By: K. Sanjay Kumar

Travels Management System with Python(3.0v) and PostgreSQL database

Introduction:

Python: This is the language used in this project for writing the script. Python is a powerful general-purpose programming language. It is used in web development, data science, creating software prototypes, and so on. Fortunately for beginners, Python has simple easy-to-use syntax. This makes Python an excellent language to learn to program for beginners.

Tkinter: This is a module used in this project for creating and arranging all the widgets in the final GUI application. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions.

Psycopg2: This is an opensource module used in python to enable two-way communication between python program and the sanjay_travels database. In other words it is a PostgreSQL adaptor for python.

PostgreSQL: PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally name POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley

Data Base: A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS) like Postgres.

However, the real beauty of the project is, its seamless communication with the postgres database in my PC.

By: K. Sanjay Kumar

Even with the whole code file you may get errors because I have created postgres database named sanjay_travels with tables

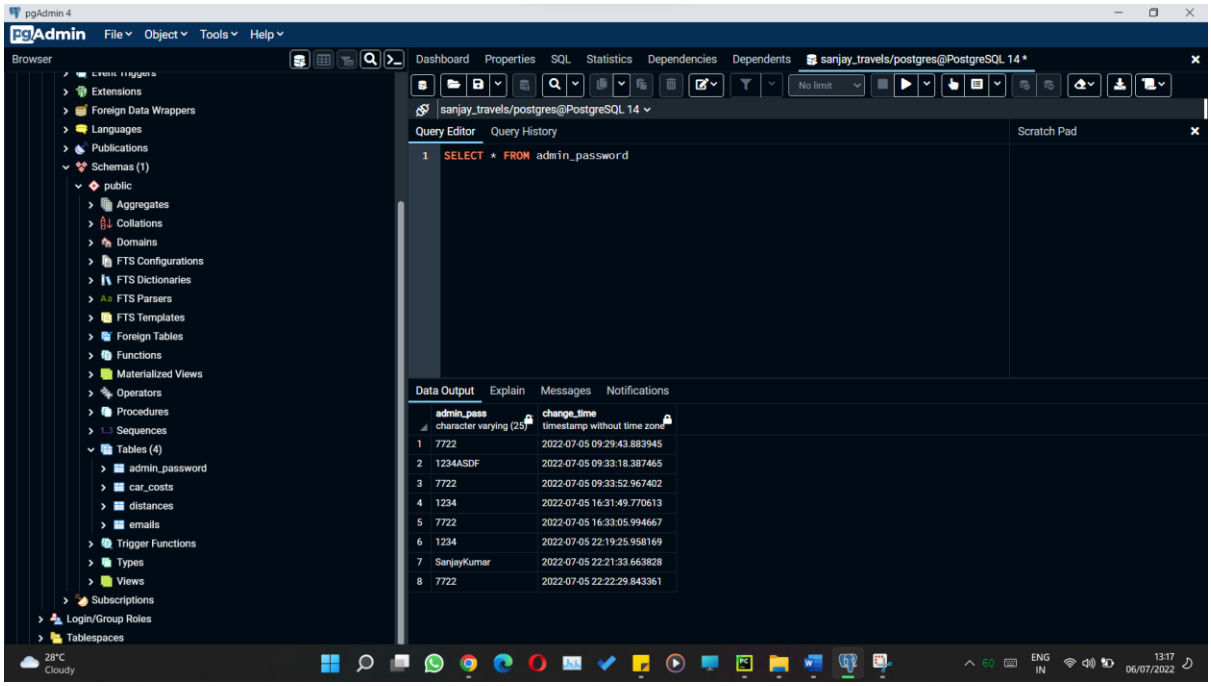
1. admin_password(stores admin passwords and its change history)
2. car_costs(stores cost per km for each type of car)
3. distances(stores destination names with their distances)
4. emails(stores customers emails who accessed contact details)

- If you want to run the code in your pc you must create the server and tables like them and edit your password in code in your pc.
- There are few constraints setup during the creation of the tables which enables the prevention of wrong data entry or duplicate data entry.
- By the use of database in this project we got many features. They will be discussed below.

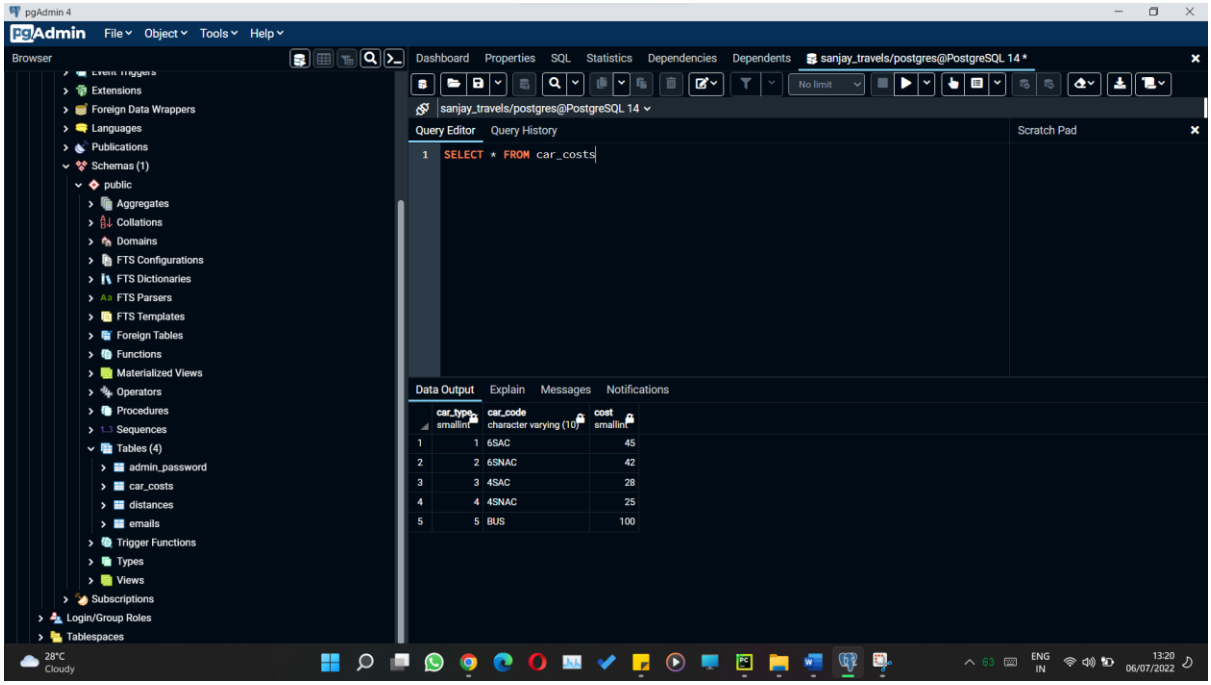
Now let's look into the contents of all the tables so that you can understand the contents of the tables using simple query

"SELECT * FROM table_name"

admin_password table: This table can be updated in the project.

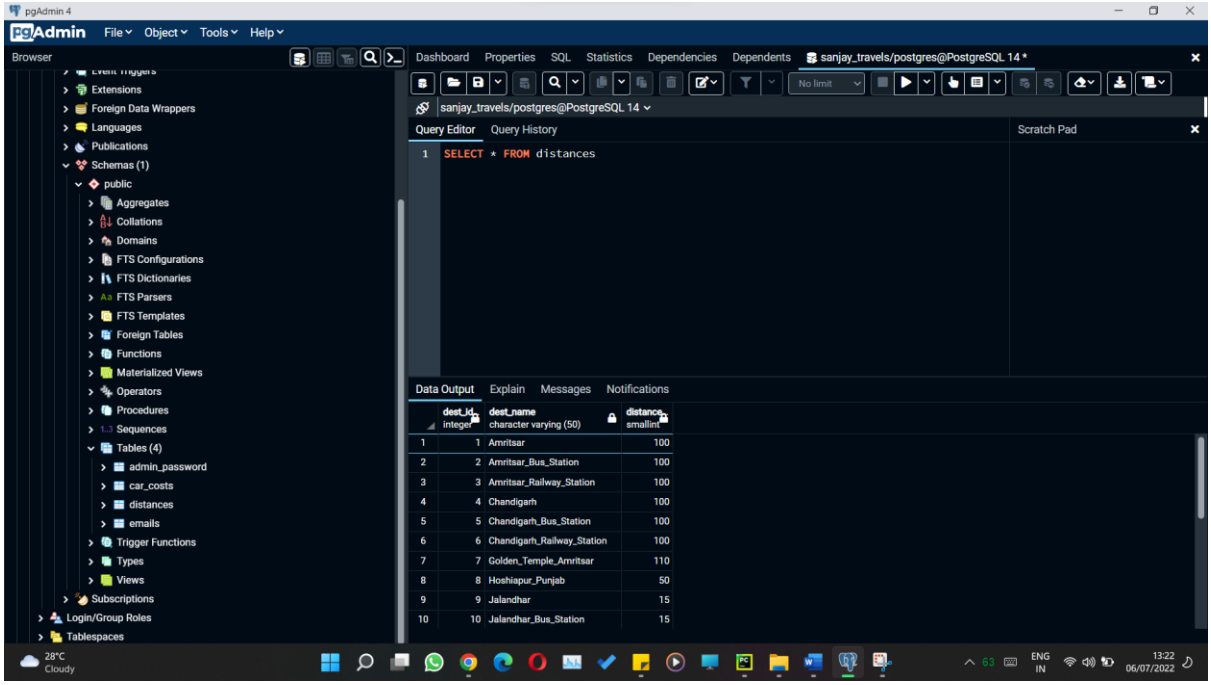


car_costs: This table can be updated in the project.



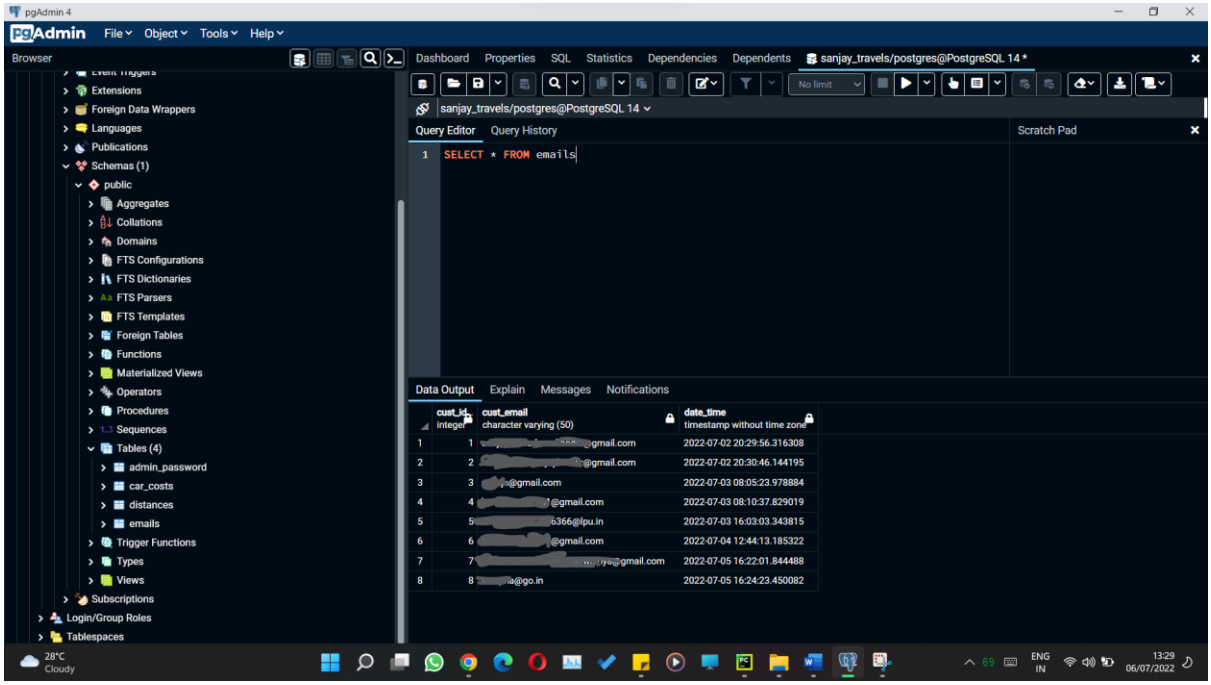
distances:

This table contains major locations near by LOVELY PROFESSIONAL UNIVERSITY, INDIA to visit in addition to that it also has their distances in km. And this table can be updated in the project.



emails:

This table contains the emails of the customers who accessed the contact details. This will update automatically.

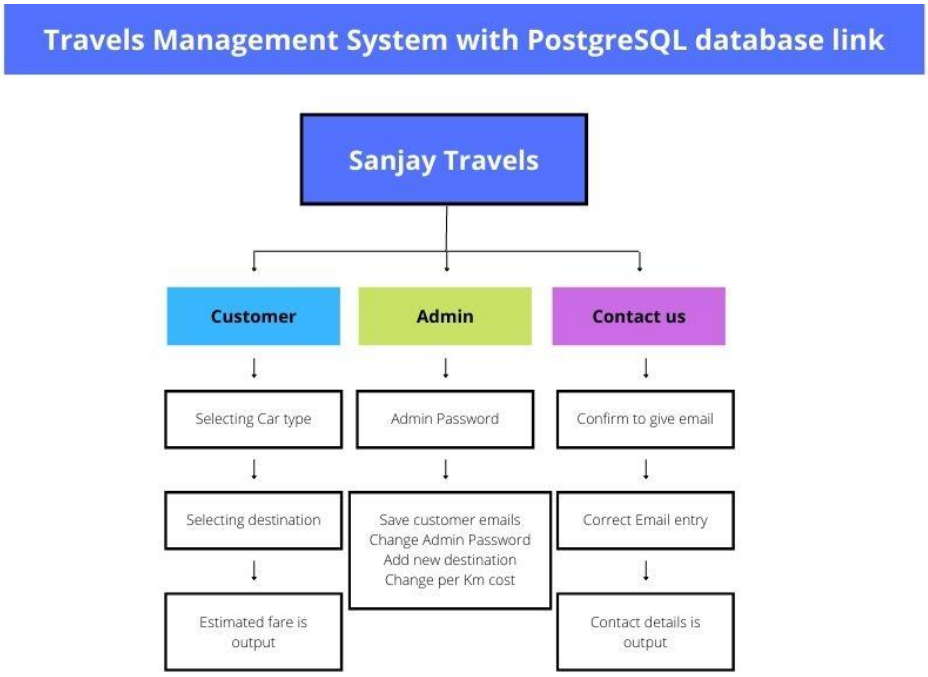


Features:

1. Live time in the Homepage
2. Able to change admin password
3. Able to prevent customers accessing contact info with fake mails.
4. Live Fare calculation in customer section
5. Able to add new destination
6. Able to update costs of vehicles
7. Able to save all customer emails list in a text file
8. Fault tolerant
9. Able to stop wrong data entry.

By: K. Sanjay Kumar

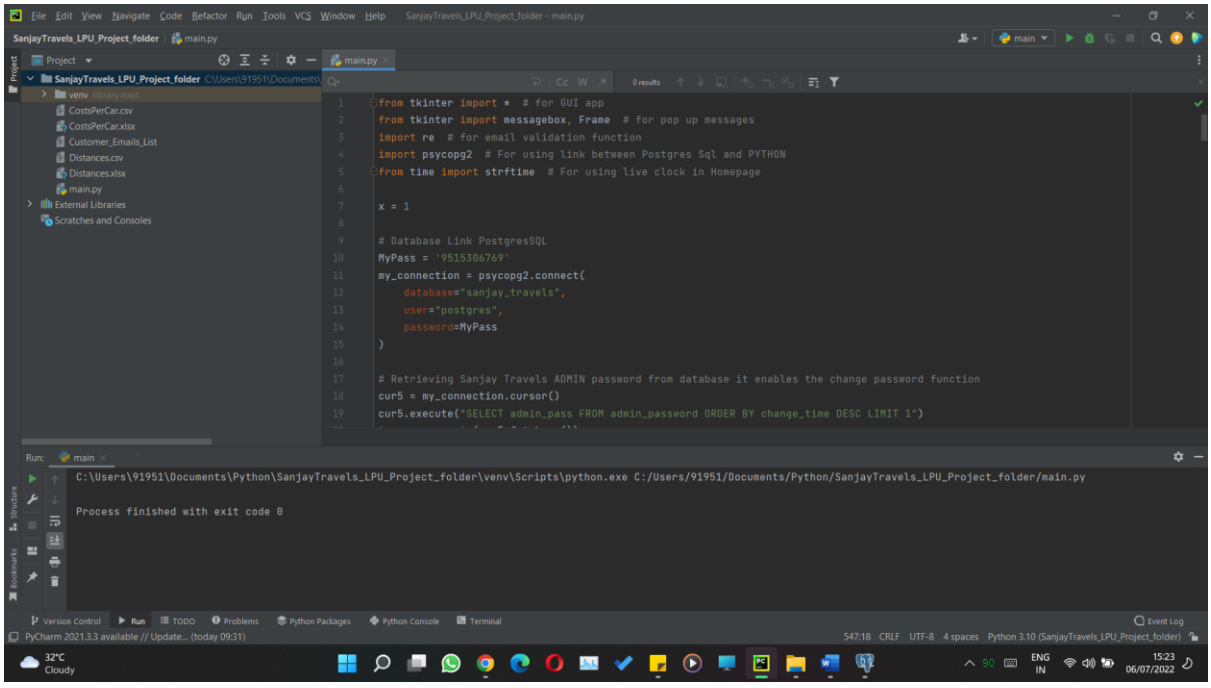
Flow chart:



Code: Code will be placed in the same branch of this file.

Working:

=>Running the code to see home page



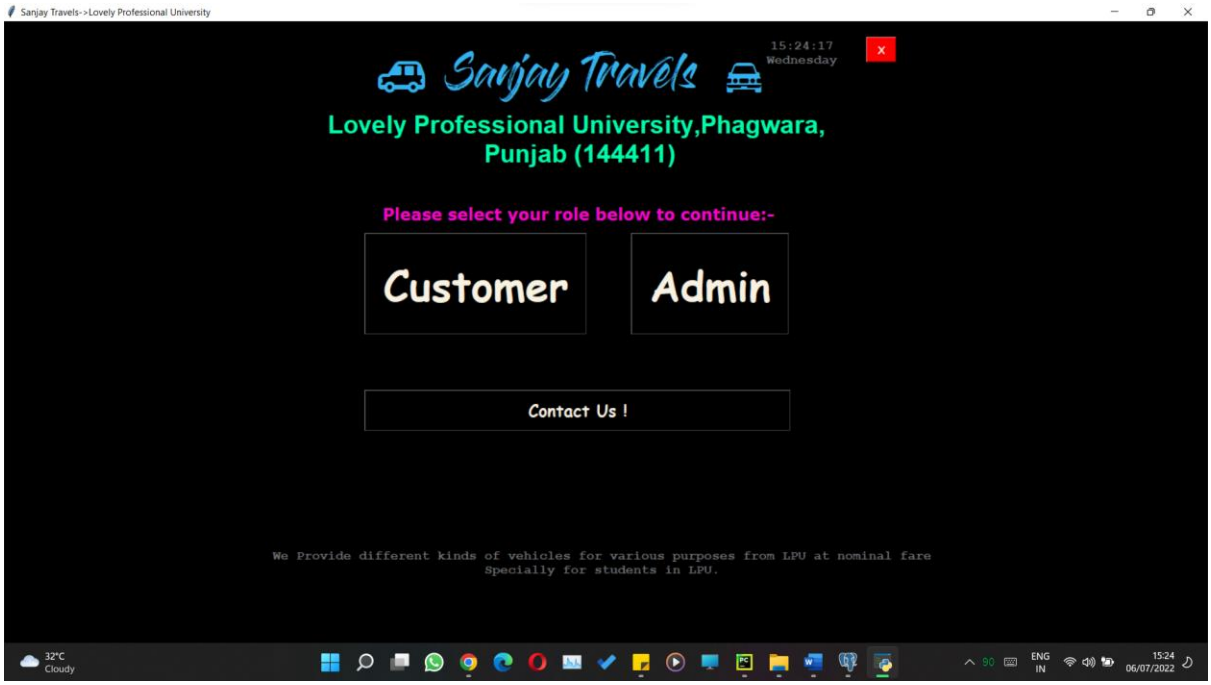
By: K. Sanjay Kumar

HomePage:

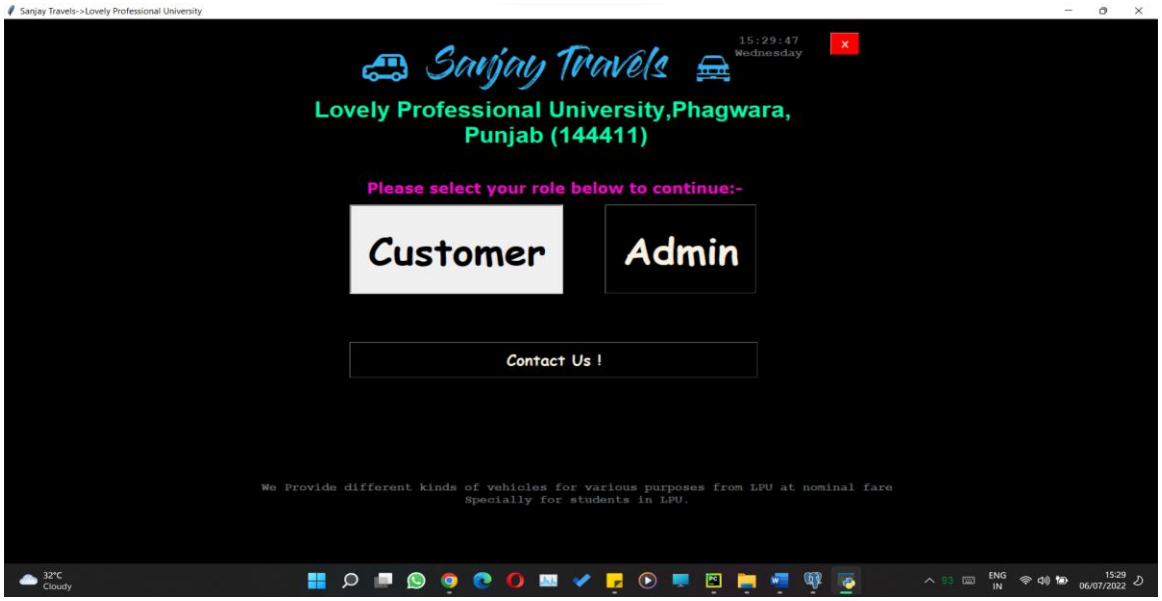
Contents:

1. Live Clock
2. Buttons(Close, Customer, Admin, Contact us)
3. Company name(Sanjay Travels)
4. Address, Few text labels

At any instance, clicking close button will close the widow and exits.

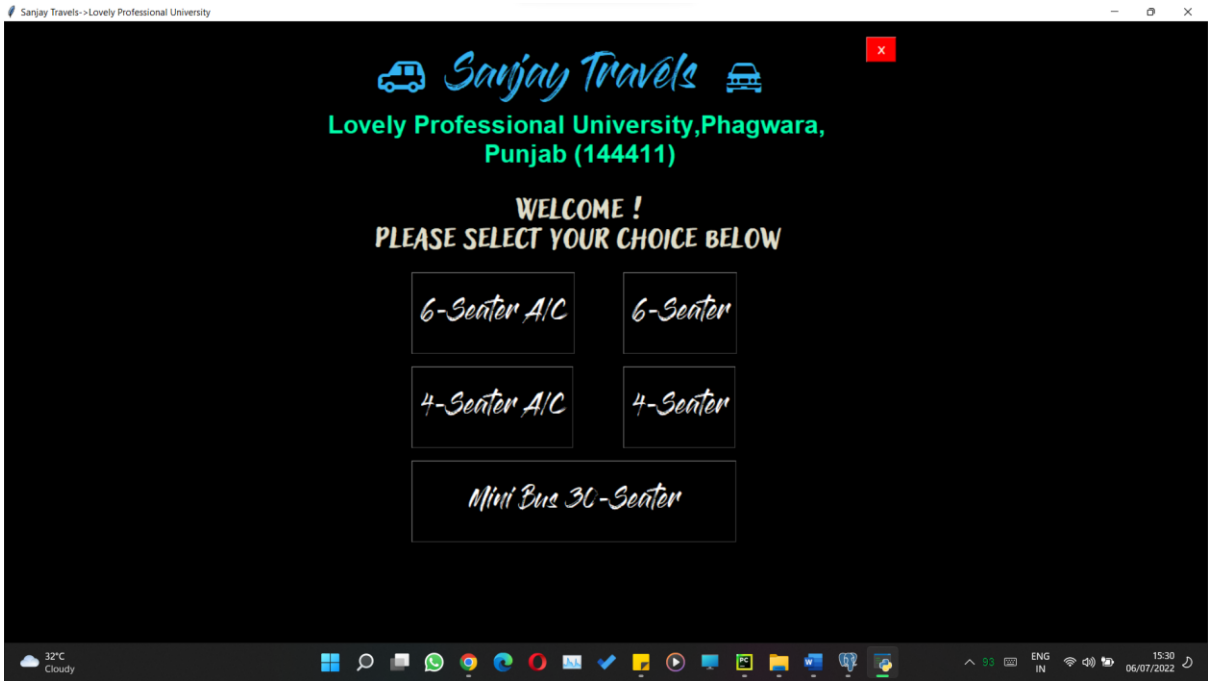


Clicking on customer:

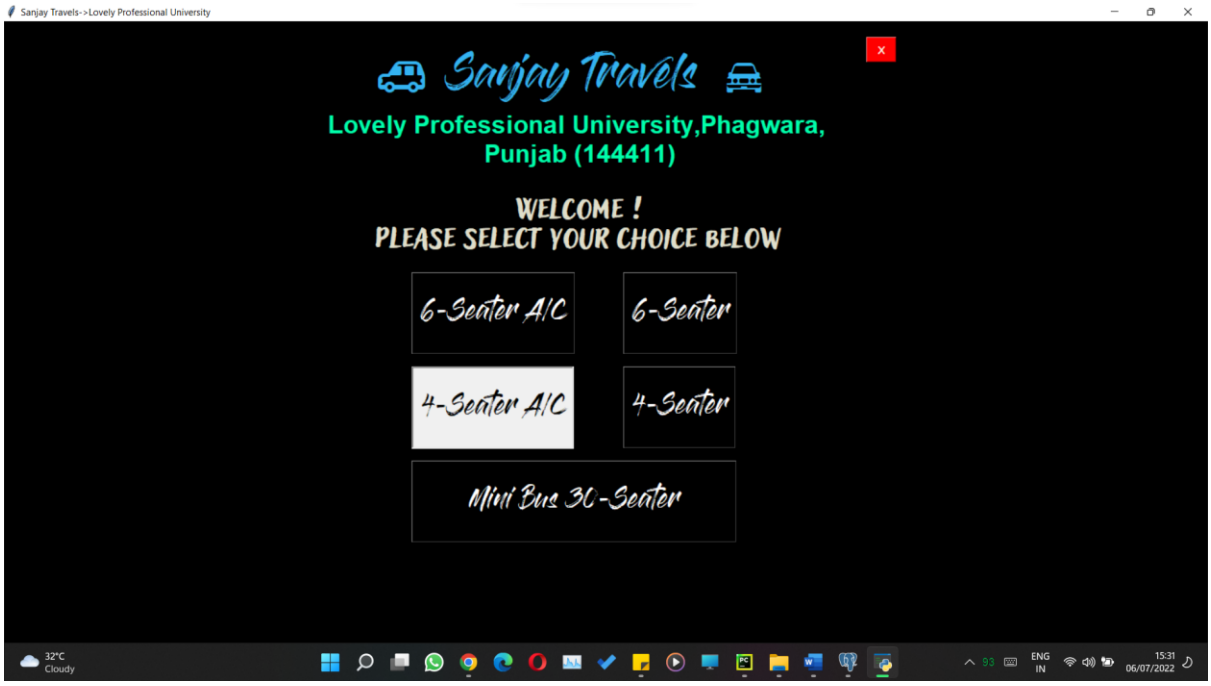


By: K. Sanjay Kumar

Customer Section(Car selection page):



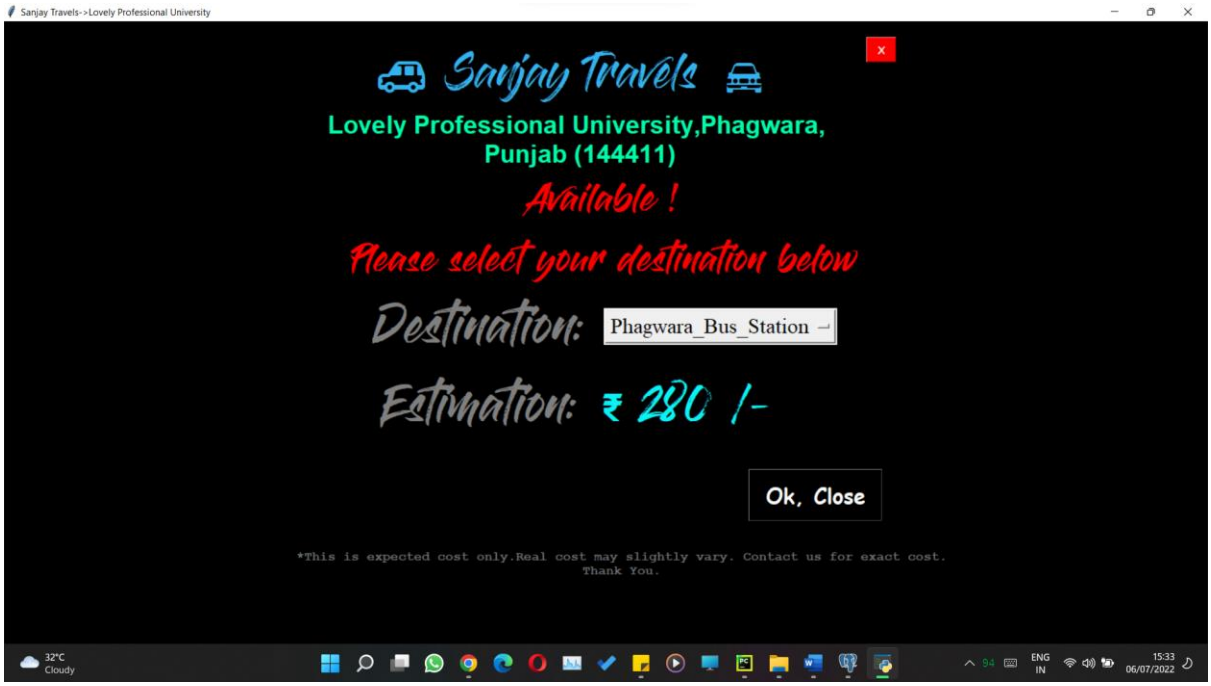
Selecting 4 Seater AC:



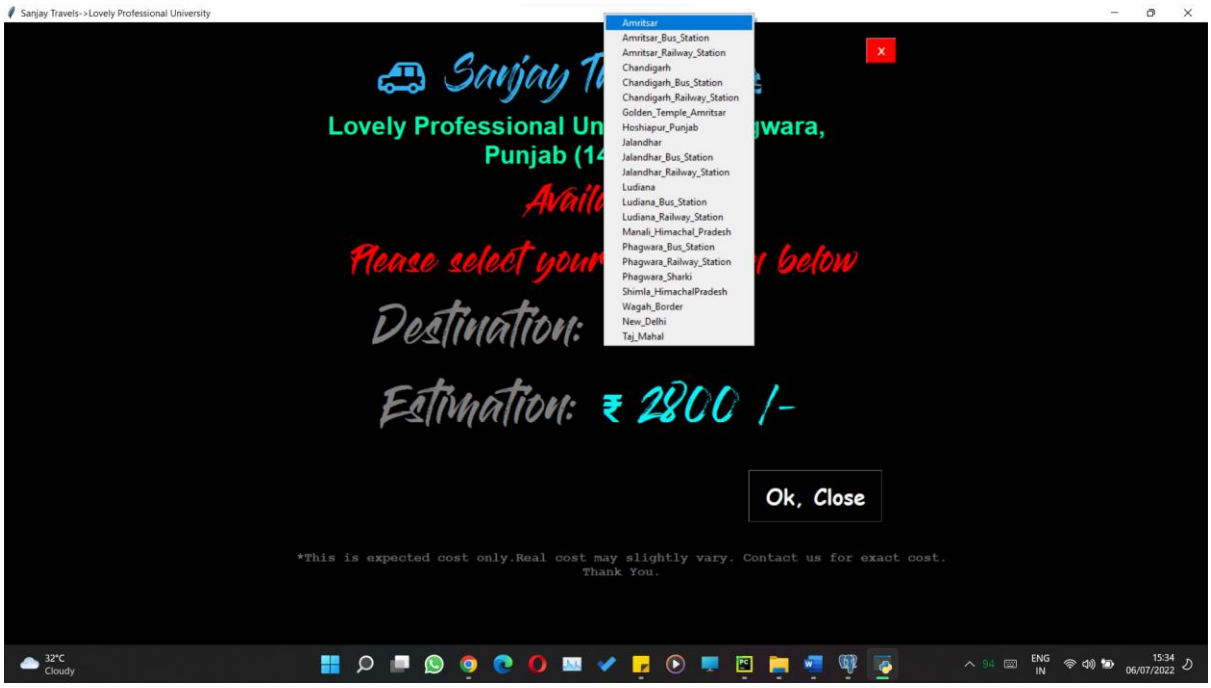
By: K. Sanjay Kumar

Live Price calculation page:

Selecting destination 'Phagwara'



Changing selection to 'Amritsar':



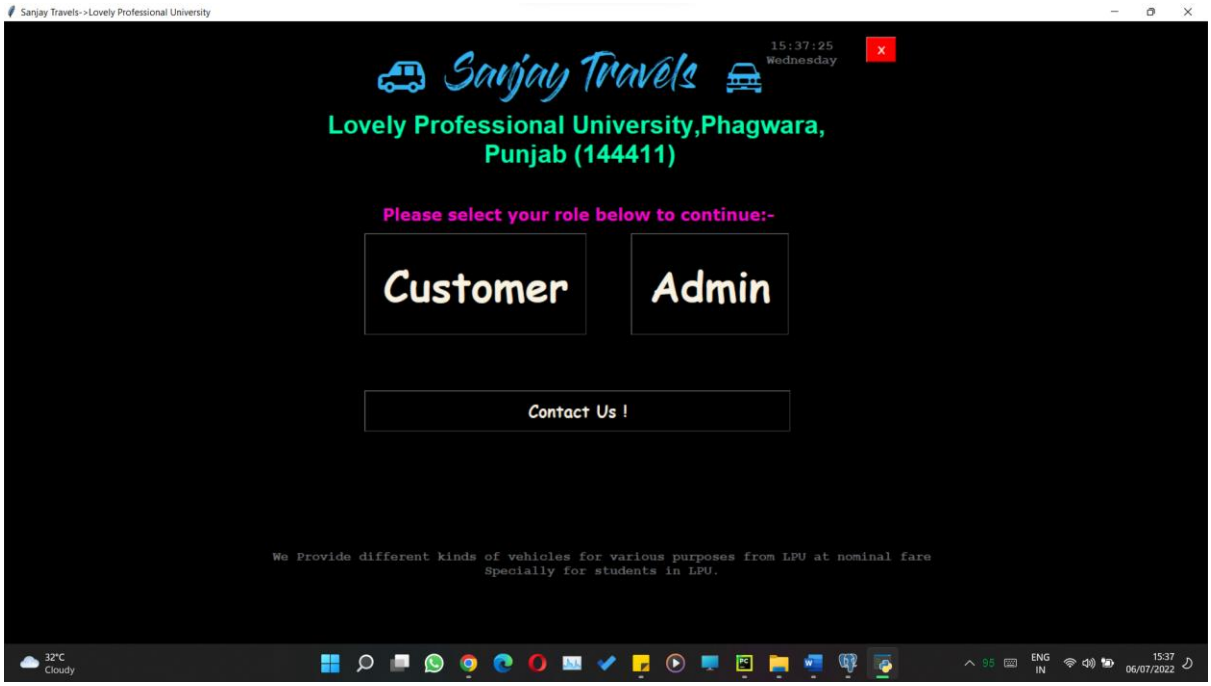
Here while we change the selection in the listbox the price automatically updates to new one.

Calculation and displaying will be on the go.

By: K. Sanjay Kumar

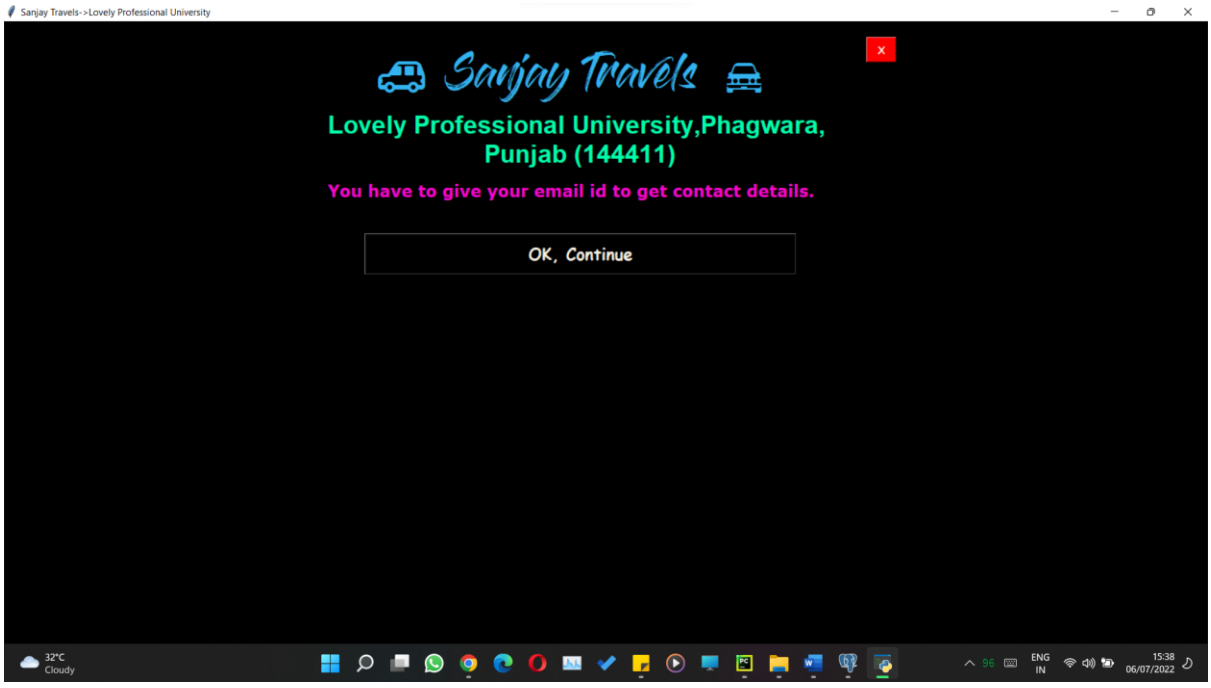
Contact us Section:

Pressing ok close and opening again: Homepage



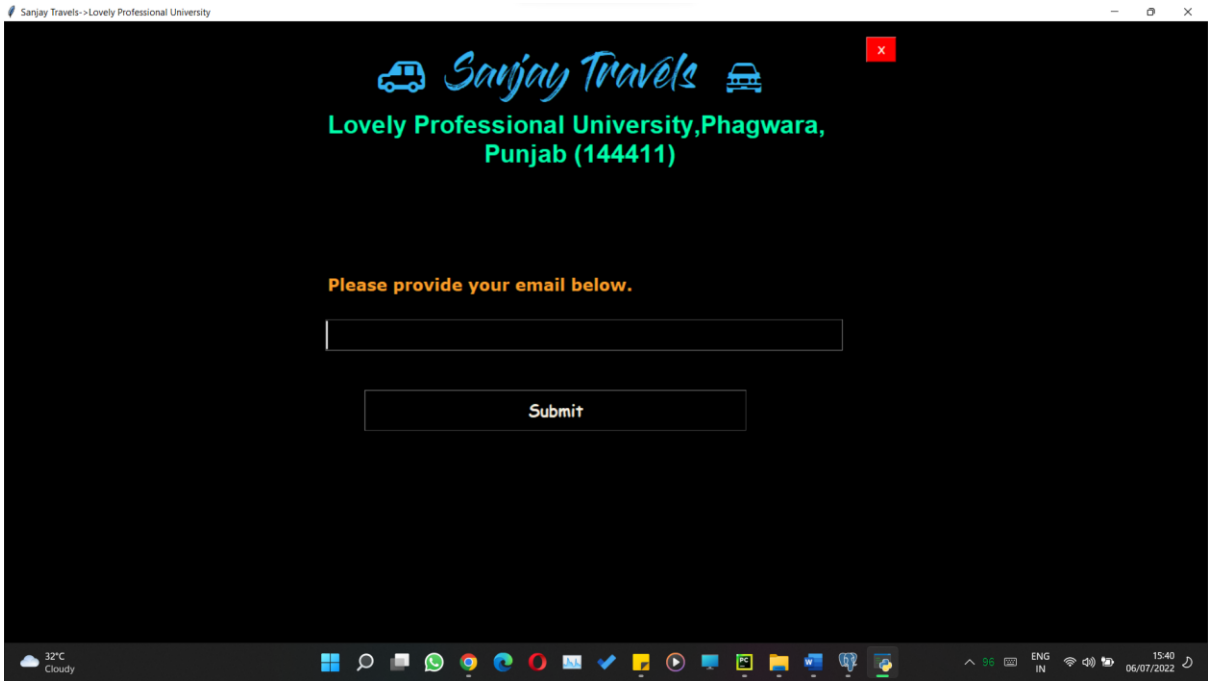
Clicking contact us:

Customer should agree to give the mail id.

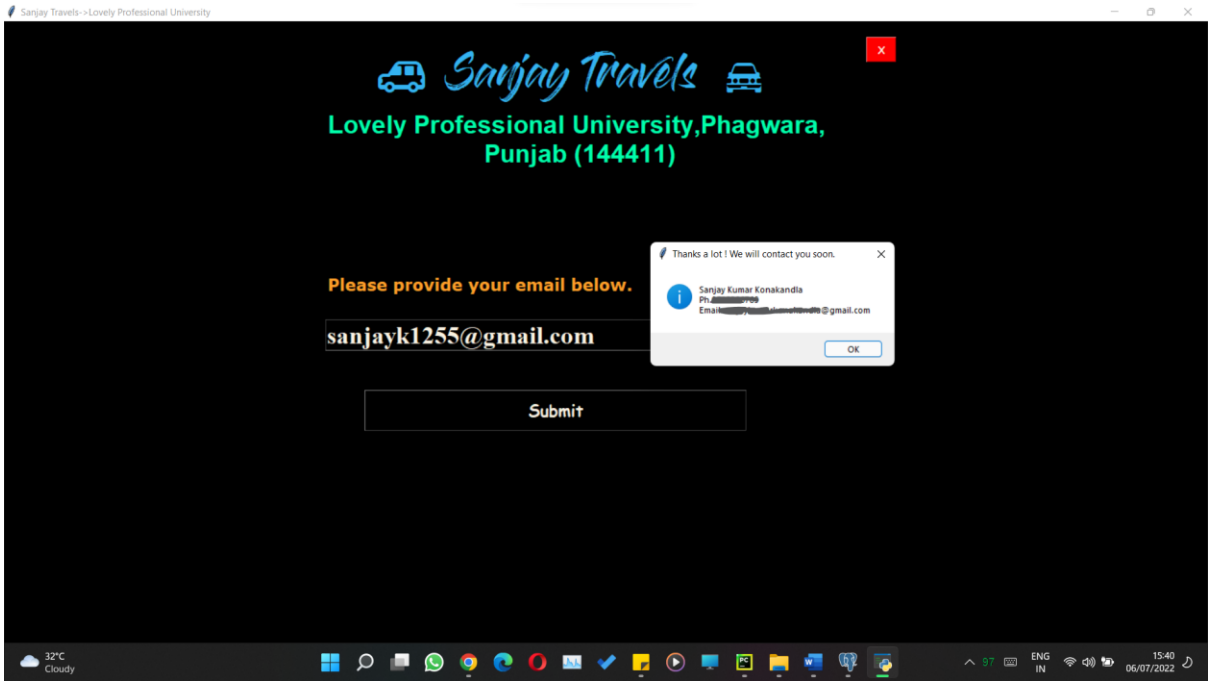


By: K. Sanjay Kumar

Customer mail entry page:



After entering correct email:



This program contains function to check mail given by customer. It simply matches that the entry reaches basic syntax of an email id or not.

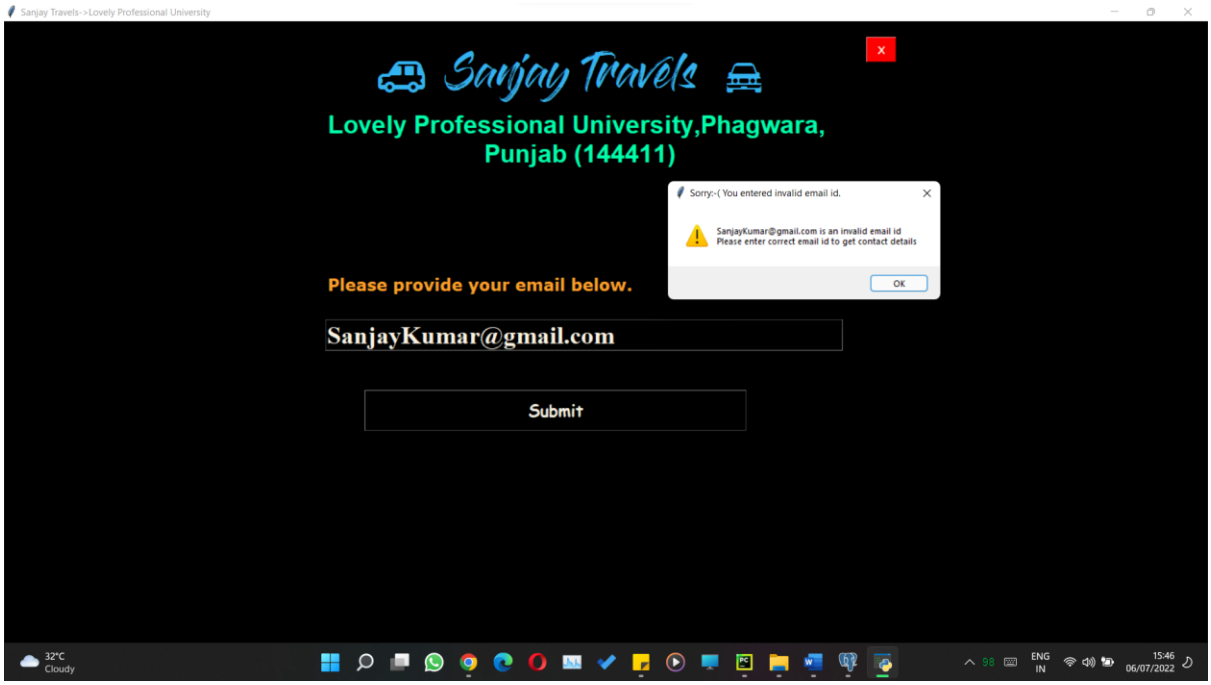
By: K. Sanjay Kumar

Mail verification function:

```
import re  # for email validation function
def validate(s):
    pat = "^[a-z0-9-_]+@[a-z0-9]+\.[a-z]{1,3}$"
    if re.match(pat, s):
        return True
    return False
```

here 's' is the email entered by the customer it is passed to this email validation function by other commands in the code. If that 's' matches the pattern then the validate function returns true otherwise it will return false.

Entering incorrect email:



Actually, this function just verifies the pattern of correct mail id. In general email id does not consists of capital letters. So this code caught incorrect email entered.

Correct email:

1. Does not contain capital letters
2. Does not contain special characters
3. Must have '@', domain name, '.' , in (or) com (or) live (or) etc.

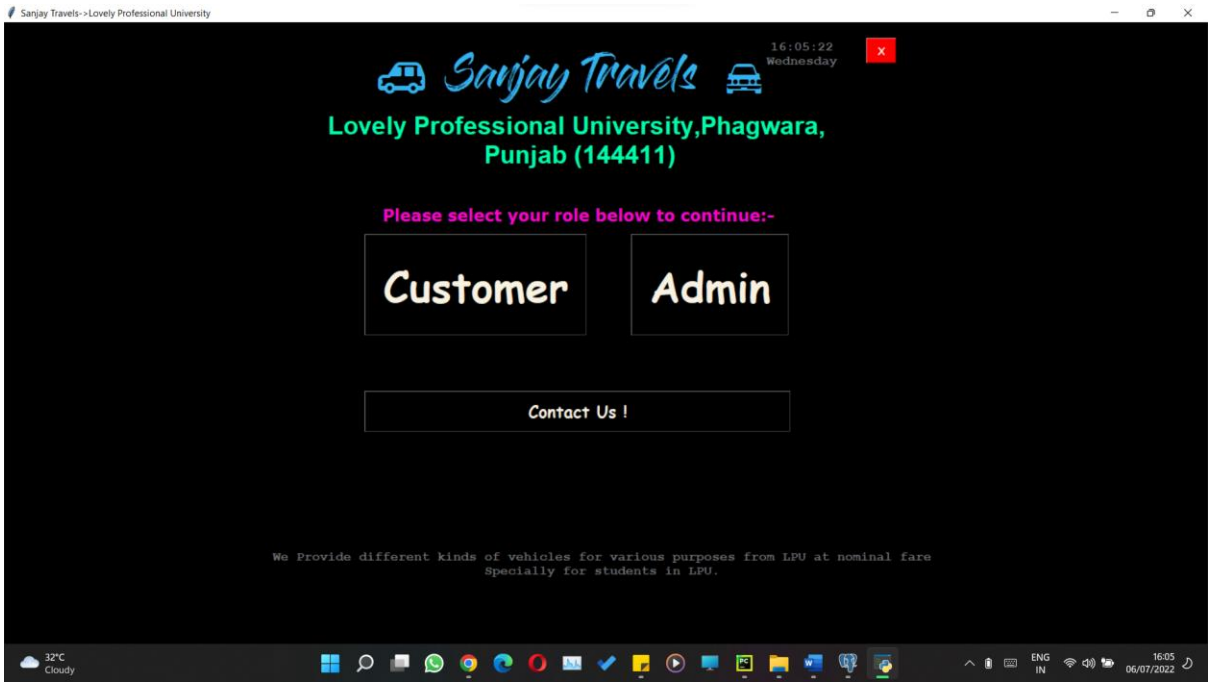
It will consider incorrect emails as correct if we match the above mentioned pattern. This is a disadvantage of this offline validation.

By: K. Sanjay Kumar

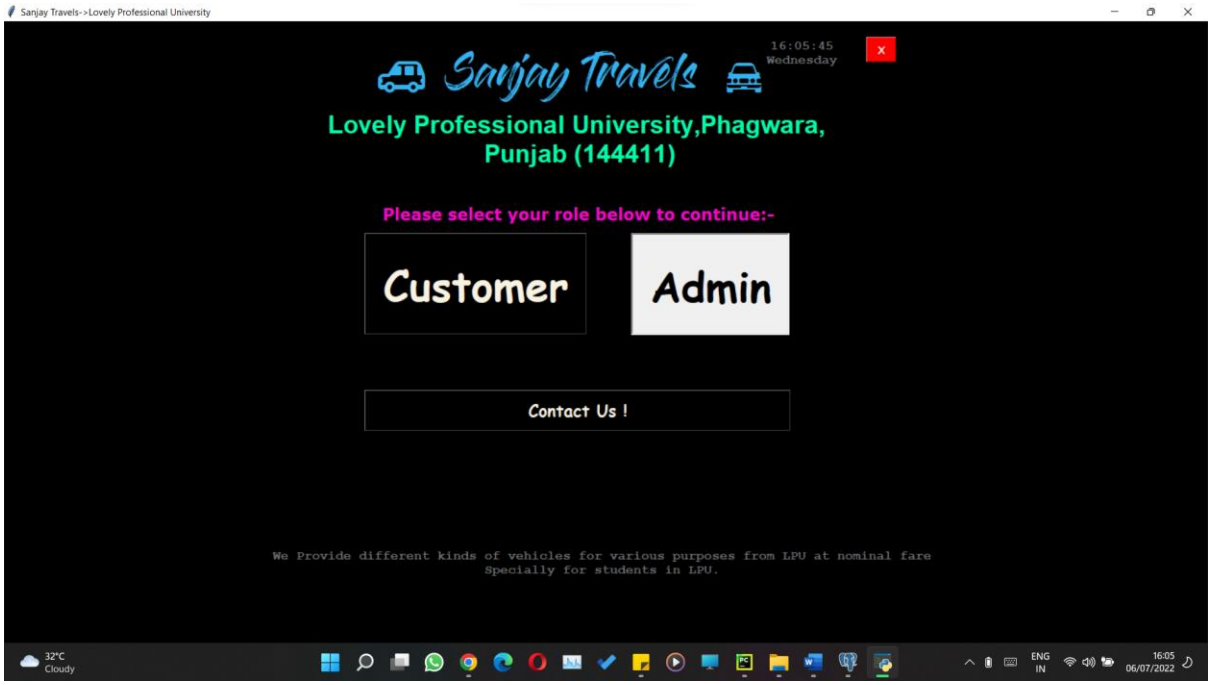
Admin Section

Closing everything and reopening again:

Home Page:



Selecting the Admin section:

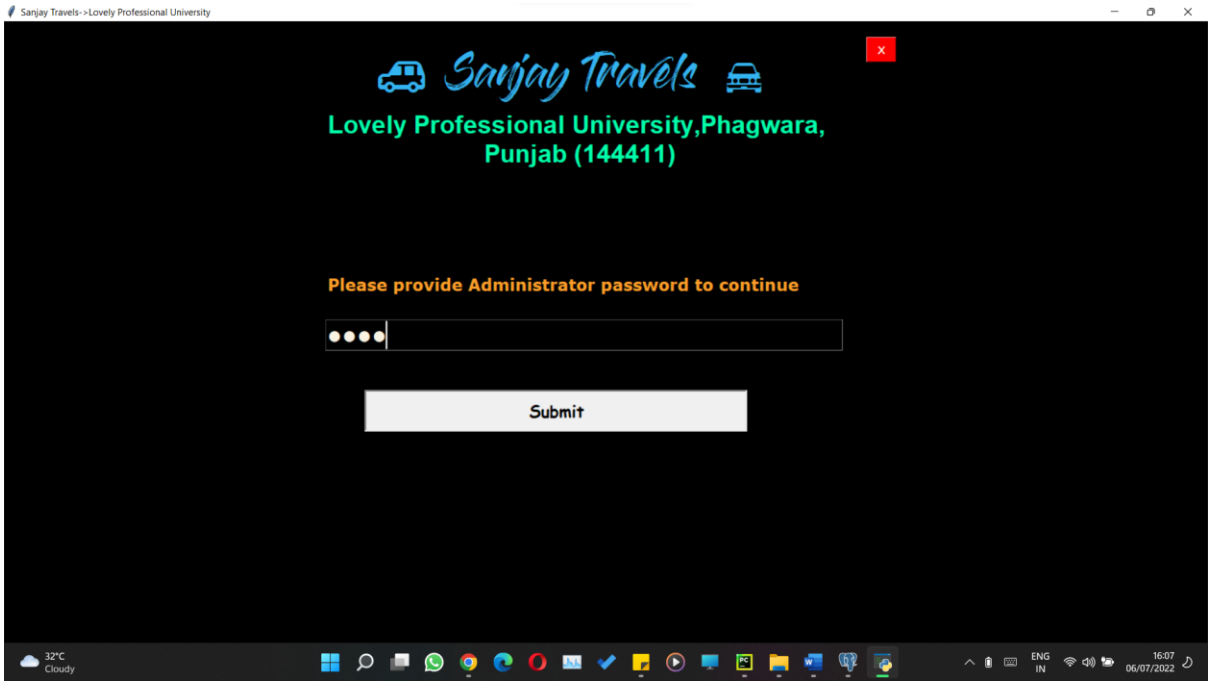


By: K. Sanjay Kumar

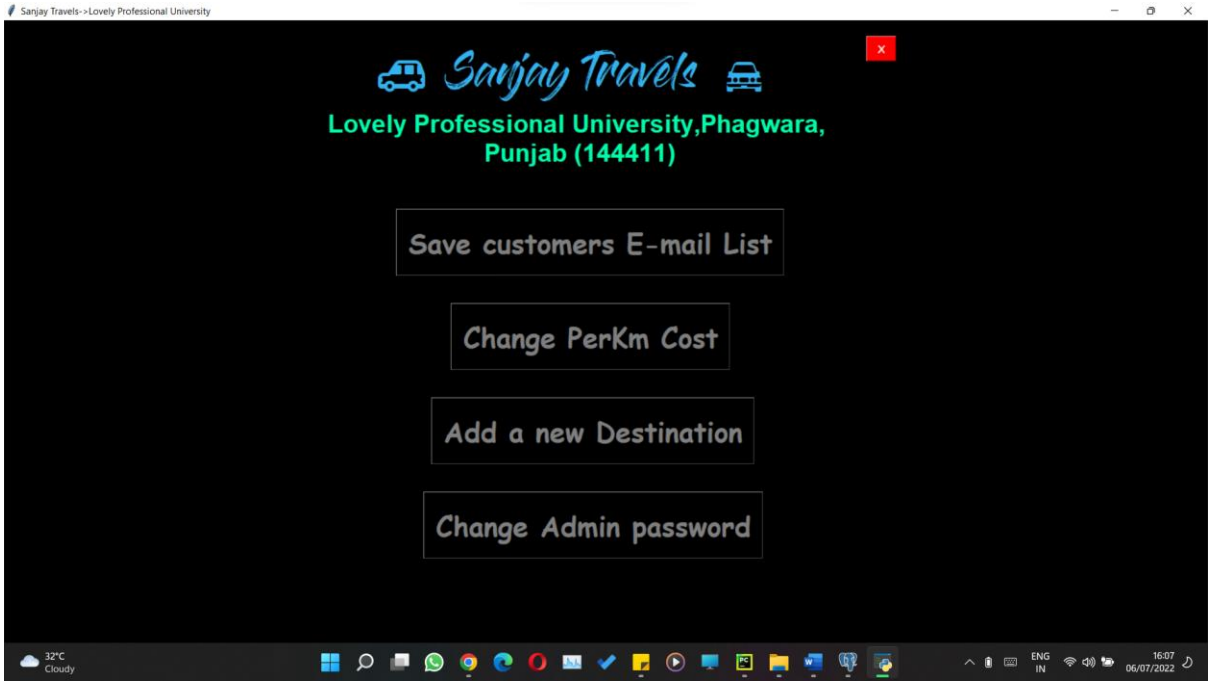
Admin password entry page:

At current instance admin password is 7722 but it can be changed as many times as possible.

Entering 7722 and clicking submit



Admin options page:



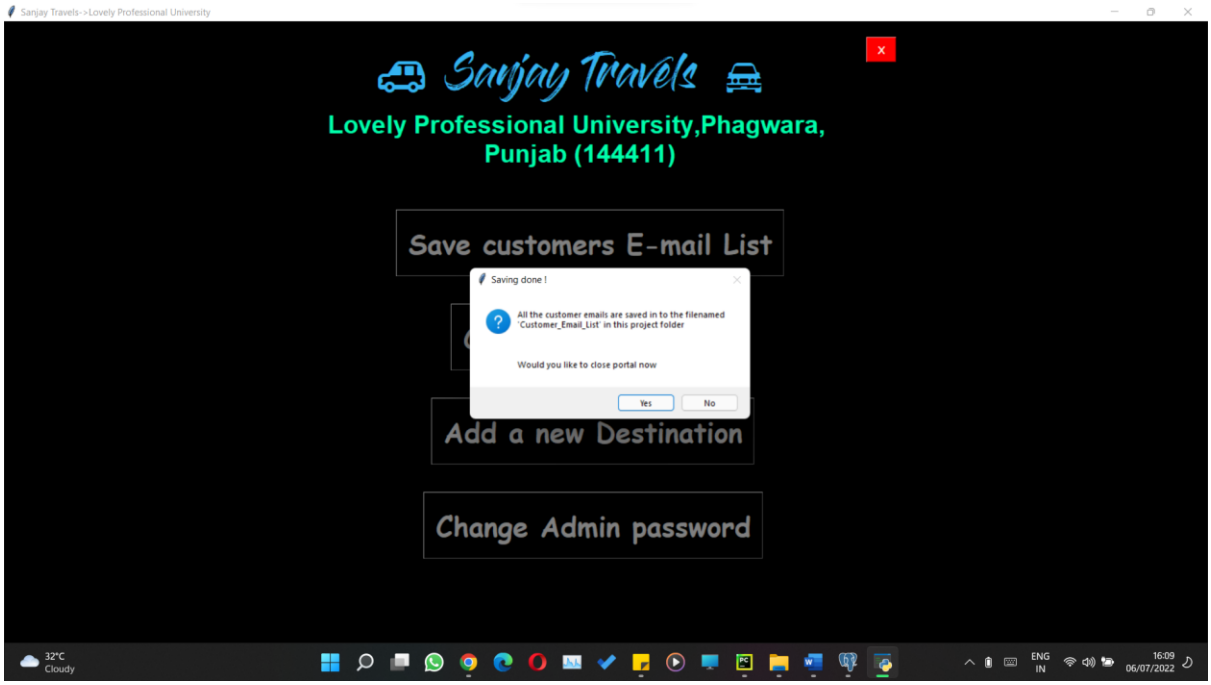
By: K. Sanjay Kumar

Admin options:

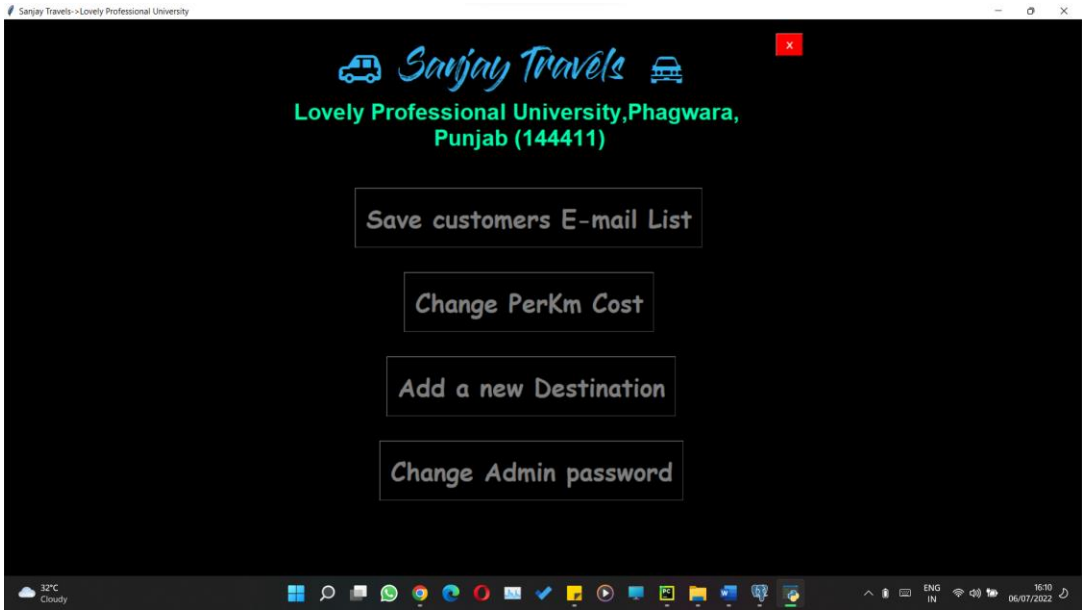
1. Save Customer Email list
2. Change PerKm Cost
3. Add a new destination
4. Change admin password

All the options available above works well.

Clicking on save customers email list:



Here if we press Yes the program will exit, I'm pressing no:



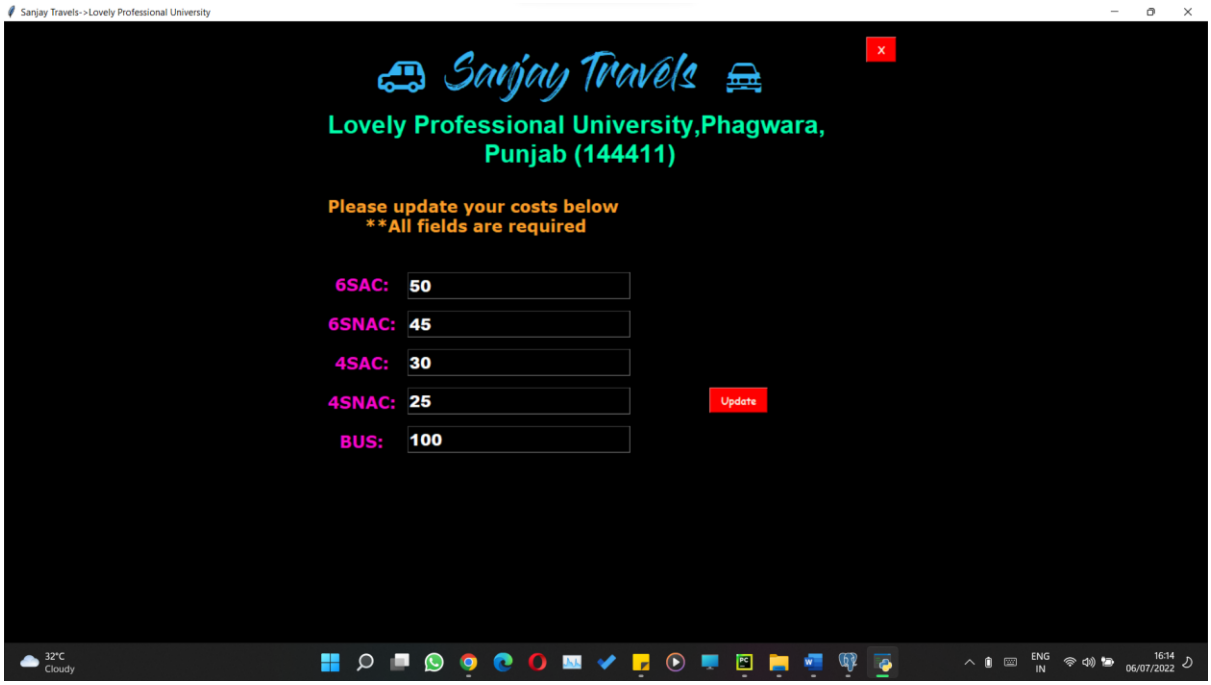
By: K. Sanjay Kumar

Change per Km cost of each car:

Data Output | Explain | Messages | Notifications

	car_type smallint	car_code character varying (10)	cost smallint
1	1	6SAC	45
2	2	6SNAC	42
3	3	4SAC	28
4	4	4SNAC	25
5	5	BUS	100

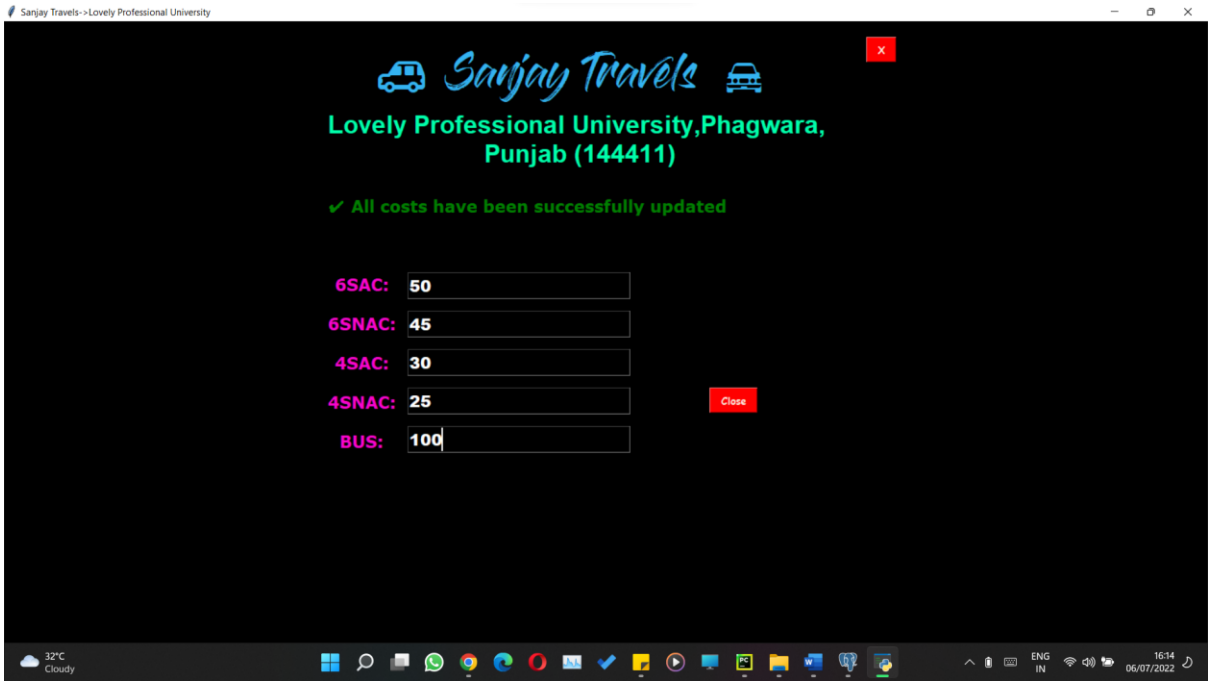
Updating costs:



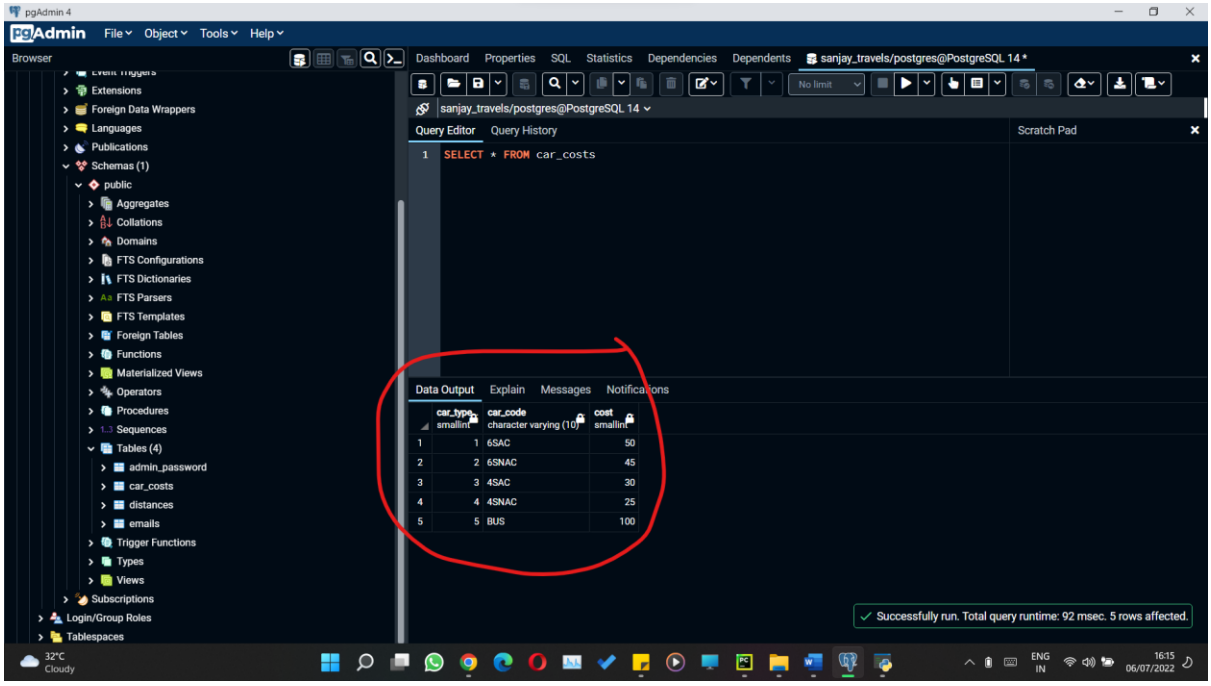
After clicking update, update button will turn in to close button and the text above turns into green and indicates that the updation successful.

By: K. Sanjay Kumar

After Clicking update:



Updated costs:

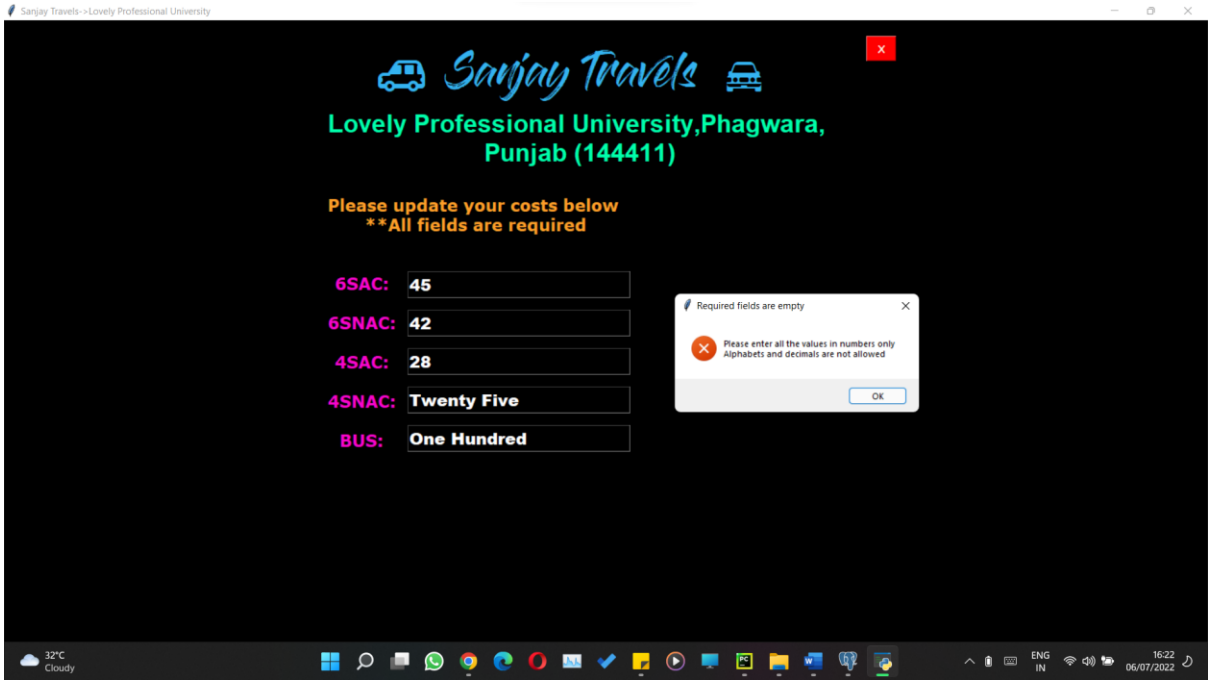


It has a fault tolerant algorithm behind the scenes which prevents insertion of decimal point numbers and other characters in to costs.

By: K. Sanjay Kumar

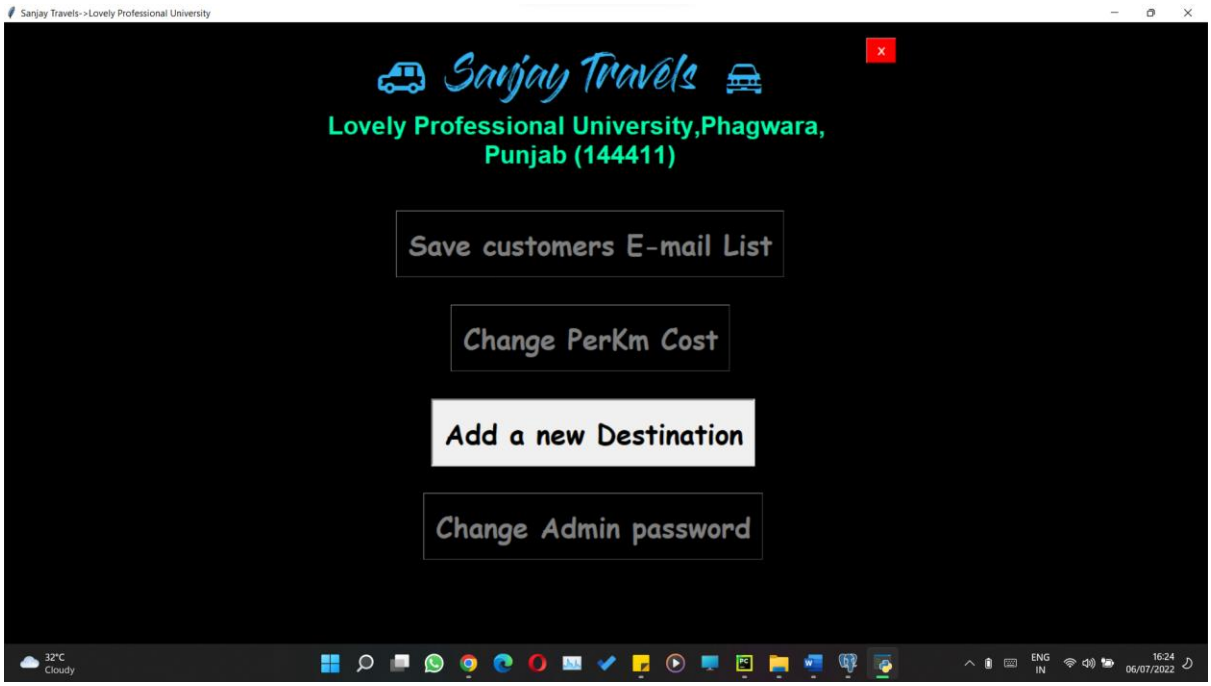
Intentionally entering wrong costs:

It shows an error.



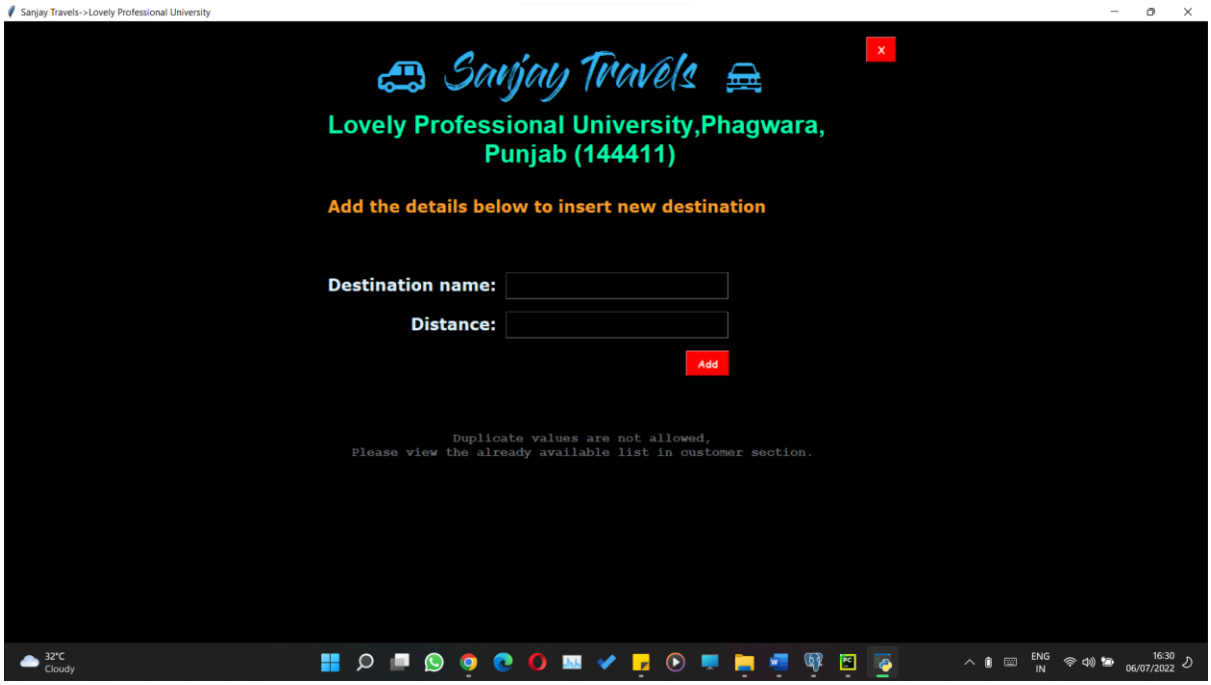
So fault tolerance worked well here.

Coming back and trying to add new destination:

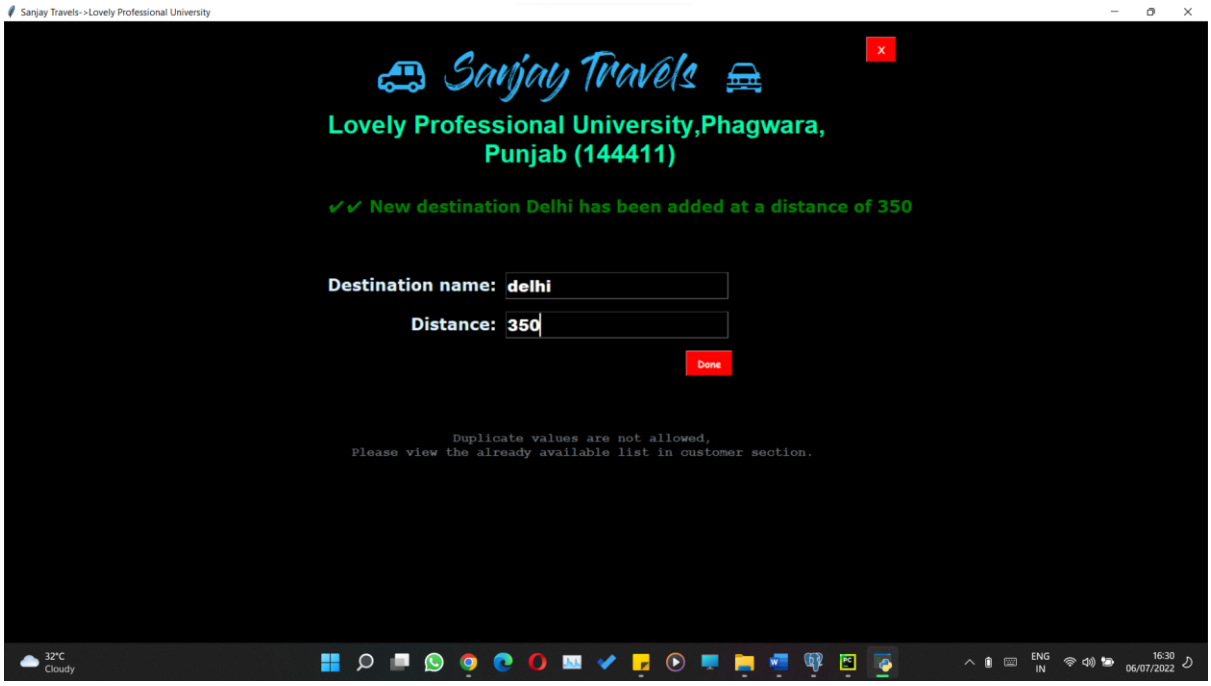


By: K. Sanjay Kumar

Add new destination details:



Adding details of Delhi and clicking add:



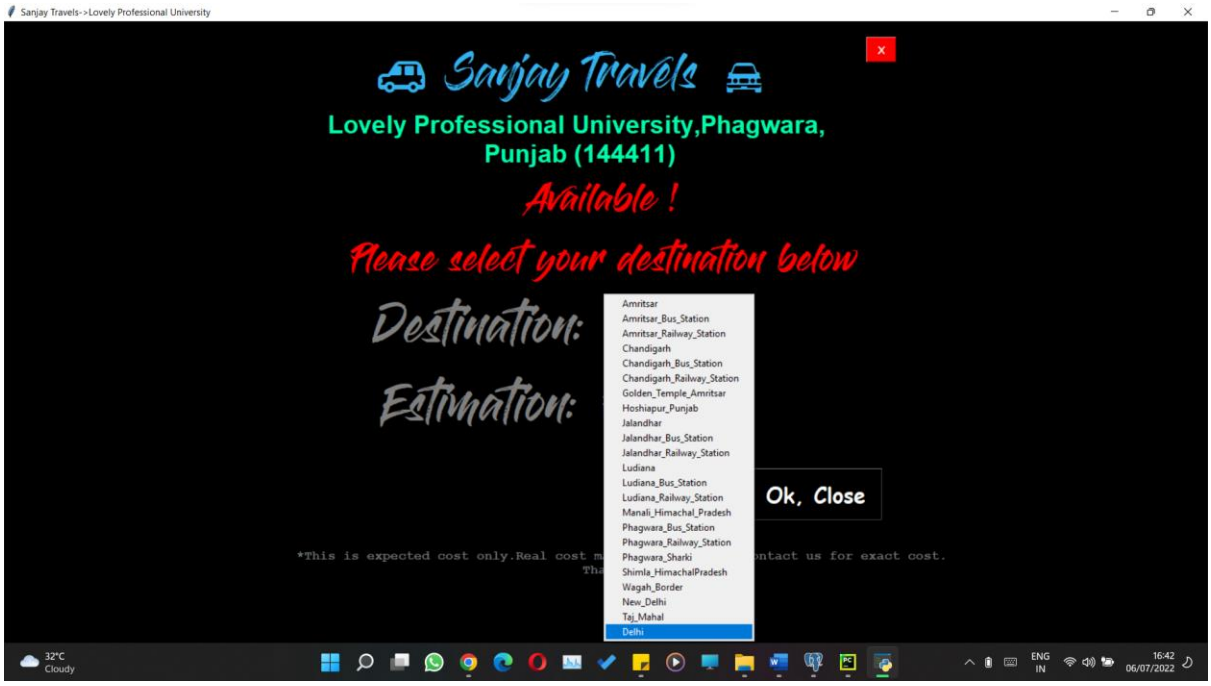
By: K. Sanjay Kumar

Previous list of destinations in calculation page:

Previously clicked list of destinations.

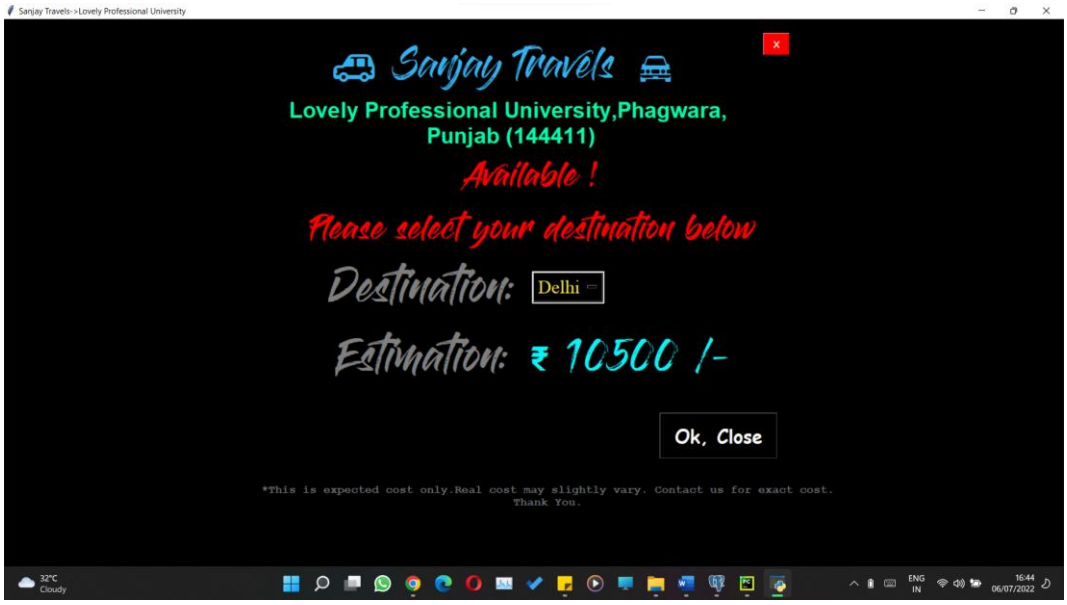
In above picture there is no destination name Delhi. So code accepted to add Delhi and its distance.

New list after adding Delhi in customer section:



Delhi price:

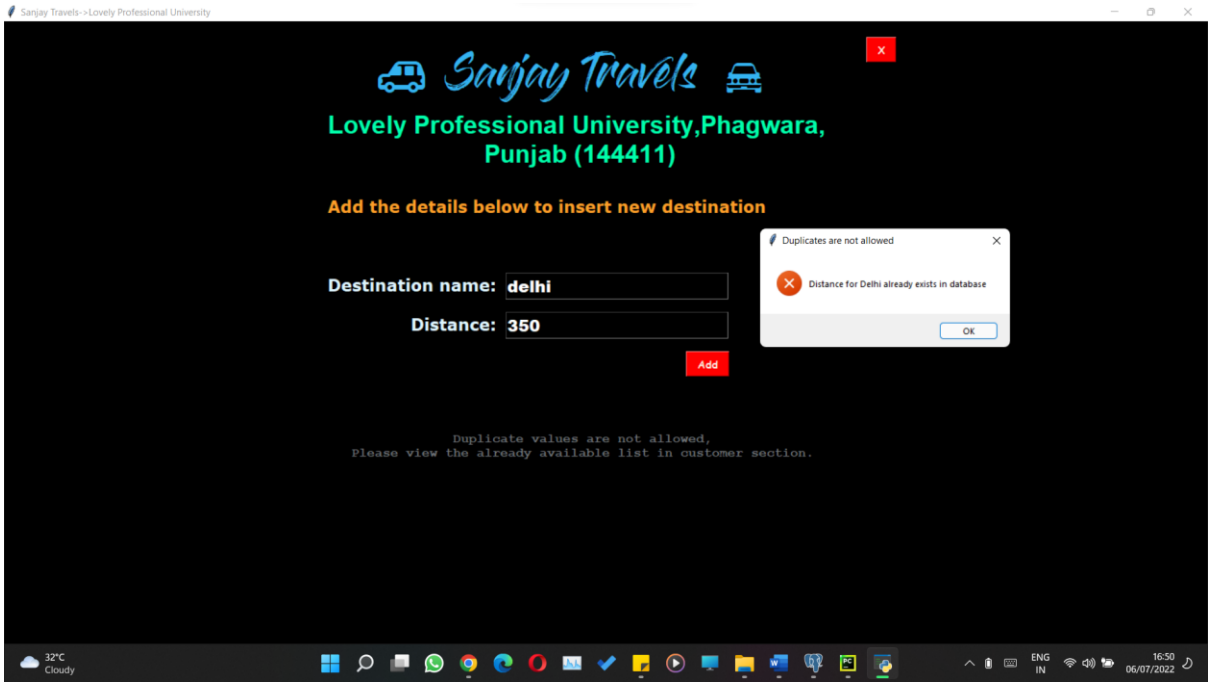
4SAC = 30(price we updated above)*350(distance) = ₹10500/-



By: K. Sanjay Kumar

Fault tolerance (It won't allow duplicate values)

Intentionally again trying to add delhi:



Formatting the destination name:

Text will be converted to title case first and then spaces will be replaced to underscores.

new delhi >> step 1 >> New Delhi >> step 2 >> New_Delhi

All this process goes in the code.

```
final_new_dest_name = str(new_dest_name_entry.get())
final_new_dest_name = final_new_dest_name.title()
final_new_dest_name = final_new_dest_name.replace(' ', '_')
```

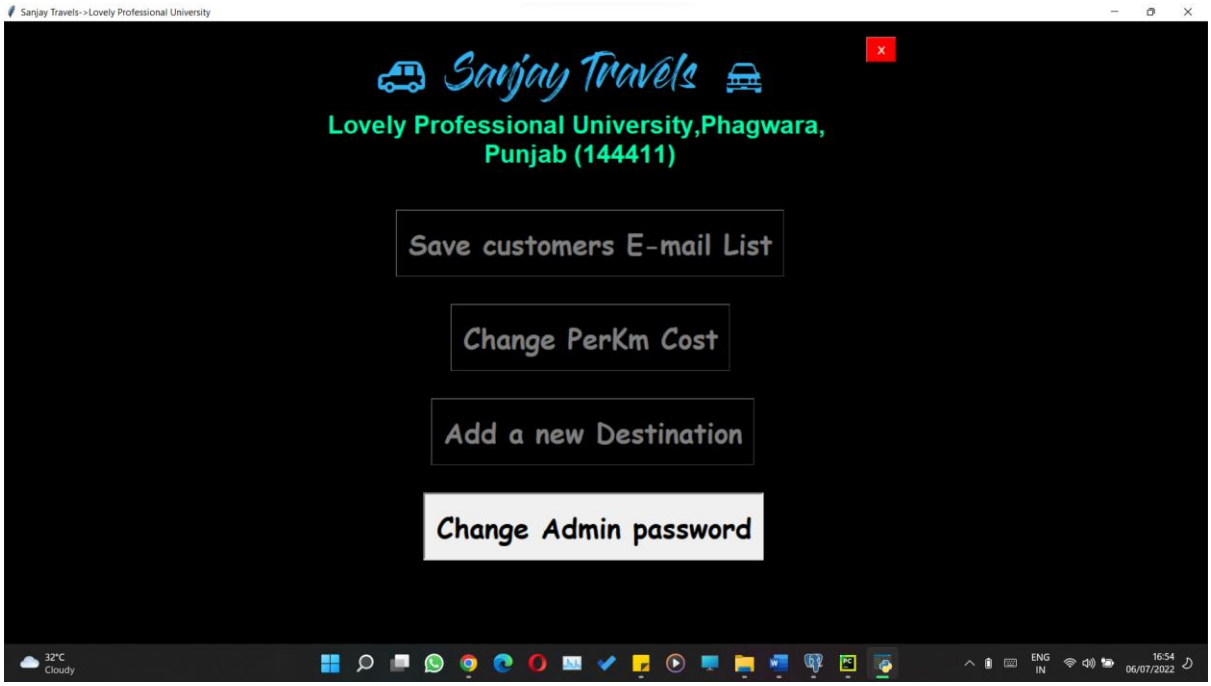
This is the code for formatting the destination name into required forms.

I actually tried to not replace the spaces with underscores but while retrieving the names back I got some different kinds of results. The name of the destination in which spaces are present is displaying correctly but they are surrounded with curly braces whereas others aren't. So I tried replacing spaces with underscores. It worked.

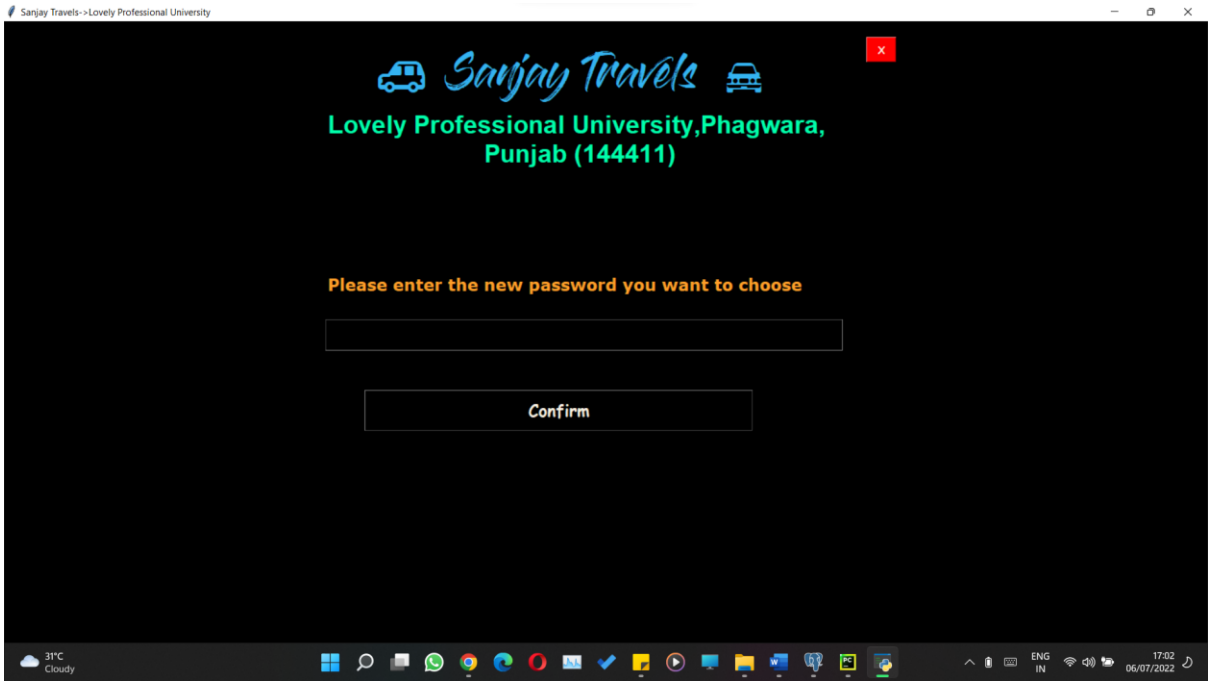
By: K. Sanjay Kumar

Change admin Password Section:

Previously I mentioned password as 7722 [Previous admin password]

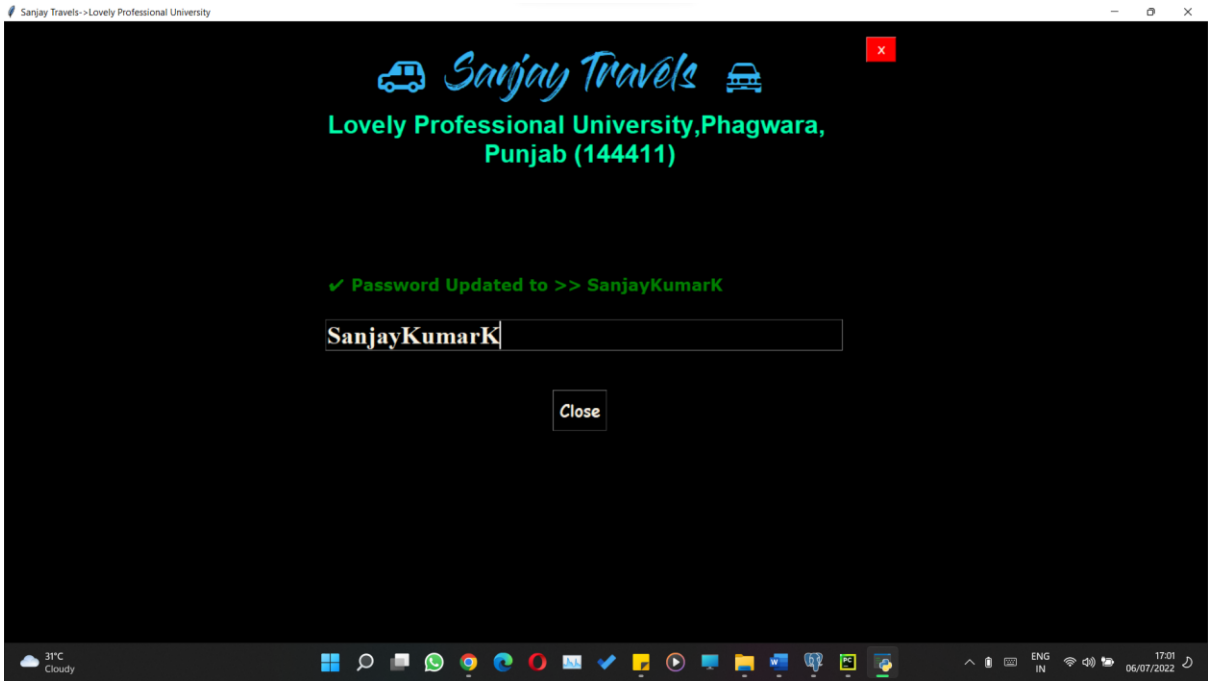


Entering new password:

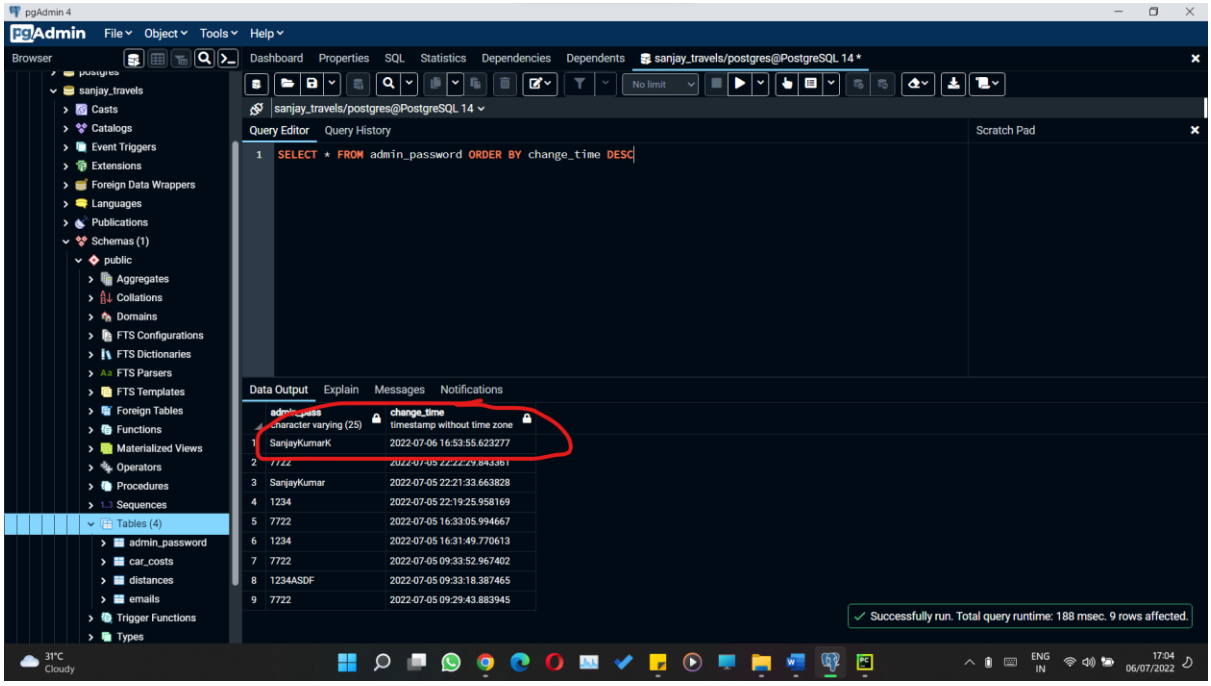


By: K. Sanjay Kumar

After entering the password clicking confirm:



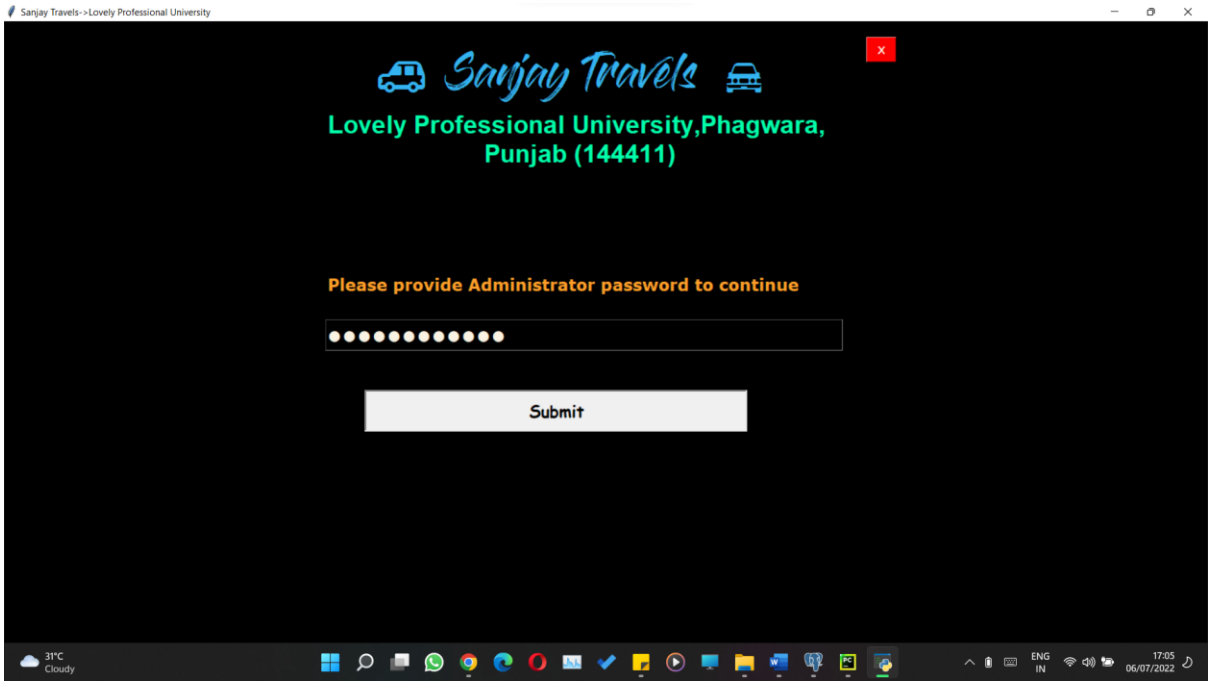
Checking password update in database:



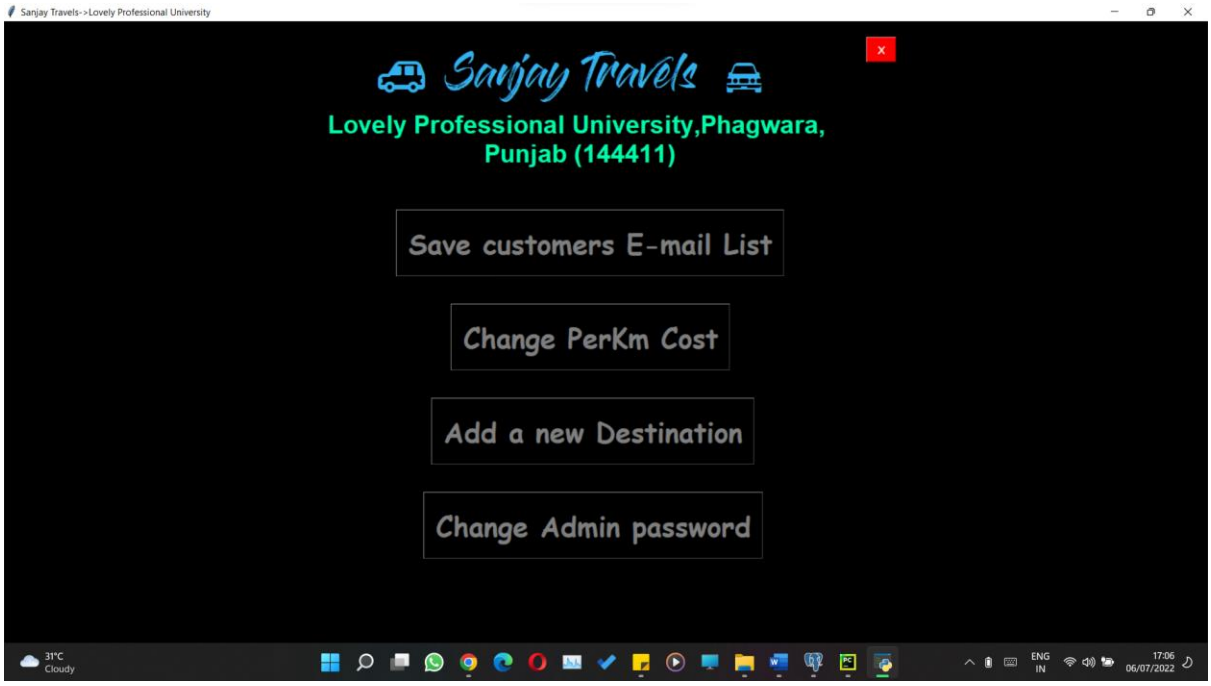
Here the password is registered.

By: K. Sanjay Kumar

Using new password:



Successfully logged in:



Conclusion:

Finally this portal can be used for lifetime because:

1. We can change admin password.
2. We can change costs of cars per kilometer because fuel cost changes.
3. We can add new destinations.
4. There is no need of changing distance because they won't grow or shrink.
5. There is a live clock on the home page which does not need any updation.