(x=50,y=240)

MOREOPTIONS3=Label(MOREOPTIONS,text="Feedback",bg="coral1").place(x=50,y=27
0)
    t3=Entry(MOREOPTIONS,width=20).place(x=50,y=300)

Submit=Button(MOREOPTIONS,text="Submit",height=2,width=10,bg="violet").place(x=22
5,y=375)
    MOREOPTIONS.mainloop()

def Credit():
    Credit=Tk()
    def back():
        Credit.destroy()
        dashboard()
```

```
    Credit.geometry("600x730")
    Credit.config(bg="pink")
    Credit.title("Credit card")
    dash=Button(Credit,text="Dashbaord",bg="blue",command=back).place(x=475,y=45)
    ti=Label(Credit,text="Credit card
Payments",font=("bold",20),bg="Red").place(x=200,y=0)

    Credit1=Label(Credit,text="Card Number",bg="coral1").place(x=50,y=90)
    t1=Entry(Credit,width=40).place(x=50,y=120)
    Credit2=Label(Credit,text="Expiry",bg="coral1").place(x=50,y=150)
    t2=Entry(Credit,width=15).place(x=50,y=180)
    Credit3=Label(Credit,text="CVV",bg="coral1").place(x=200,y=150)
    t3=Entry(Credit,width=10).place(x=200,y=181)
    Credit4=Label(Credit,text="Name on card",bg="coral1").place(x=50,y=210)
    t4=Entry(Credit,width=40).place(x=50,y=240)
    Pay=Button(Credit,text="Pay",height=2,width=10,bg="violet").place(x=240,y=300)

    Credit.mainloop()

def Debit():
    Debit=Tk()
    def back():
        Debit.destroy()
        dashboard()
    Debit.geometry("600x730")
    Debit.config(bg="pink")
    Debit.title("Debit card")
    dash=Button(Debit,text="Dashbaord",bg="blue",command=back).place(x=475,y=45)
    ti=Label(Debit,text="Debit card Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    Debit1=Label(Debit,text="Card Number",bg="coral1").place(x=50,y=90)
    t1=Entry(Debit,width=40).place(x=50,y=120)
    Debit2=Label(Debit,text="Expiry",bg="coral1").place(x=50,y=150)
    t2=Entry(Debit,width=15).place(x=50,y=180)
    Debit3=Label(Debit,text="CVV",bg="coral1").place(x=200,y=150)
    t3=Entry(Debit,width=10).place(x=200,y=181)
    Debit4=Label(Debit,text="Name on card",bg="coral1").place(x=50,y=210)
    t4=Entry(Debit,width=40).place(x=50,y=240)
    Pay=Button(Debit,text="Pay",height=2,width=10,bg="violet").place(x=240,y=300)
    Debit.mainloop()

def Banks():
    Banks=Tk()
    def back():
        Banks.destroy()
        dashboard()
    Banks.geometry("600x730")
    Banks.config(bg="light blue")
    Banks.title("PAYMENTS")
    dash=Button(Banks,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(Banks,text="Banks",font=("bold",20),bg="magenta").place(x=180,y=0)
```

```
    Bank1=Button(Banks,text="State Bank of
India",font=("bold",10),command=transaction).place(x=20,y=120)
    Bank2=Button(Banks,text="City Union
Bank",font=("bold",10),command=transaction).place(x=20,y=180)
    Bank3=Button(Banks,text="Andhra
Bank",font=("bold",10),command=transaction).place(x=20,y=240)
    Bank4=Button(Banks,text="Canara
Bank",font=("bold",10),command=transaction).place(x=20,y=300)
    Bank5=Button(Banks,text="Axis
Bank",font=("bold",10),command=transaction).place(x=20,y=360)
    Bank6=Button(Banks,text="Syndicate
Bank",font=("bold",10),command=transaction).place(x=20,y=420)
    Bank7=Button(Banks,text="ICICI
Bank",font=("bold",10),command=transaction).place(x=20,y=480)
    Bank8=Button(Banks,text="HDFC
Bank",font=("bold",10),command=transaction).place(x=20,y=540)
    Bank9=Button(Banks,text="CITI
Bank",font=("bold",10),command=transaction).place(x=20,y=600)
    Banks.mainloop()

def UPI():
    UPI=Tk()
    def back():
        UPI.destroy()
        dashboard()
    UPI.geometry("600x730")
    UPI.config(bg="pink")
    UPI.title("UPI")
    dash=Button(UPI,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(UPI,text="UPI Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    UPI1=Button(UPI,text="Paytm",font=("bold",10),command=UPIid).place(x=100,y=120)
    UPI2=Button(UPI,text="Gpay",font=("bold",10),command=UPIid).place(x=100,y=300)
    UPI3=Button(UPI,text="Phone
pe",font=("bold",10),command=UPIid).place(x=400,y=120)
    UPI4=Button(UPI,text="Amazon
pay",font=("bold",10),command=UPIid).place(x=400,y=300)
    UPI.mainloop()

def payments():
    payments=Tk()
    def back():
        payments.destroy()
        dashboard()
    payments.geometry("600x730")
    payments.config(bg="light blue")
    payments.title("PAYMENTS")
    dash=Button(payments,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(payments,text=" Payments",font=("bold",20),bg="blue").place(x=180,y=0)
    pay1=Button(payments,text="Debit
card",font=("bold",10),command=Debit).place(x=20,y=120)
```

```
    pay2=Button(payments,text="Credit
card",font=("bold",10),command=Credit).place(x=20,y=180)
    pay3=Button(payments,text="Net
Banking",font=("bold",10),command=Banks).place(x=20,y=240)
    pay4=Button(payments,text="UPI
payments",font=("bold",10),command=UPI).place(x=20,y=300)

    payments.mainloop()

def reports():
    rep=Tk()
    def back():
        rep.destroy()
        dashboard()
    rep.geometry("600x730")
    rep.config(bg="light blue")
    rep.title("REPORTS")
    dash=Button(rep,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(rep,text=" REPORTS",font=("bold",20),bg="magenta").place(x=180,y=0)
    year1=Button(rep,text="REPORT FOR 2014-2015",font=("bold",10)).place(x=20,y=60)
    year2=Button(rep,text="REPORT FOR 2015-2016",font=("bold",10)).place(x=20,y=120)
    year3=Button(rep,text="REPORT FOR 2016-2017",font=("bold",10)).place(x=20,y=180)
    year4=Button(rep,text="REPORT FOR 2017-2018",font=("bold",10)).place(x=20,y=240)
    year5=Button(rep,text="REPORT FOR 2018-2019",font=("bold",10)).place(x=20,y=300)
    year6=Button(rep,text="REPORT FOR 2019-2020",font=("bold",10)).place(x=20,y=360)
    year7=Button(rep,text="REPORT FOR 2020-2021",font=("bold",10)).place(x=20,y=420)
    update=Button(rep,text="Update",height=2,width=10,bg="violet").place(x=50,y=500)
    prin=Button(rep,text="Print",height=2,width=10,bg="violet").place(x=250,y=500)
    generate=Button(rep,text="Generate",height=2,width=10,bg="violet").place(x=450,y=500)
    rep.mainloop()


def dashboard():
    window1=Tk()
    window1.geometry("1450x750")
    window1.title("Dashbaord")
    def quit():
        window1.destroy()
        messagebox.showinfo("status","logged out successfully")
    window1.config(bg="White")
    name=Label(window1,text="Oline bus Reservation System",font=(15),width=40,bg="light
blue",fg="black").place(x=550,y=10)
    home=Button(window1,text="Settings",font=(6),width=10,height=1,bg="light
blue",command=setting).place(x=20,y=10)
    quit=Button(window1,text="Log out",font=(6),width=10,height=1,bg="light
blue",command=quit).place(x=1275,y=10)
    sd=Button(window1,text="Source station",font=(10),width=20,height=3,bg="light
blue",command=Sourcestation).place(x=60,y=150)
    bill=Button(window1,text="Destination station",font=(10),width=20,height=3,bg="light
blue",fg="black",command=Destinationstation).place(x=600,y=150)
```

```
    cusdb=Button(window1,text="Payments",font=(10),width=20,height=3,bg="light
blue",command=payments).place(x=1200,y=150)
    invo=Button(window1,text="INVOICES",font=(10),width=20,height=3,bg="light
blue").place(x=60,y=500)
    rep=Button(window1,text="REPORTS",font=(10),width=20,height=3,bg="light
blue",command=reports).place(x=600,y=500)
    more=Button(window1,text="MORE\nOPTIONS",font=(10),width=20,height=3,bg="light
blue",command=MOREOPTIONS ).place(x=1200,y=500)
    window1.mainloop()

log=Button(text="Login",width=100,command=check)
log.place(x=650,y=300,width=100)
log.config(bg="light blue")

don=Label(window,text="Don't have an account?")
don.place(x=600,y=340,width=130)
con(don)

def contactus():
    contactus=Tk()
    def back():
        contactus.destroy()
        dashboard()
    contactus.geometry("600x730")
    contactus.config(bg="light blue")
    contactus.title("setting")
    dash=Button(contactus,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(contactus,text=" setting",font=("bold",20),bg="magenta").place(x=180,y=0)

name=Label(contactus,1,text="namratha.nayani@gmail.com",font=(15),width=40,bg="light
blue",fg="black").place(x=20,y=10)
    contactus.mainloop()

def setting():
    setting=Tk()
    def back():
        setting.destroy()
        dashboard()
    setting.geometry("600x730")
    setting.config(bg="light blue")
    setting.title("setting")
    dash=Button(setting,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(setting,text=" setting",font=("bold",20),bg="magenta").place(x=180,y=0)
    changepassword=Button(setting,text="CHANGE
PASSWORD",font=("bold",10)).place(x=20,y=60)
    contactus=Button(setting,text="CONTACT US",font=("bold",10)).place(x=20,y=120)
    setting.mainloop()
```

```
def sign_up():

    supage=Tk()
    supage.title("Register")
    supage.geometry("600x600")
    supage.config(bg="mistyrose2")

    head=Label(supage,text="Register your
account",bg="red",font=("bold",18)).grid(column=1,row=0,columnspan=3,padx=10,pady=10
)

    fn=Label(supage,text="First name     ").grid(column=0,row=1,padx=10,pady=10)
    t2=Entry(supage,width=30,bg="white").grid(column=1,row=1,padx=10,pady=10)

    ln=Label(supage,text="Last name      ").grid(column=0,row=2,padx=10,pady=10)
    t3=Entry(supage,width=30).grid(column=1,row=2,padx=10,pady=10)

    dob=Label(supage,text="Date of Birth")
    dob.grid(column=0,row=3,padx=10,pady=10)
    dates=[
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]

months=["january","febrary","march","april","may","june","july","august","september","octo
ber","november","december"]

years=[1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2
010,2011,2012,2013,2015,2016]

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *dates)
    droplist.config(width=5)
    c.set("date")
    droplist.grid(row=3,column=1,padx=10,pady=10)

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *months)
    droplist.config(width=5)
    c.set("month")
    droplist.grid(row=3,column=2,padx=10,pady=10)

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *years)
    droplist.config(width=5)
    c.set("year")
    droplist.grid(row=3,column=3,padx=10,pady=10)

    bn=Label(supage,text="Gender        ")
    bn.grid(column=0,row=4,padx=10,pady=10)
    rad1 = Radiobutton(supage,text='male',value=1).grid(column=1,row=4,padx=10,pady=10)
```

```
    rad2 =
Radiobutton(supage,text='female',value=0).grid(column=2,row=4,pady=10,padx=10)

    em=Label(supage,text="Email address     ").grid(column=0,row=6,padx=10,pady=10)
    t6=Entry(supage,width=30).grid(column=1,row=6,padx=10,pady=10)

    mn=Label(supage,text="Mobile number     ").grid(column=0,row=7,padx=10,pady=10)
    t5=Entry(supage,width=30).grid(column=1,row=7,pady=10,padx=10)

    pw=Label(supage,text="Password          ").grid(column=0,row=8,padx=10,pady=10)
    t7=Entry(supage,show="*",width=30).grid(column=1,row=8,padx=10,pady=10)

    cpw=Label(supage,text="Confirm Password ").grid(column=0,row=9,padx=10,pady=10)
    t7=Entry(supage,show="*",width=30).grid(column=1,row=9,padx=10,pady=10)

    st=Label(supage,text="State             ").grid(column=0,row=10,padx=10,pady=10)
    t8=Entry(supage,width=30).grid(column=1,row=10,padx=10,pady=10)

    ctry=Label(supage,text="Country         ").grid(column=0,row=11,padx=10,pady=10)
    t8=Entry(supage,width=30).grid(column=1,row=11,padx=10,pady=10)

    em=Label(supage,text="Username     ").grid(column=0,row=5,padx=10,pady=10)
    t6=Entry(supage,width=30).grid(column=1,row=5,padx=10,pady=10)

    def created():
        supage.destroy()
        msg=messagebox.showinfo("Registered","Completed registration successfully")

submit=Button(supage,text="Submit",font=("bold",10),fg="Black",bg="green3",command=c
reated).grid(row=12,column=1,columnspan=3)

    supage.mainloop()

sp=Button(window,text="sign up",width=60,command=sign_up)
sp.place(x=750,y=340,width=70)
sp.config(bg="light blue")

window.mainloop()
```

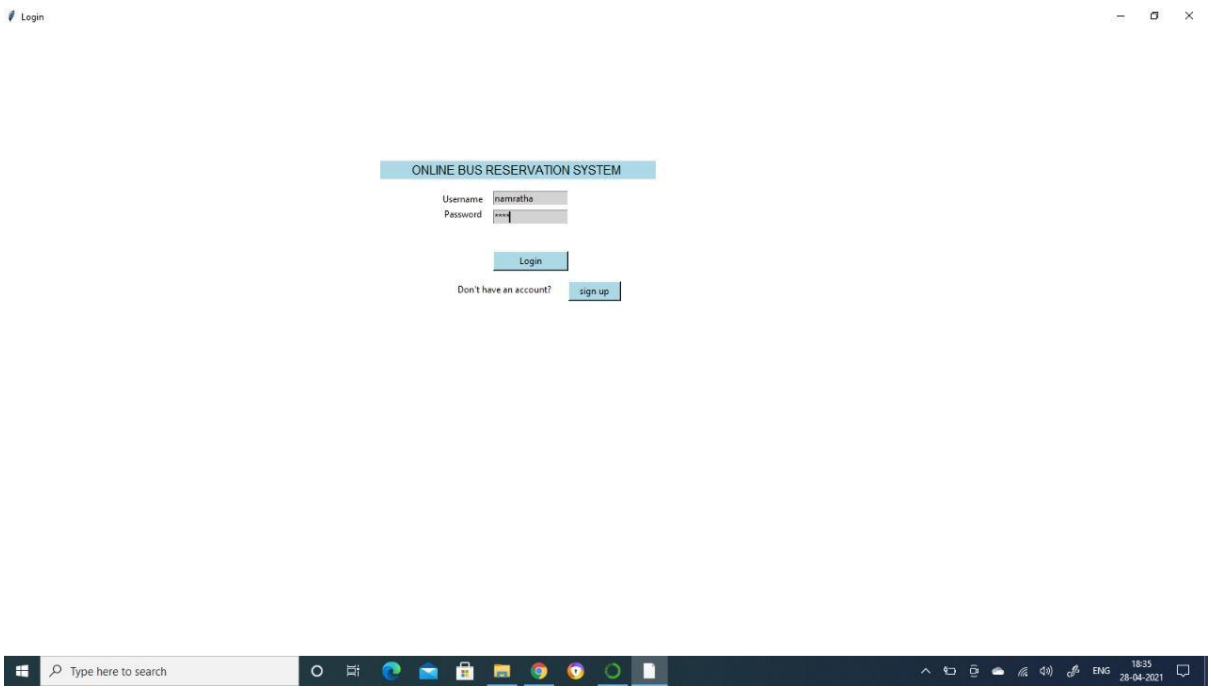
LOGIN PAGE

->LOGIN

If you have an account then u can give your username and password to login to the page

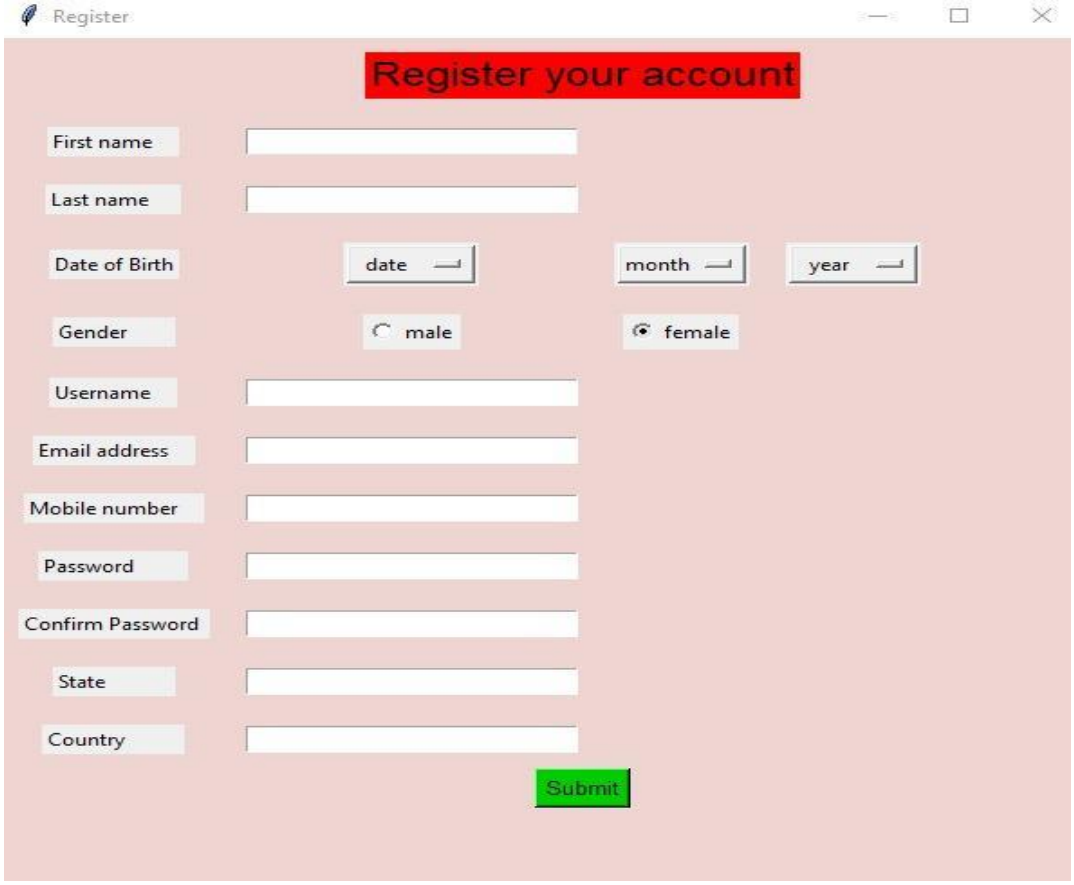
->SIGN IN

If you don't have an account u have to click the sign in button to register yourself



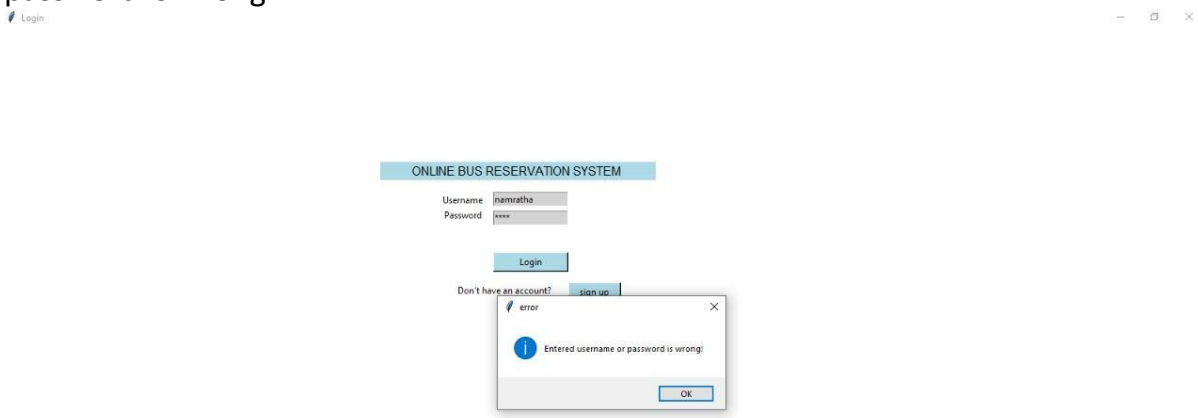
REGISTRATION

After u click the sign in button ,the register dialogue box will appear. there you have to fill in all your details such as name , user name , password, country etc



->INCORRECT USERNAME/PASSWORD

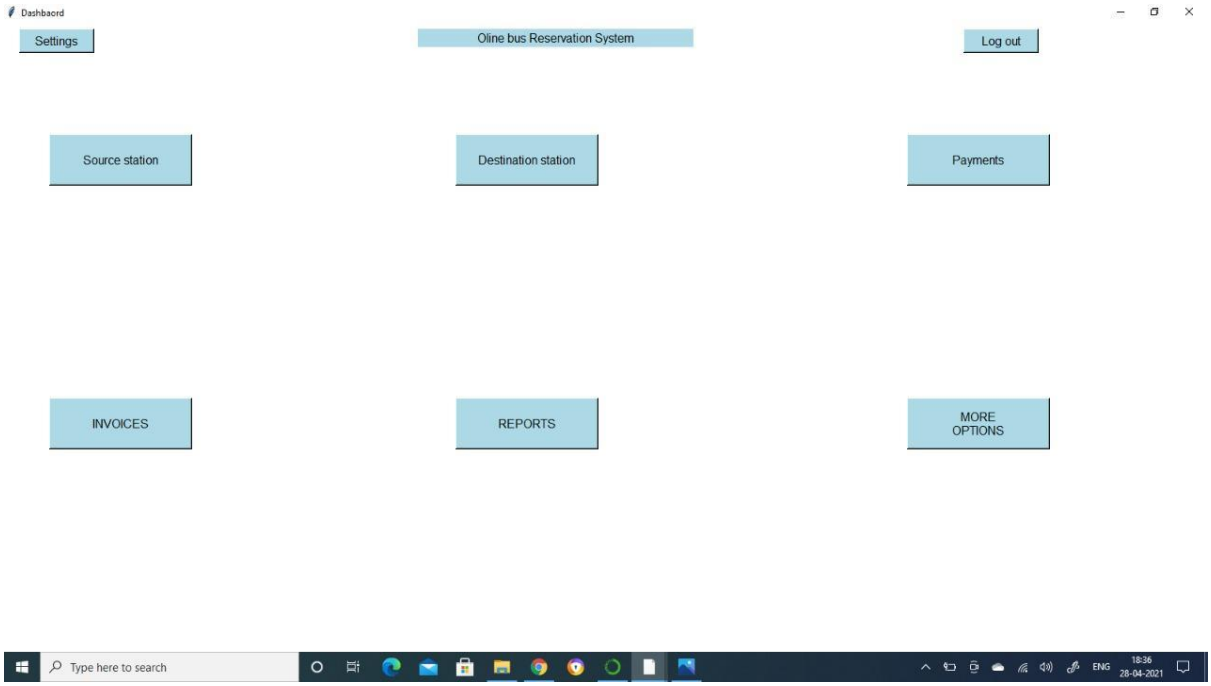
After registering yourself u have to give the correct username and password otherwise u will get an error message mentioning that the entered username or password is wrong



->DASH BOARD

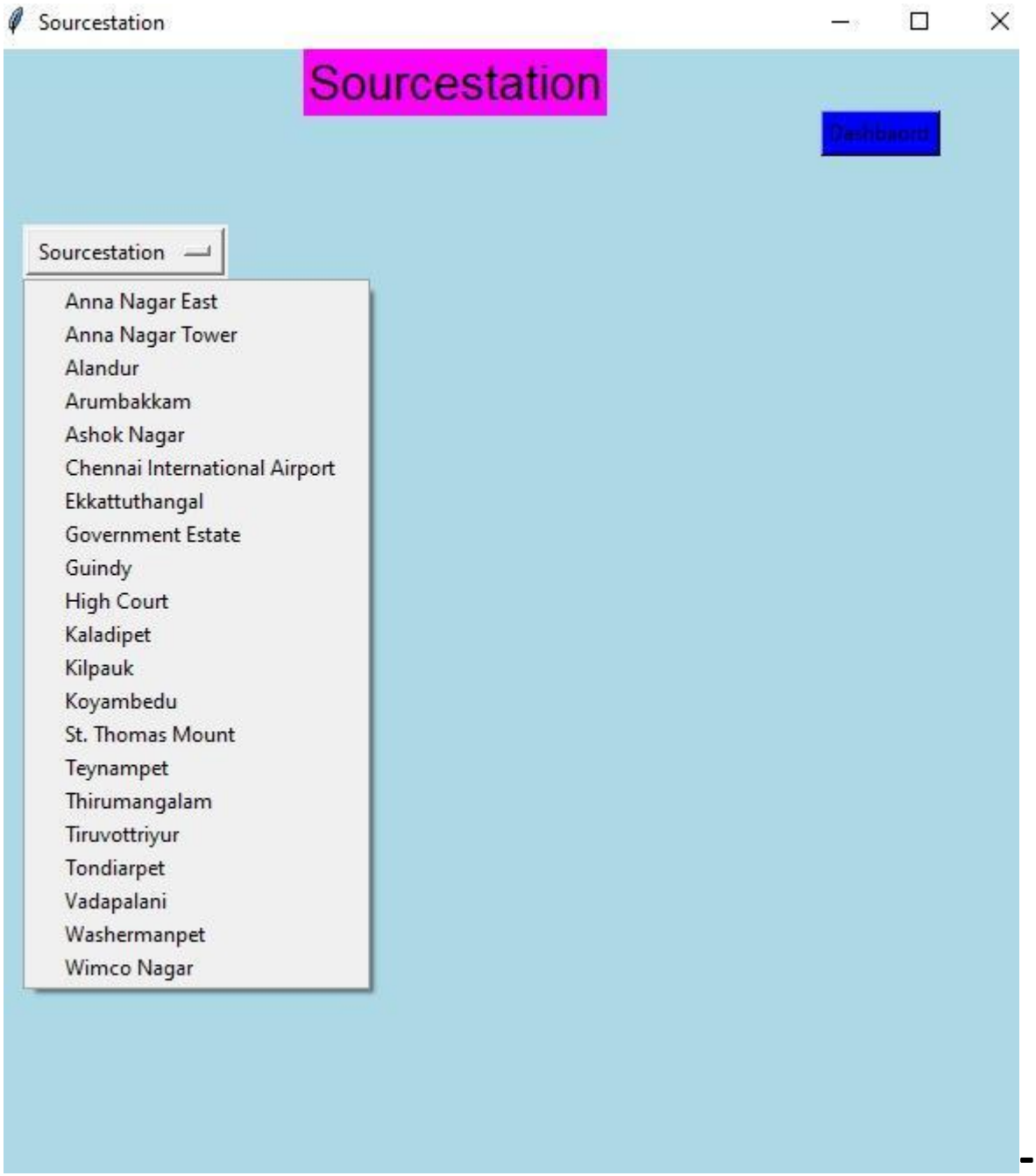
After you log in successfully , you will enter a new page called the dashboard where u have different options like

- source station
- destination station
- payments
- invoices
- reports
- more options



->SOURCE STATION

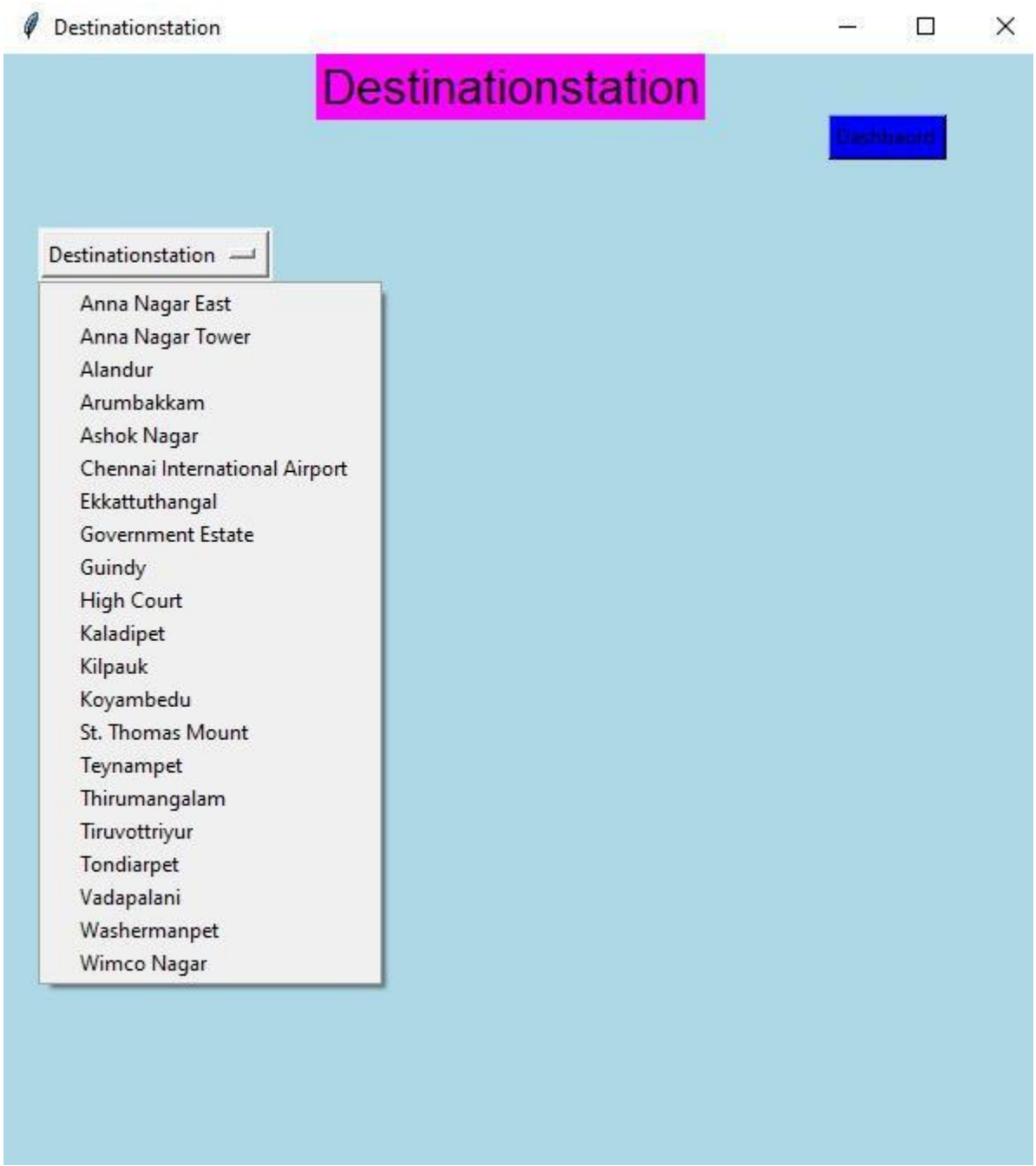
After u click on the source station , a drop down menu will appear from which u can select a station



--

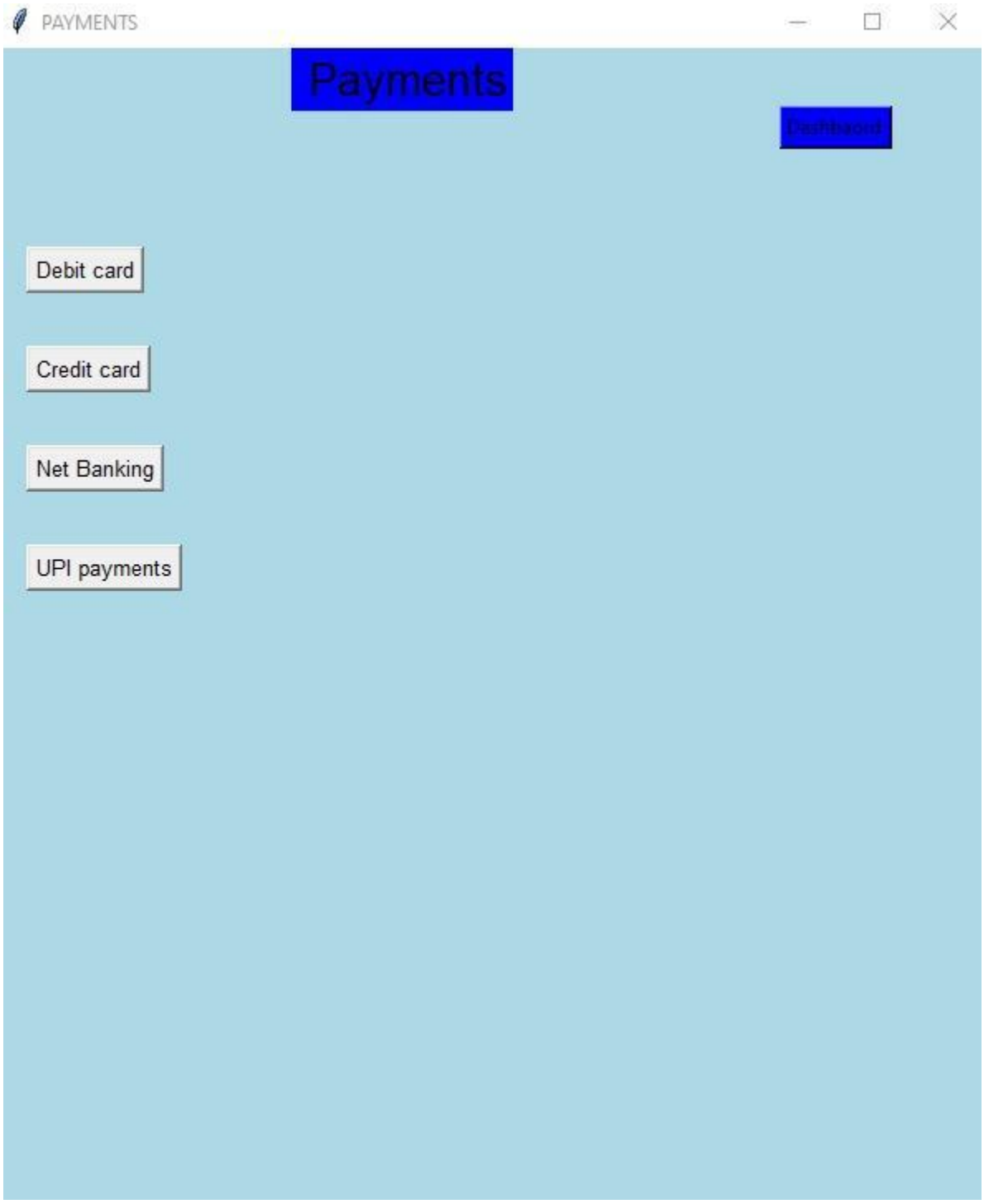
>DESTINATION STATION

After selecting the source station select the destination station in the same way



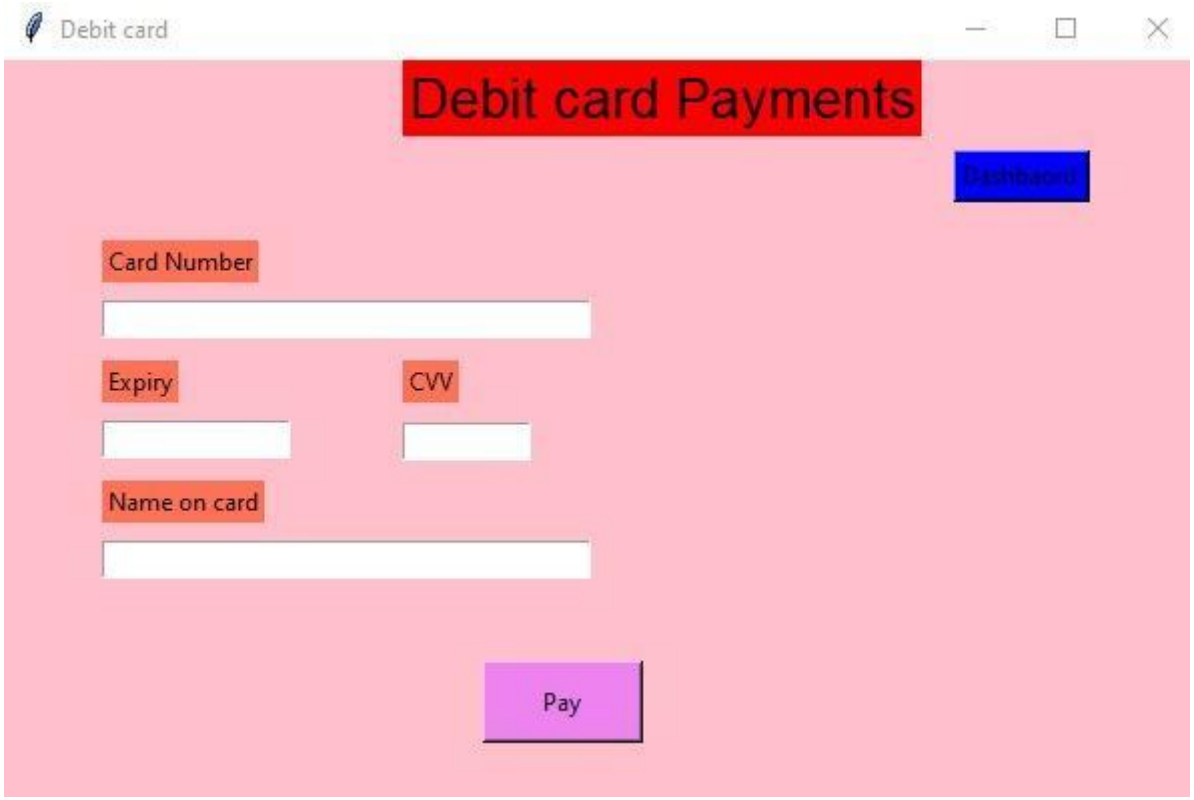
->PAYMENTS

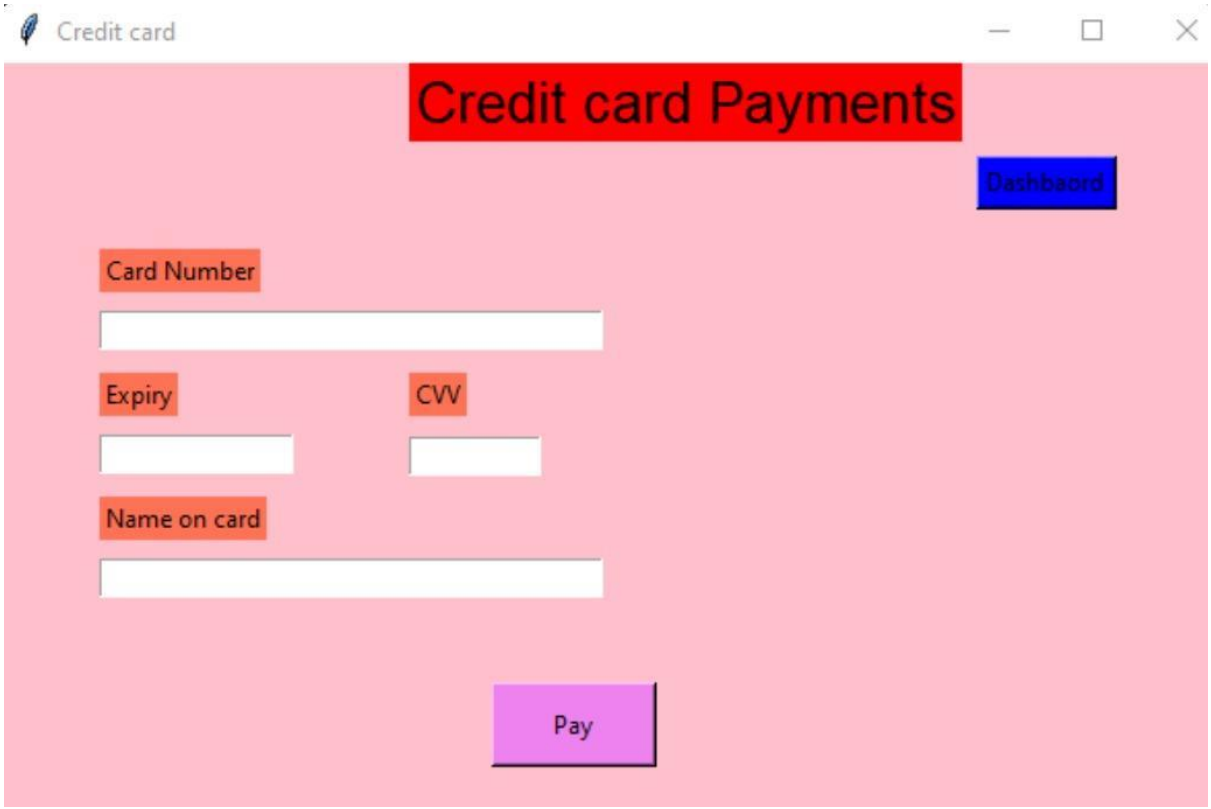
Next comes the payment option where u have to do the payment. Here when u click on payments option a new dialogue box will appear which contains different payment methods



->CREDIT/DEBIT CARD

FOR THE PAYMENT THROUGH CREDIT OR DEBIT CARDS,YOU HAVE TO GIVE THE NECESSARY DETAILS OF THE CARD AND CLICK ON PAY



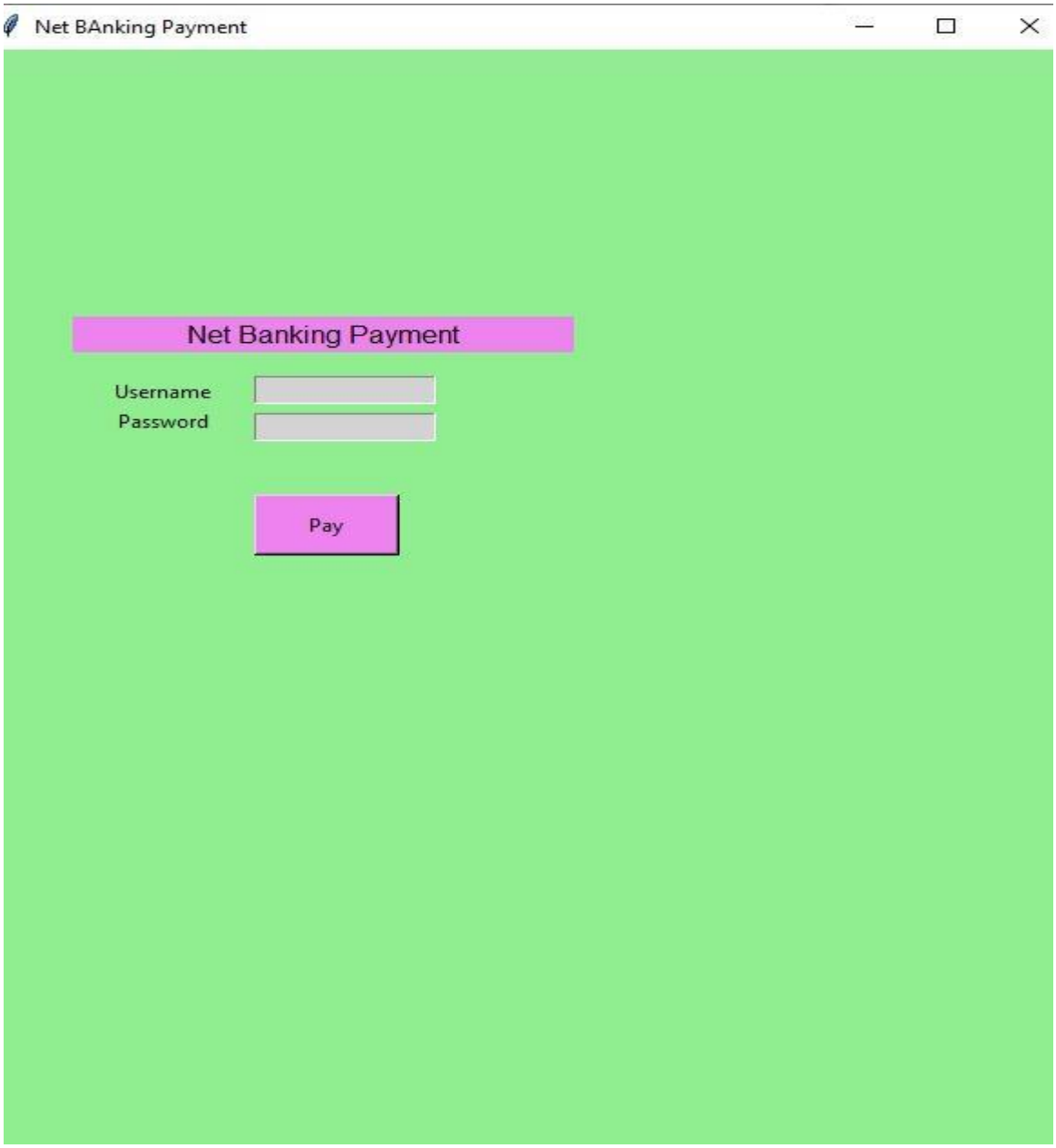


->NET BANKING

If u select the net banking option , a new window will appear where we have different banks

->NET BANKING PAGE

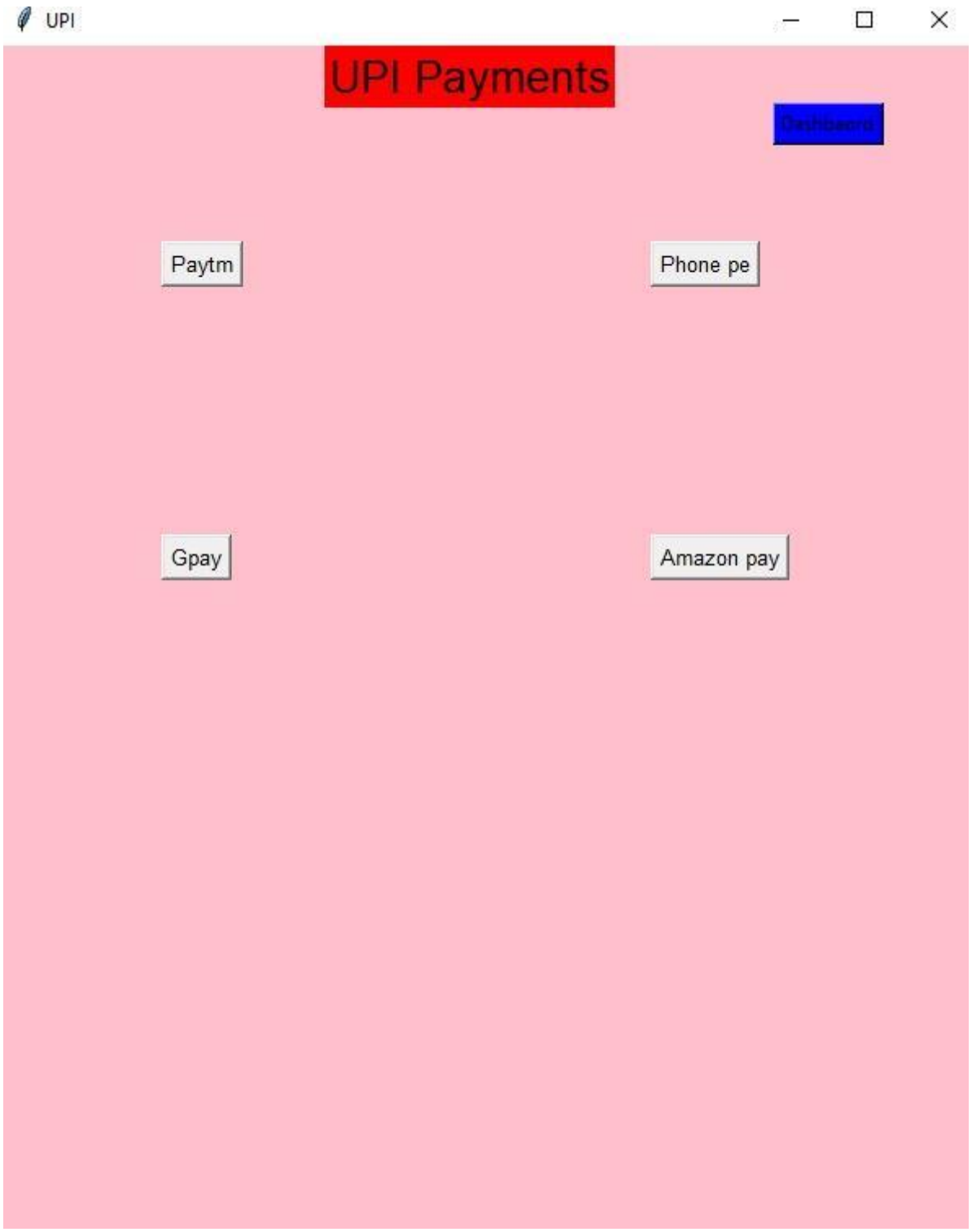
After u specify a bank you will redirect to a page where u have to give your id and password for net banking and click on pay



->UPI PAYMENT

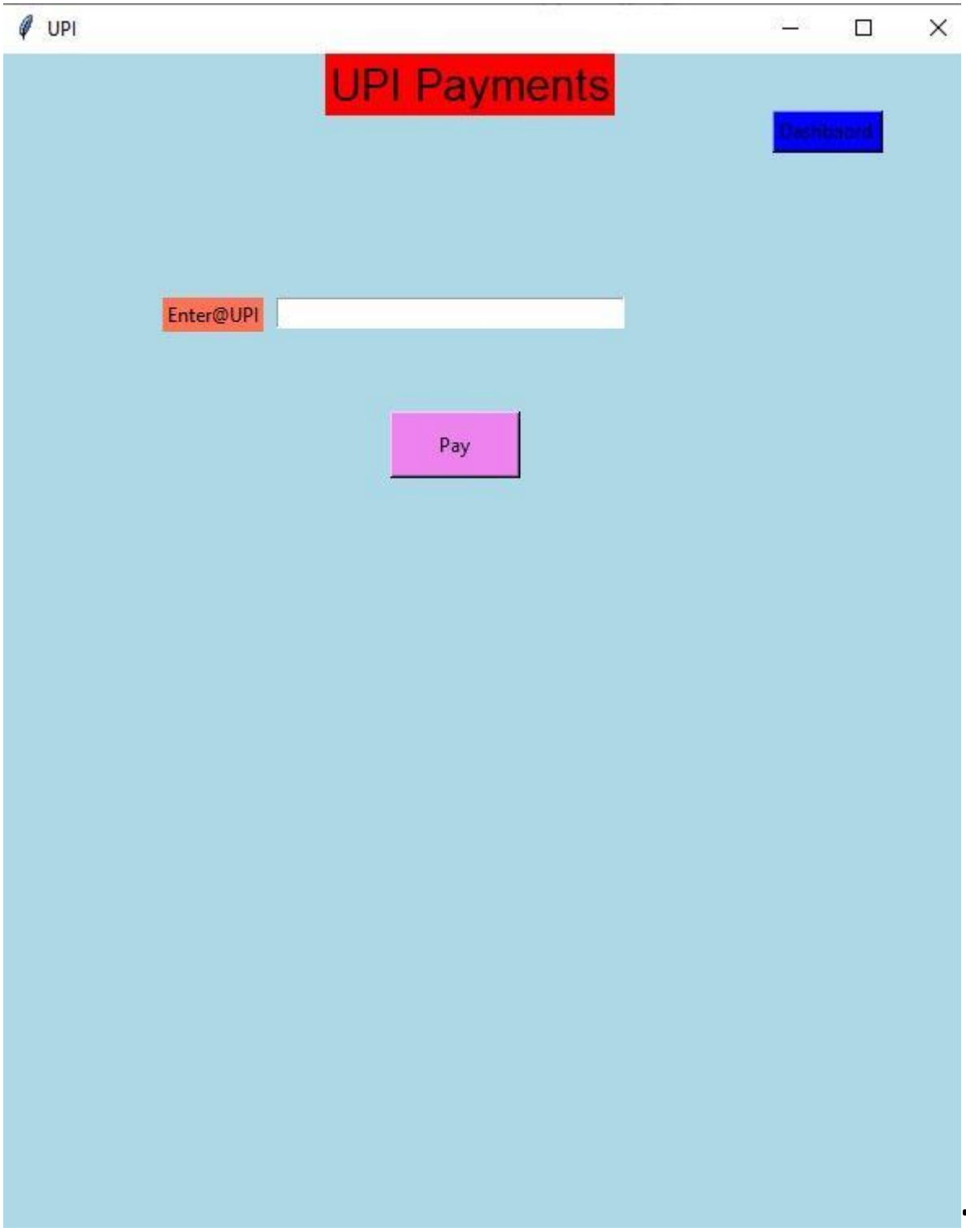
If u select this payment option, u will have different UPI options to pay like

- Paytm
- Gpay
- Phone Pay
- Amazon Pay



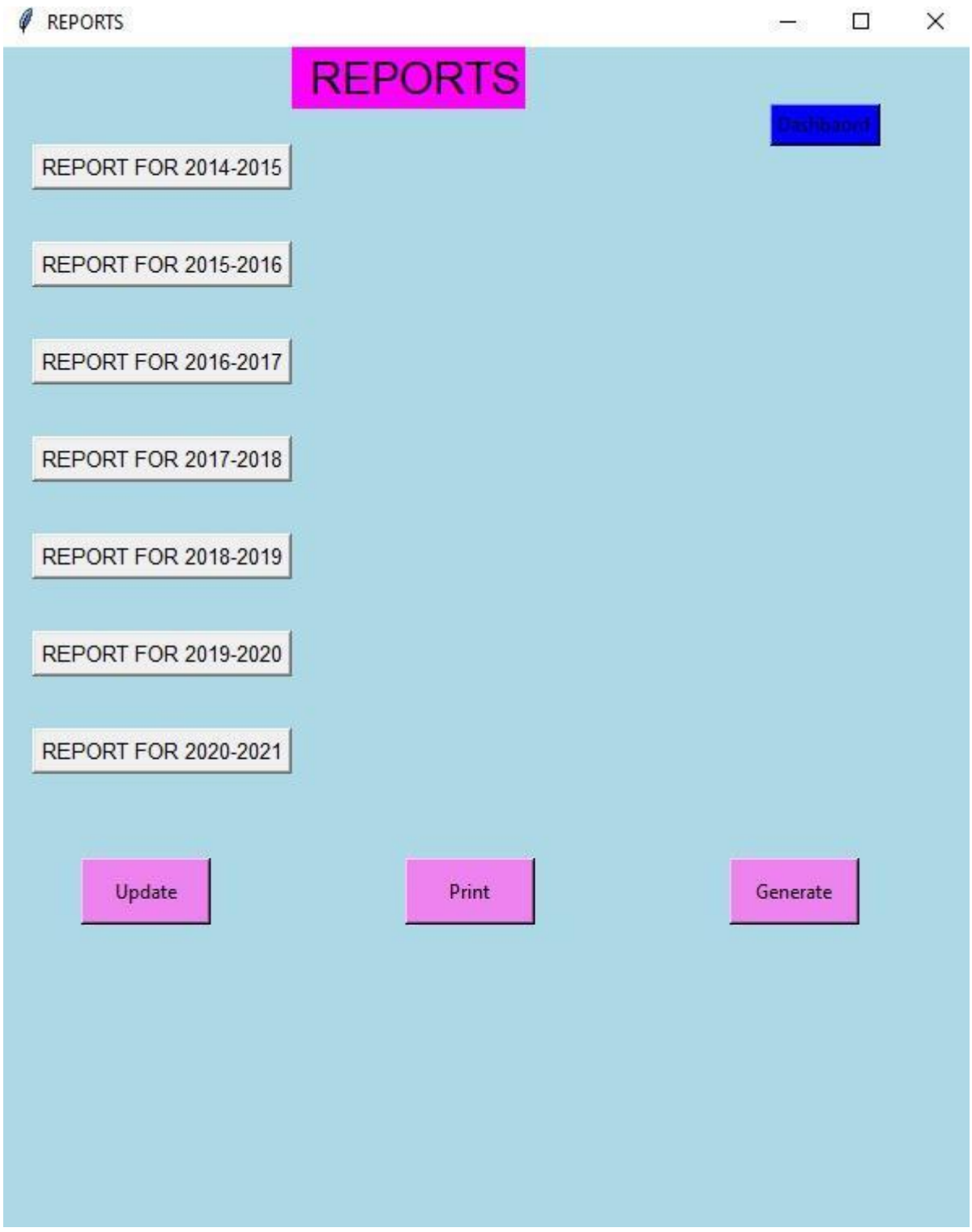
->UPI

After u select any one of UPI methods an other new window will appear where you have to give your UPI id and then click on pay



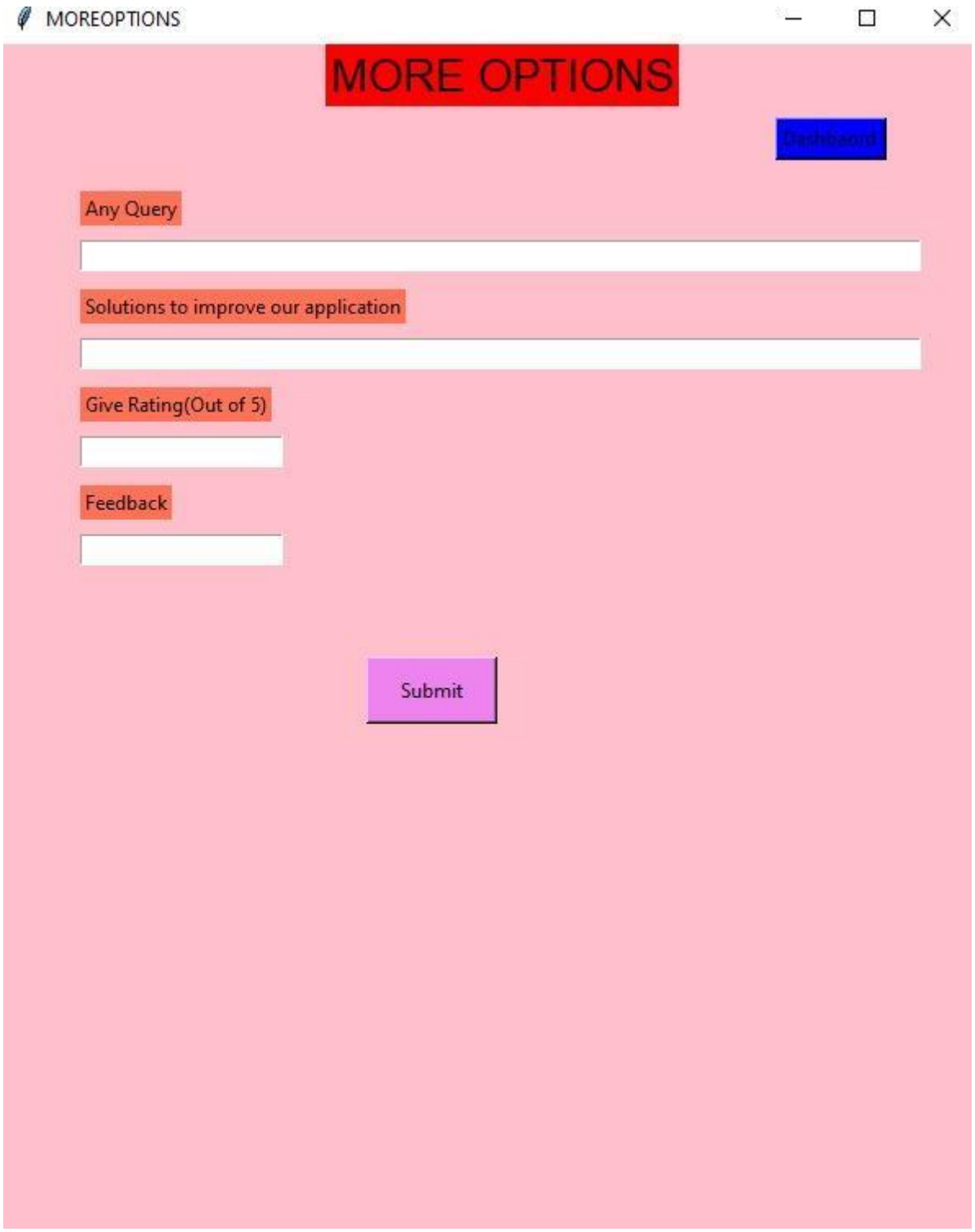
>REPORTS

If u select the reports option, you will get the previous year bookings list



->MORE OPTIONS

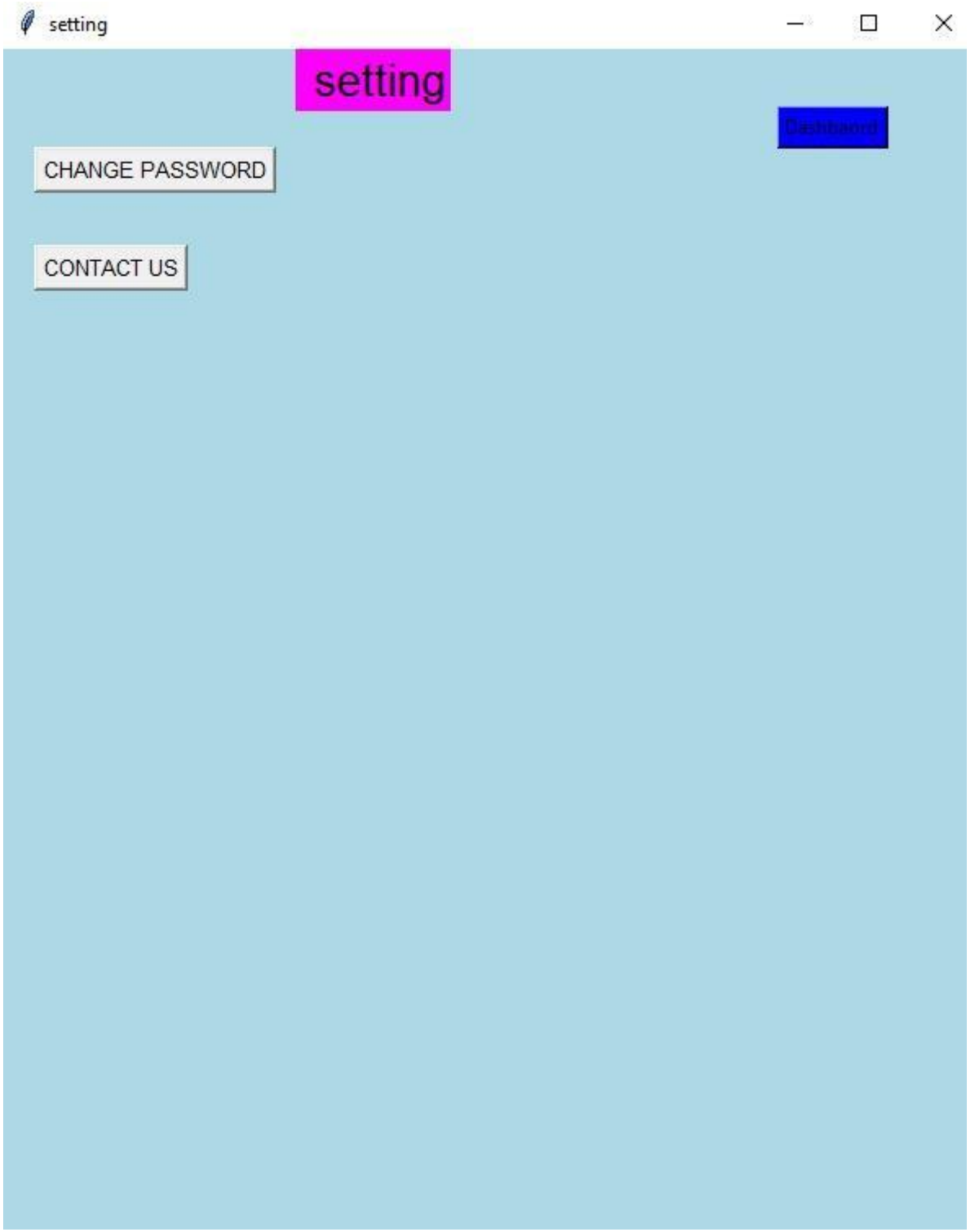
If you have any issue, you can click on the more options button and give your feedback.



->SETTINGS

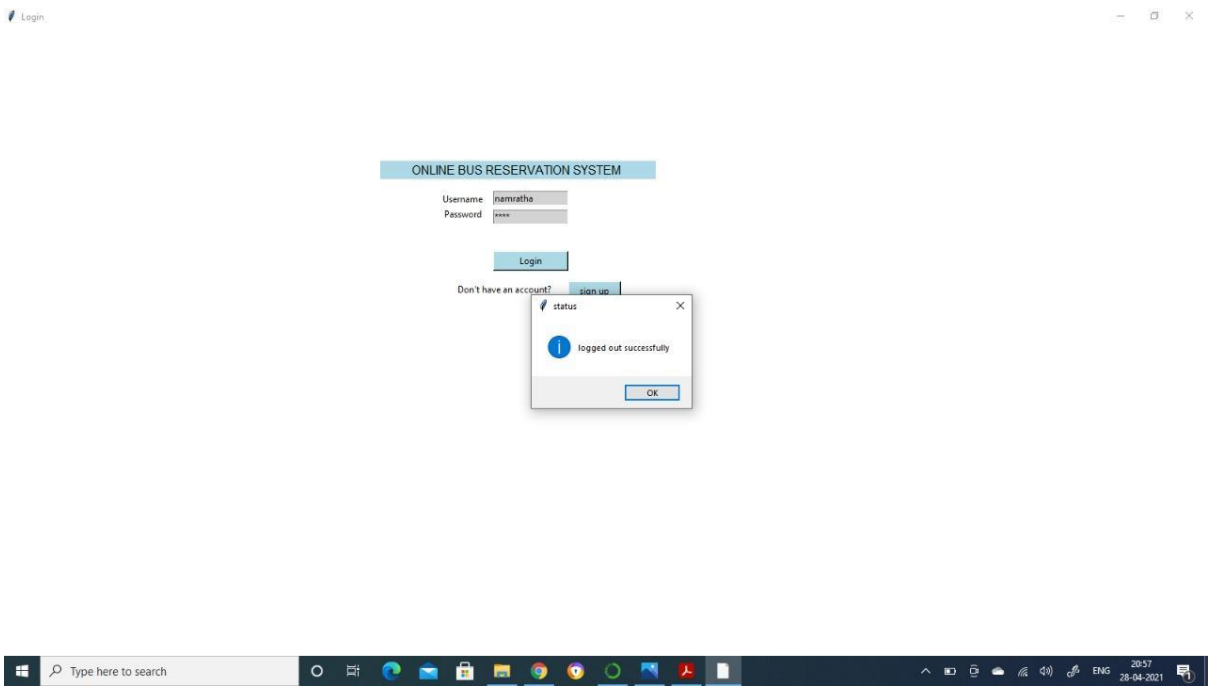
If u click on the settings button, you will have two options

- change password
- contact us



->LOGOUT

After you click on the log out button a message box will pop up mentioning you are successfully logged out



CODING MODULE – 3

Code:-

```
from tkinter import *
from tkinter import messagebox
import os
import pyqrcode
window=Tk()

window.title("Login")
window.geometry("1450x750")
window.config(bg="light green")
def con(a):
    a.config(bg="light green")
def b(j):
    j.config(bg="light grey")

login=Label(window,text="Online Bus Reservation Management",font=(10),width=40)
login.place(x=525,y=180)
login.config(bg="Violet")

user=Label(window,text="Username")
user.place(x=560,y=220,width=100)
con(user)
```

```
pw=Label(window,text="Password")
pw.place(x=560,y=240,width=100)
con(pw)

u=Entry(window,width=50)
u.place(x=650,y=220,width=100)
b(u)
p=Entry(window,show="*",width=50)
p.place(x=650,y=245,width=100)
b(p)

def check():
    user=u.get()
    passwd=p.get()
    if(user == "DineshReddy" and passwd == "DineshReddy"):
        dashboard()
    else:
        msg=messagebox.showinfo("error","Entered username or password is wrong!")

def IN():
    window1=Tk()
    def back():
        window1.destroy()
        dashboard()
        window1.geometry("600x730")
        window1.config(bg="pink")
        window1.title("QRCode")

dash=Button(window1,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    def generate():
        if len(Subject.get())!=0 :
            global qr,photo
            qr = pyqrcode.create(Subject.get())
            photo = BitmapImage(data = qr.xbm(scale=8))
        else:
            messagebox.showinfo("Please Enter some Subject")
        try:
            showcode()
        except:
            pass

    def showcode():
        imageLabel.config(image = photo)
        subLabel.config(text="QR of " + Subject.get())

    def save():
        dir = os.getcwd() + "\\QR Codes"
        if not os.path.exists(dir):
            os.makedirs(dir)
```

```
        try:
            if len(name.get())!=0:
                qr.png(os.path.join(dir,name.get()+".png"),scale=8)
            else:
                messagebox.showinfo("Please enter a File Name")
        except:
                messagebox.showinfo("Generate the QR code first!")

        Sub = Label(window1,text="Enter subject")
        Sub.grid(row =0,column =0,sticky=N+S+W+E)

        FName = Label(window1,text="Enter FileName")
        FName.grid(row =1,column =0,sticky=N+S+W+E)

        Subject = StringVar()
        SubEntry = Entry(window1,textvariable = Subject)
        SubEntry.grid(row =0,column =1,sticky=N+S+W+E)

        name = StringVar()
        nameEntry = Entry(window1,textvariable = name)
        nameEntry.grid(row =1,column =1,sticky=N+S+W+E)

        button = Button(window1,text = "Generate",width=15,command = generate)
        button.grid(row =0,column =3,sticky=N+S+W+E)

        imageLabel = Label(window1)
        imageLabel.grid(row =2,column =1,sticky=N+S+W+E)

        subLabel = Label(window1,text="")
        subLabel.grid(row =3,column =1,sticky=N+S+W+E)

        saveB = Button(window1,text="Save as PNG",width=15,command = save)
        saveB.grid(row =1,column =3,sticky=N+S+W+E)
        Rows = 3
        Columns = 3

        for row in range(Rows+1):
            window1.grid_rowconfigure(row,weight=1)

        for col in range(Columns+1):
            window1.grid_columnconfigure(col,weight=1)
        window1.mainloop()

def UPIid():
    UPIid=Tk()
    def back():
        UPIid.destroy()
        dashboard()
    UPIid.geometry("600x730")
    UPIid.config(bg="pink")
```

```
    UPIid.title("UPI")
    dash=Button(UPIid,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(UPIid,text="UPI Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    UPI1=Label(UPIid,text="Enter@UPI",bg="coral1").place(x=100,y=150)
    t1=Entry(UPIid,width=35).place(x=170,y=150)
    Pay=Button(UPIid,text="Pay",height=2,width=10,bg="violet").place(x=240,y=220)
    UPIid.mainloop()

def Sourcestation():
    Sourcestation=Tk()
    def back():
        Sourcestation.destroy()
        dashboard()
    Sourcestation.geometry("600x650")
    Sourcestation.config(bg="yellow")
    Sourcestation.title("Sourcestation")

dash=Button(Sourcestation,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)

ti=Label(Sourcestation,text="Sourcestation",font=("bold",20),bg="magenta").place(x=180,y=
0)
    a=["Anna Nagar East","Anna Nagar Tower","Alandur","Arumbakkam","Ashok
Nagar","Chennai International Airport","Ekkattuthangal","Government
Estate","Guindy","High Court","Kaladipet","Kilpauk","Koyambedu","St. Thomas
Mount","Teynampet","Thirumangalam","Tiruvottriyur","Tondiarpet","Vadapalani","Washer
manpet","Wimco Nagar"]
    c=StringVar(Sourcestation)
    droplist=OptionMenu(Sourcestation,c, *a)
    droplist.config(width=12)
    c.set("Sourcestation")
    droplist.place(x=20,y=100)
    Sourcestation.mainloop()

def Destinationstation():
    Destinationstation=Tk()
    def back():
        Destinationstation.destroy()
        dashboard()
    Destinationstation.geometry("600x650")
    Destinationstation.config(bg="yellow")
    Destinationstation.title("Destinationstation")

dash=Button(Destinationstation,text="Dashbaord",bg="blue",command=back).place(x=475,y
=35)

ti=Label(Destinationstation,text="Destinationstation",font=("bold",20),bg="magenta").place(
x=180,y=0)
    a=["Anna Nagar East","Anna Nagar Tower","Alandur","Arumbakkam","Ashok
Nagar","Chennai International Airport","Ekkattuthangal","Government
Estate","Guindy","High Court","Kaladipet","Kilpauk","Koyambedu","St. Thomas
```

```
Mount","Teynampet","Thirumangalam","Tiruvottriyur","Tondiarpet","Vadapalani","Washer
manpet","Wimco Nagar"]
    c=StringVar(Destinationstation)
    droplist=OptionMenu(Destinationstation,c, *a)
    droplist.config(width=15)
    c.set("Destinationstation")
    droplist.place(x=20,y=100)
    Destinationstation.mainloop()

def transaction():
    transaction=Tk()
    transaction.title("Net BAnking Payment")
    transaction.geometry("600x740")
    transaction.config(bg="light green")
    def con(a):
        a.config(bg="light green")
    def b(j):
        j.config(bg="light grey")

    login=Label(transaction,text="Net Banking Payment",font=(10),width=30)
    login.place(x=50,y=180)
    login.config(bg="Violet")

    user=Label(transaction,text="Username")
    user.place(x=50,y=220,width=100)
    con(user)

    pw=Label(transaction,text="Password")
    pw.place(x=50,y=240,width=100)
    con(pw)

    u=Entry(transaction,width=50)
    u.place(x=150,y=220,width=100)
    b(u)
    p=Entry(transaction,show="*",width=50)
    p.place(x=150,y=245,width=100)
    b(p)
    Pay=Button(transaction,text="Pay",height=2,width=10,bg="violet").place(x=150,y=300)
    transaction.mainloop()

def MOREOPTIONS():
    MOREOPTIONS=Tk()
    def back():
        MOREOPTIONS.destroy()
        dashboard()
    MOREOPTIONS.geometry("600x730")
    MOREOPTIONS.config(bg="pink")
    MOREOPTIONS.title("MOREOPTIONS")
```

```
dash=Button(MOREOPTIONS,text="Dashbaord",bg="blue",command=back).place(x=475,y=45)
   ti=Label(MOREOPTIONS,text="MORE OPTIONS",font=("bold",20),bg="Red").place(x=200,y=0)

   MOREOPTIONS1=Label(MOREOPTIONS,text="Any Query",bg="coral1").place(x=50,y=90)
   t1=Entry(MOREOPTIONS,width=85).place(x=50,y=120)
   MOREOPTIONS2=Label(MOREOPTIONS,text="Solutions to improve our application",bg="coral1").place(x=50,y=150)
   t2=Entry(MOREOPTIONS,width=85).place(x=50,y=180)
   MOREOPTIONS3=Label(MOREOPTIONS,text="Give Rating(Out of 5)",bg="coral1").place(x=50,y=210)
   t3=Entry(MOREOPTIONS,width=20).place(x=50,y=240)

MOREOPTIONS3=Label(MOREOPTIONS,text="Feedback",bg="coral1").place(x=50,y=270)
   t3=Entry(MOREOPTIONS,width=20).place(x=50,y=300)

Submit=Button(MOREOPTIONS,text="Submit",height=2,width=10,bg="violet").place(x=225,y=375)
   MOREOPTIONS.mainloop()

def Credit():
   Credit=Tk()
   def back():
      Credit.destroy()
      dashboard()
   Credit.geometry("600x730")
   Credit.config(bg="pink")
   Credit.title("Credit card")
   dash=Button(Credit,text="Dashbaord",bg="blue",command=back).place(x=475,y=45)
   ti=Label(Credit,text="Credit card Payments",font=("bold",20),bg="Red").place(x=200,y=0)

   Credit1=Label(Credit,text="Card Number",bg="coral1").place(x=50,y=90)
   t1=Entry(Credit,width=40).place(x=50,y=120)
   Credit2=Label(Credit,text="Expiry",bg="coral1").place(x=50,y=150)
   t2=Entry(Credit,width=15).place(x=50,y=180)
   Credit3=Label(Credit,text="CVV",bg="coral1").place(x=200,y=150)
   t3=Entry(Credit,width=10).place(x=200,y=181)
   Credit4=Label(Credit,text="Name on card",bg="coral1").place(x=50,y=210)
   t4=Entry(Credit,width=40).place(x=50,y=240)
   Pay=Button(Credit,text="Pay",height=2,width=10,bg="violet").place(x=240,y=300)

   Credit.mainloop()

def Debit():
   Debit=Tk()
```

```
    def back():
        Debit.destroy()
        dashboard()
    Debit.geometry("600x730")
    Debit.config(bg="pink")
    Debit.title("Debit card")
    dash=Button(Debit,text="Dashbaord",bg="blue",command=back).place(x=475,y=45)
    ti=Label(Debit,text="Debit card Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    Debit1=Label(Debit,text="Card Number",bg="coral1").place(x=50,y=90)
    t1=Entry(Debit,width=40).place(x=50,y=120)
    Debit2=Label(Debit,text="Expiry",bg="coral1").place(x=50,y=150)
    t2=Entry(Debit,width=15).place(x=50,y=180)
    Debit3=Label(Debit,text="CVV",bg="coral1").place(x=200,y=150)
    t3=Entry(Debit,width=10).place(x=200,y=181)
    Debit4=Label(Debit,text="Name on card",bg="coral1").place(x=50,y=210)
    t4=Entry(Debit,width=40).place(x=50,y=240)
    Pay=Button(Debit,text="Pay",height=2,width=10,bg="violet").place(x=240,y=300)
    Debit.mainloop()

def Banks():
    Banks=Tk()
    def back():
        Banks.destroy()
        dashboard()
    Banks.geometry("600x730")
    Banks.config(bg="blue")
    Banks.title("PAYMENTS")
    dash=Button(Banks,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(Banks,text="Banks",font=("bold",20),bg="magenta").place(x=180,y=0)
    Bank1=Button(Banks,text="State Bank of
India",font=("bold",10),command=transaction).place(x=20,y=120)
    Bank2=Button(Banks,text="City Union
Bank",font=("bold",10),command=transaction).place(x=20,y=180)
    Bank3=Button(Banks,text="Andhra
Bank",font=("bold",10),command=transaction).place(x=20,y=240)
    Bank4=Button(Banks,text="Canara
Bank",font=("bold",10),command=transaction).place(x=20,y=300)
    Bank5=Button(Banks,text="Axis
Bank",font=("bold",10),command=transaction).place(x=20,y=360)
    Bank6=Button(Banks,text="Syndicate
Bank",font=("bold",10),command=transaction).place(x=20,y=420)
    Bank7=Button(Banks,text="ICICI
Bank",font=("bold",10),command=transaction).place(x=20,y=480)
    Bank8=Button(Banks,text="HDFC
Bank",font=("bold",10),command=transaction).place(x=20,y=540)
    Bank9=Button(Banks,text="CITI
Bank",font=("bold",10),command=transaction).place(x=20,y=600)
    Banks.mainloop()

def UPI():
```

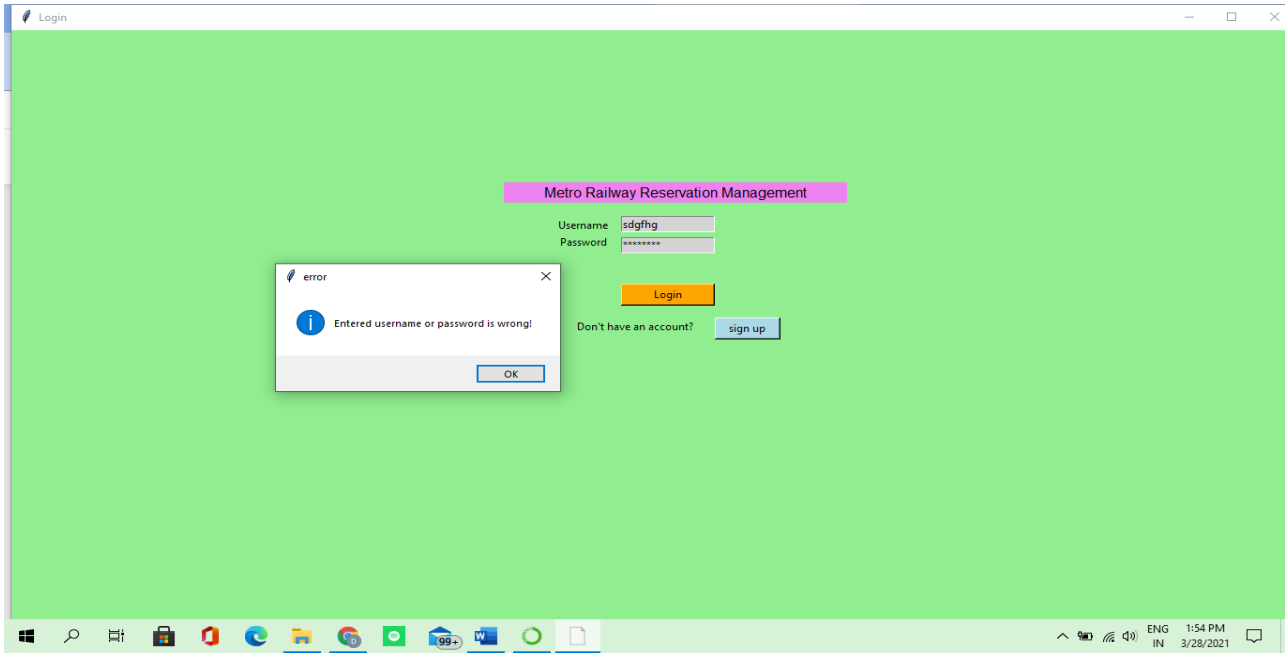
```
    UPI=Tk()
    def back():
        UPI.destroy()
        dashboard()
    UPI.geometry("600x730")
    UPI.config(bg="pink")
    UPI.title("UPI")
    dash=Button(UPI,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(UPI,text="UPI Payments",font=("bold",20),bg="Red").place(x=200,y=0)
    UPI1=Button(UPI,text="Paytm",font=("bold",10),command=UPIid).place(x=100,y=120)
    UPI2=Button(UPI,text="Gpay",font=("bold",10),command=UPIid).place(x=100,y=300)
    UPI3=Button(UPI,text="Phone
pe",font=("bold",10),command=UPIid).place(x=400,y=120)
    UPI4=Button(UPI,text="Amazon
pay",font=("bold",10),command=UPIid).place(x=400,y=300)
    UPI.mainloop()

def payments():
    payments=Tk()
    def back():
        payments.destroy()
        dashboard()
    payments.geometry("600x730")
    payments.config(bg="blue")
    payments.title("PAYMENTS")
    dash=Button(payments,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(payments,text=" Payments",font=("bold",20),bg="magenta").place(x=180,y=0)
    pay1=Button(payments,text="Debit
card",font=("bold",10),command=Debit).place(x=20,y=120)
    pay2=Button(payments,text="Credit
card",font=("bold",10),command=Credit).place(x=20,y=180)
    pay3=Button(payments,text="Net
Banking",font=("bold",10),command=Banks).place(x=20,y=240)
    pay4=Button(payments,text="UPI
payments",font=("bold",10),command=UPI).place(x=20,y=300)

    payments.mainloop()

def reports():
    rep=Tk()
    def back():
        rep.destroy()
        dashboard()
    rep.geometry("600x730")
    rep.config(bg="yellow")
    rep.title("REPORTS")
    dash=Button(rep,text="Dashbaord",bg="blue",command=back).place(x=475,y=35)
    ti=Label(rep,text=" REPORTS",font=("bold",20),bg="magenta").place(x=180,y=0)
    year1=Button(rep,text="REPORT FOR 2014-2015",font=("bold",10)).place(x=20,y=60)
    year2=Button(rep,text="REPORT FOR 2015-2016",font=("bold",10)).place(x=20,y=120)
```

```
    year3=Button(rep,text="REPORT FOR 2016-2017",font=("bold",10)).place(x=20,y=180)
    year4=Button(rep,text="REPORT FOR 2017-2018",font=("bold",10)).place(x=20,y=240)
    year5=Button(rep,text="REPORT FOR 2018-2019",font=("bold",10)).place(x=20,y=300)
    year6=Button(rep,text="REPORT FOR 2019-2020",font=("bold",10)).place(x=20,y=360)
    year7=Button(rep,text="REPORT FOR 2020-2021",font=("bold",10)).place(x=20,y=420)
    update=Button(rep,text="Update",height=2,width=10,bg="violet").place(x=50,y=500)
    prin=Button(rep,text="Print",height=2,width=10,bg="violet").place(x=250,y=500)
    generate=Button(rep,text="Generate",height=2,width=10,bg="violet").place(x=450,y=500)
    rep.mainloop()


def dashboard():
    window1=Tk()
    window1.geometry("1450x750")
    window1.title("Dashbaord")
    def quit():
        window1.destroy()
        messagebox.showinfo("status","logged out successfully")
    window1.config(bg="thistle2")
    name=Label(window1,text="Metro Railway Reservation
MANAGEMENT",font=(15),width=40,bg="blue",fg="pink").place(x=550,y=10)

home=Button(window1,text="Settings",font=(6),width=10,height=1,bg="maroon1").place(x
=20,y=10)
    quit=Button(window1,text="Log out",font=(6),width=10,height=1,bg="light
grey",command=quit).place(x=1275,y=10)
    sd=Button(window1,text="Source station",font=(10),width=20,height=3,bg="light
green",command=Sourcestation).place(x=60,y=150)
    bill=Button(window1,text="Destination
station",font=(10),width=20,height=3,bg="silver",fg="red",command=Destinationstation).pla
ce(x=600,y=150)
    cusdb=Button(window1,text="Payments",font=(10),width=20,height=3,bg="dark olive
green3",command=payments).place(x=1200,y=150)

invo=Button(window1,text="INVOICES",font=(10),width=20,height=3,bg="gold",command
=IN).place(x=60,y=500)
    rep=Button(window1,text="REPORTS",font=(10),width=20,height=3,bg="sky
blue",command=reports).place(x=600,y=500)

more=Button(window1,text="MORE\nOPTIONS",font=(10),width=20,height=3,bg="sienna
2",command=MOREOPTIONS).place(x=1200,y=500)
    window1.mainloop()

log=Button(text="Login",width=100,command=check)
log.place(x=650,y=300,width=100)
log.config(bg="orange")

don=Label(window,text="Don't have an account?")
don.place(x=600,y=340,width=130)
con(don)
```

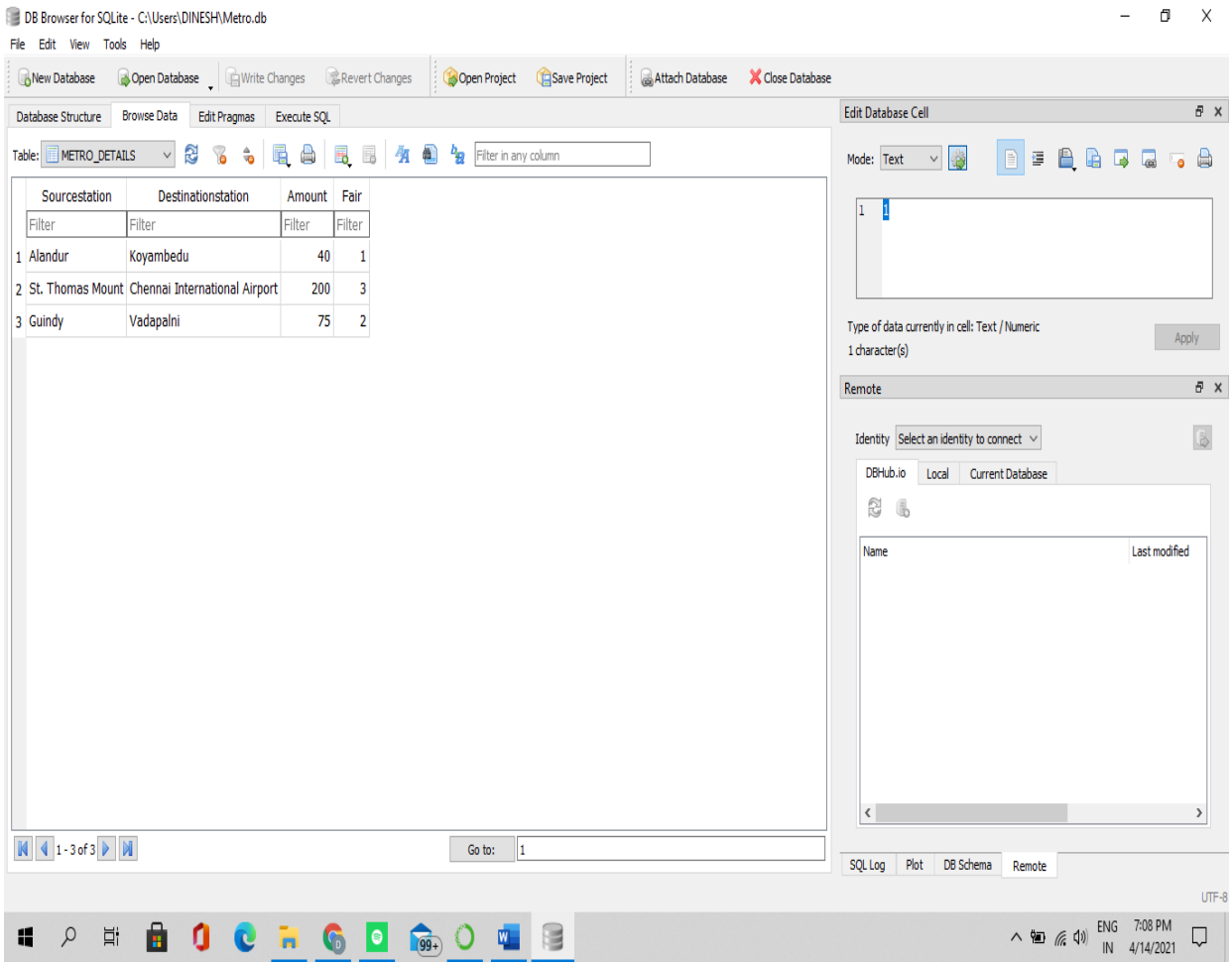
```
def sign_up():

    supage=Tk()
    supage.title("Register")
    supage.geometry("600x600")
    supage.config(bg="mistyrose2")

    head=Label(supage,text="Register                                                                    your
account",bg="red",font=("bold",18)).grid(column=1,row=0,columnspan=3,padx=10,pady=10
)

    fn=Label(supage,text="First name     ").grid(column=0,row=1,padx=10,pady=10)
    t2=Entry(supage,width=30,bg="white").grid(column=1,row=1,padx=10,pady=10)

    ln=Label(supage,text="Last name      ").grid(column=0,row=2,padx=10,pady=10)
    t3=Entry(supage,width=30).grid(column=1,row=2,padx=10,pady=10)

    dob=Label(supage,text="Date of Birth")
    dob.grid(column=0,row=3,padx=10,pady=10)
    dates=[
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]

months=["january","febrary","march","april","may","june","july","august","september","octo
ber","november","december"]

years=[1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2
010,2011,2012,2013,2015,2016]

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *dates)
    droplist.config(width=5)
    c.set("date")
    droplist.grid(row=3,column=1,padx=10,pady=10)

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *months)
    droplist.config(width=5)
    c.set("month")
    droplist.grid(row=3,column=2,padx=10,pady=10)

    c=StringVar(supage)
    droplist=OptionMenu(supage,c, *years)
    droplist.config(width=5)
    c.set("year")
    droplist.grid(row=3,column=3,padx=10,pady=10)

    bn=Label(supage,text="Gender        ")
```

```
    bn.grid(column=0,row=4,padx=10,pady=10)
    rad1 = Radiobutton(supage,text='male',value=1).grid(column=1,row=4,padx=10,pady=10)
    rad2 =
Radiobutton(supage,text='female',value=0).grid(column=2,row=4,pady=10,padx=10)

    em=Label(supage,text="Email address    ").grid(column=0,row=5,padx=10,pady=10)
    t6=Entry(supage,width=30).grid(column=1,row=5,padx=10,pady=10)

    mn=Label(supage,text="Mobile number    ").grid(column=0,row=6,padx=10,pady=10)
    t5=Entry(supage,width=30).grid(column=1,row=6,pady=10,padx=10)

    pw=Label(supage,text="Password         ").grid(column=0,row=7,padx=10,pady=10)
    t7=Entry(supage,show="*",width=30).grid(column=1,row=7,padx=10,pady=10)

    cpw=Label(supage,text="Confirm Password ").grid(column=0,row=8,padx=10,pady=10)
    t7=Entry(supage,show="*",width=30).grid(column=1,row=8,padx=10,pady=10)

    st=Label(supage,text="State            ").grid(column=0,row=9,padx=10,pady=10)
    t8=Entry(supage,width=30).grid(column=1,row=9,padx=10,pady=10)

    ctry=Label(supage,text="Country          ").grid(column=0,row=10,padx=10,pady=10)
    t8=Entry(supage,width=30).grid(column=1,row=10,padx=10,pady=10)

    def created():
        supage.destroy()
        msg=messagebox.showinfo("Registered","Completed registration successfully")

submit=Button(supage,text="Submit",font=("bold",10),fg="Black",bg="green3",command=c
reated).grid(row=11,column=1,columnspan=3)

    supage.mainloop()

sp=Button(window,text="sign up",width=60,command=sign_up)
sp.place(x=750,y=340,width=70)
sp.config(bg="light blue")

window.mainloop()
```

If u enter the wrong username and wrong password,it shows the error that the username and password u have entered is wrong.



This is the data base of our project . All the data will be saved here.



NEW REQUIREMENTS/NEW CHNAGES

- We have to configure the reset password button
- Should add a new feature to select the type of bus they want
- To select the seat number in the bus
- Payments should be done
- A website should be made
- A page where total fare of the ticket should be shown
- Should upload the reports of before years

COMPLEXITY AND MASTER TEST REPORT

Types of testing, Methodology, Tools:

Category	Methodology	Tools requirement
Functional requirements	manual	GUI testing, regression testing, backend testing
Non-functional requirements	manual	Testing

Functional test cases:

Test ID	Test scenario	Test case	Execution steps	Expected outcome	Actual outcome	status	remarks
#FT1	Verifying data	Accept valid names and numbers	User needs to enter the data in each module	matched	matched	Pass	success
#FT2	Data stored in sqlite	Verification of stations	Need to save the correct station names by default type	matched	matched	Pass	success

Non-functional test cases:

Test ID	Test scenario	Test case	Execution steps	Expected outcome	Actual outcome	status	remarks
#NFT1	Login credentials	Accept username and password	User needs to enter the data in given entries	Logged in and dashboard will appear	Logged in and dashboard will appear	Pass	success

Functional test cases:

Test ID	Test scenario	Test case	Execution steps	Expected outcome	Actual outcome	status	remarks
#FT1	Verifying data	Accept valid names and numbers	User needs to enter the data in each module	matched	matched	Pass	success
#FT2	Data stored in sqlite	Verification of stations	Need to save the correct station names by default type	matched	matched	Pass	success

Non-functional test cases:

Test ID	Test scenario	Test case	Execution steps	Expected outcome	Actual outcome	status	remarks
#NFT1	Login credentials	Accept username	User needs to enter the	Logged in and dashboard	Logged in and dashboard	Pass	success

		and password	data in given entries	will appear	will appear		

Defect Log:-

Requirement	Defect ID	Defect Description	Assignee	Status
M1R2	Database	Some selected entries of stations are not storing in database	Project Manager	Under working

Test Report:-

Category	Progress	Status
Functional Testing	Completed	Successful
Non Functional Testing	Completed	Successful

Functional	Test case Coverage	Status
Verifying data	100%	Completed
Verifying booked stations	80%	In Progress