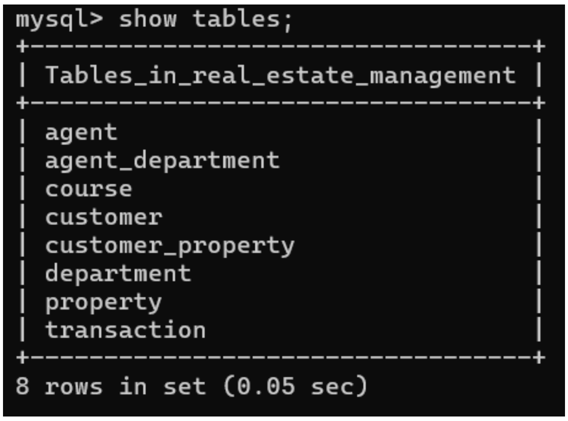
REAL ESTATE MANAGEMENT SYSTEM

In this project, for backend, MySQL database is used to store the data. So, in MySQL,

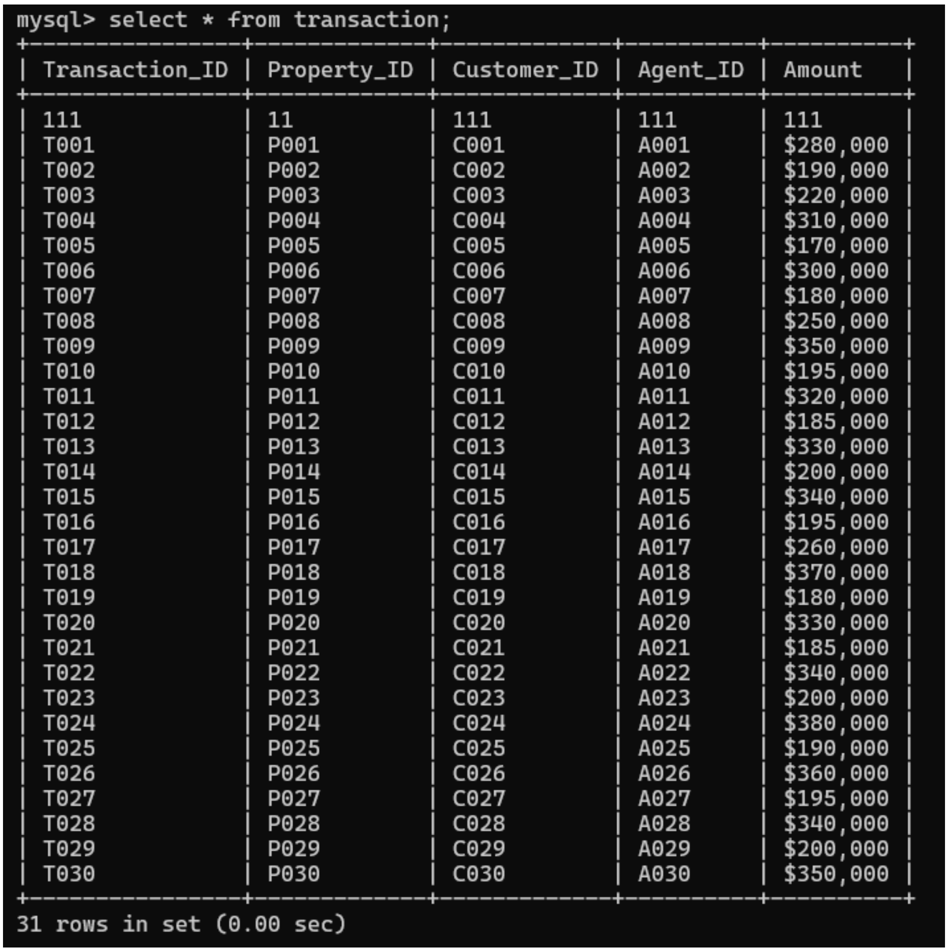
Database Name – Real_Estate_Management

Tables –

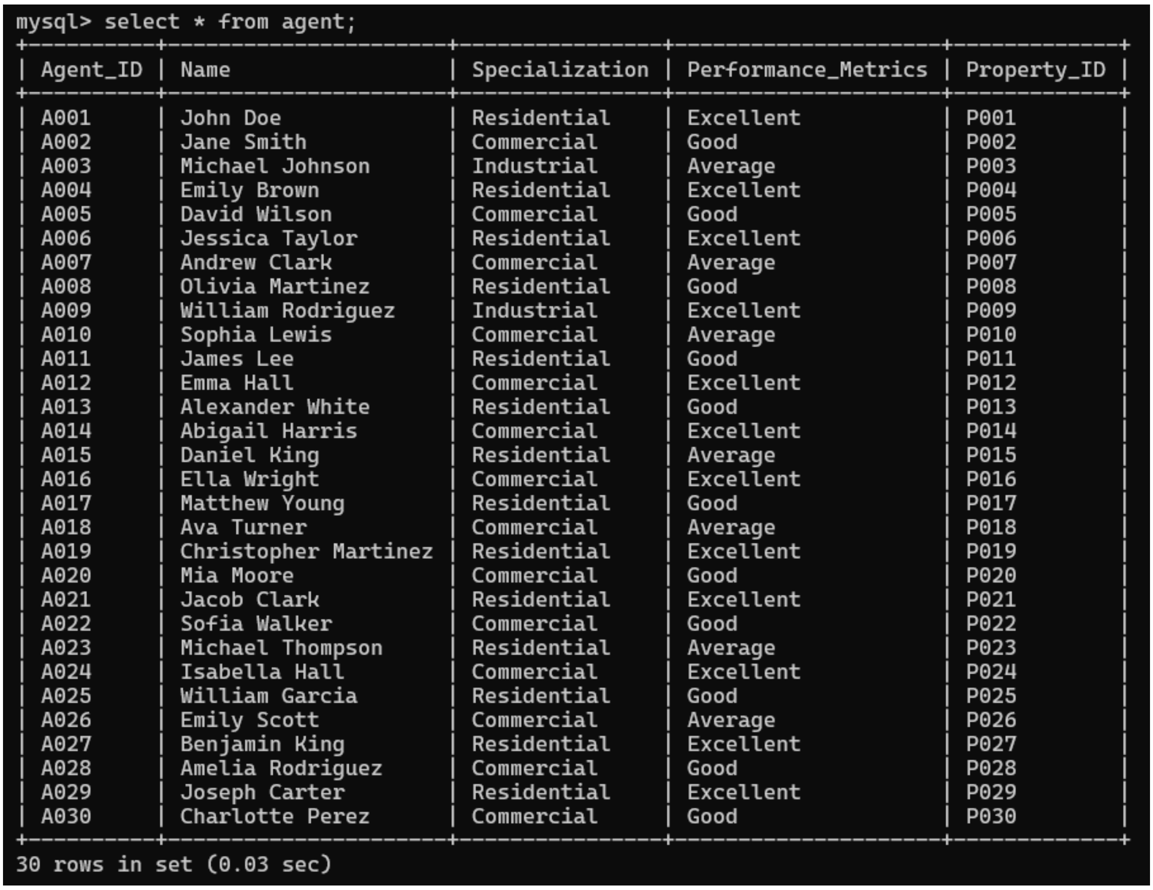


```
mysql> show tables;
+----------------------------------+
| Tables_in_real_estate_management |
+----------------------------------+
| agent                            |
| agent_department                 |
| course                           |
| customer                         |
| customer_property                |
| department                       |
| property                         |
| transaction                      |
+----------------------------------+
8 rows in set (0.05 sec)
```

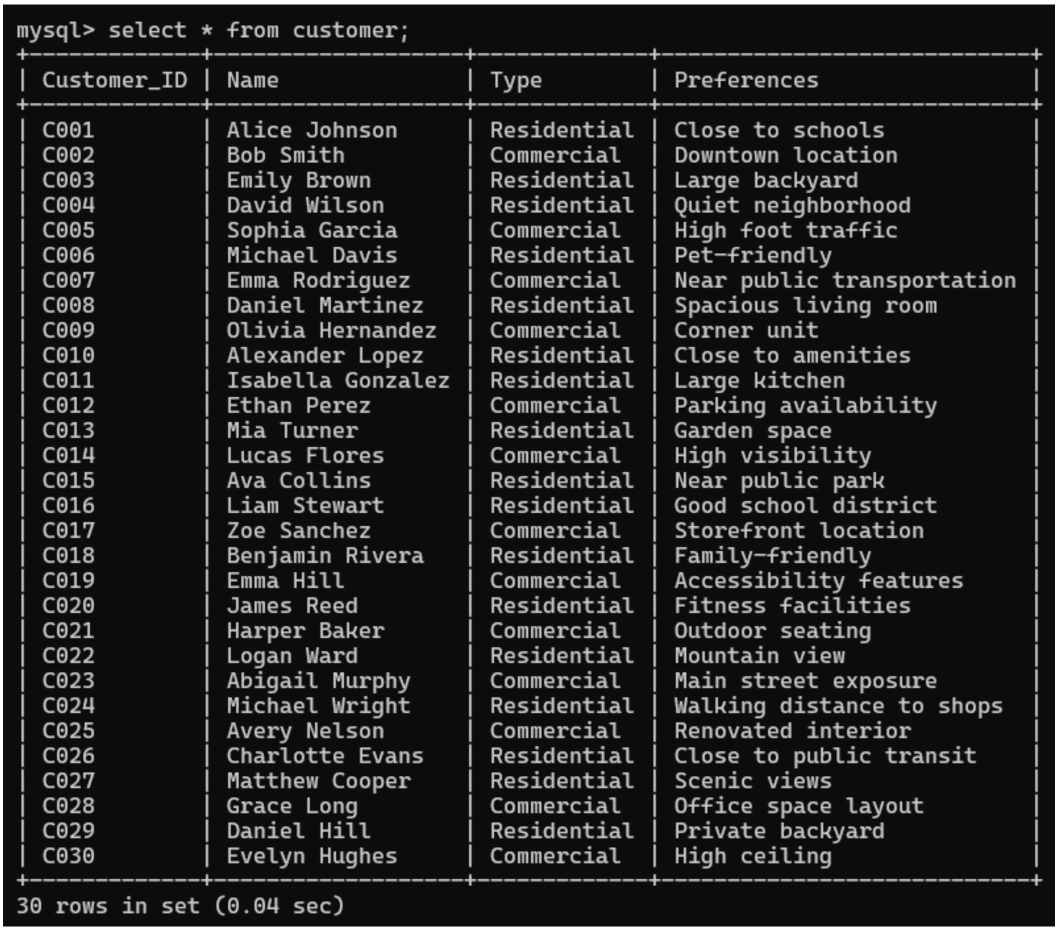
Sample data in a table –



```
mysql> select * from transaction;
+----------------+-------------+-------------+----------+-----------+
| Transaction_ID | Property_ID | Customer_ID | Agent_ID | Amount    |
+----------------+-------------+-------------+----------+-----------+
| 111            | 11          | 111         | 111      | 111       |
| T001           | P001        | C001        | A001     | $280,000  |
| T002           | P002        | C002        | A002     | $190,000  |
| T003           | P003        | C003        | A003     | $220,000  |
| T004           | P004        | C004        | A004     | $310,000  |
| T005           | P005        | C005        | A005     | $170,000  |
| T006           | P006        | C006        | A006     | $300,000  |
| T007           | P007        | C007        | A007     | $180,000  |
| T008           | P008        | C008        | A008     | $250,000  |
| T009           | P009        | C009        | A009     | $350,000  |
| T010           | P010        | C010        | A010     | $195,000  |
| T011           | P011        | C011        | A011     | $320,000  |
| T012           | P012        | C012        | A012     | $185,000  |
| T013           | P013        | C013        | A013     | $330,000  |
| T014           | P014        | C014        | A014     | $200,000  |
| T015           | P015        | C015        | A015     | $340,000  |
| T016           | P016        | C016        | A016     | $195,000  |
| T017           | P017        | C017        | A017     | $260,000  |
| T018           | P018        | C018        | A018     | $370,000  |
| T019           | P019        | C019        | A019     | $180,000  |
| T020           | P020        | C020        | A020     | $330,000  |
| T021           | P021        | C021        | A021     | $185,000  |
| T022           | P022        | C022        | A022     | $340,000  |
| T023           | P023        | C023        | A023     | $200,000  |
| T024           | P024        | C024        | A024     | $380,000  |
| T025           | P025        | C025        | A025     | $190,000  |
| T026           | P026        | C026        | A026     | $360,000  |
| T027           | P027        | C027        | A027     | $195,000  |
| T028           | P028        | C028        | A028     | $340,000  |
| T029           | P029        | C029        | A029     | $200,000  |
| T030           | P030        | C030        | A030     | $350,000  |
+----------------+-------------+-------------+----------+-----------+
31 rows in set (0.00 sec)
```



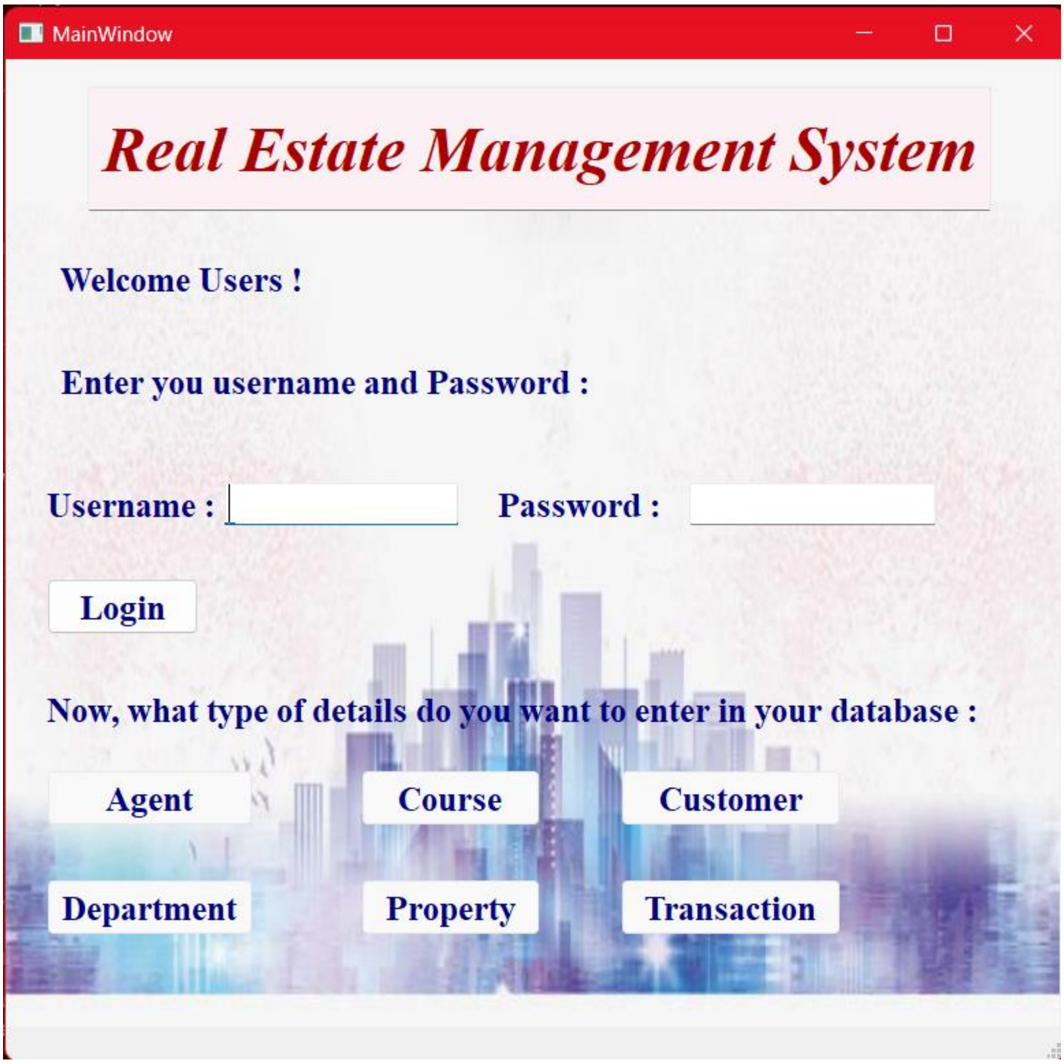
```
mysql> select * from agent;
+----------+----------------------+----------------+---------------------+-------------+
| Agent_ID | Name                 | Specialization | Performance_Metrics | Property_ID |
+----------+----------------------+----------------+---------------------+-------------+
| A001     | John Doe             | Residential    | Excellent           | P001        |
| A002     | Jane Smith           | Commercial     | Good                | P002        |
| A003     | Michael Johnson      | Industrial     | Average             | P003        |
| A004     | Emily Brown          | Residential    | Excellent           | P004        |
| A005     | David Wilson         | Commercial     | Good                | P005        |
| A006     | Jessica Taylor       | Residential    | Excellent           | P006        |
| A007     | Andrew Clark         | Commercial     | Average             | P007        |
| A008     | Olivia Martinez      | Residential    | Good                | P008        |
| A009     | William Rodriguez    | Industrial     | Excellent           | P009        |
| A010     | Sophia Lewis         | Commercial     | Average             | P010        |
| A011     | James Lee            | Residential    | Good                | P011        |
| A012     | Emma Hall            | Commercial     | Excellent           | P012        |
| A013     | Alexander White      | Residential    | Good                | P013        |
| A014     | Abigail Harris       | Commercial     | Excellent           | P014        |
| A015     | Daniel King          | Residential    | Average             | P015        |
| A016     | Ella Wright          | Commercial     | Excellent           | P016        |
| A017     | Matthew Young        | Residential    | Good                | P017        |
| A018     | Ava Turner           | Commercial     | Average             | P018        |
| A019     | Christopher Martinez | Residential    | Excellent           | P019        |
| A020     | Mia Moore            | Commercial     | Good                | P020        |
| A021     | Jacob Clark          | Residential    | Excellent           | P021        |
| A022     | Sofia Walker         | Commercial     | Good                | P022        |
| A023     | Michael Thompson     | Residential    | Average             | P023        |
| A024     | Isabella Hall        | Commercial     | Excellent           | P024        |
| A025     | William Garcia       | Residential    | Good                | P025        |
| A026     | Emily Scott          | Commercial     | Average             | P026        |
| A027     | Benjamin King        | Residential    | Excellent           | P027        |
| A028     | Amelia Rodriguez     | Commercial     | Good                | P028        |
| A029     | Joseph Carter        | Residential    | Excellent           | P029        |
| A030     | Charlotte Perez      | Commercial     | Good                | P030        |
+----------+----------------------+----------------+---------------------+-------------+
30 rows in set (0.03 sec)
```



```
mysql> select * from customer;
+-------------+-------------------+-------------+----------------------------+
| Customer_ID | Name              | Type        | Preferences                |
+-------------+-------------------+-------------+----------------------------+
| C001        | Alice Johnson     | Residential | Close to schools           |
| C002        | Bob Smith         | Commercial  | Downtown location          |
| C003        | Emily Brown       | Residential | Large backyard             |
| C004        | David Wilson      | Residential | Quiet neighborhood         |
| C005        | Sophia Garcia     | Commercial  | High foot traffic          |
| C006        | Michael Davis     | Residential | Pet-friendly               |
| C007        | Emma Rodriguez    | Commercial  | Near public transportation |
| C008        | Daniel Martinez   | Residential | Spacious living room       |
| C009        | Olivia Hernandez  | Commercial  | Corner unit                |
| C010        | Alexander Lopez   | Residential | Close to amenities         |
| C011        | Isabella Gonzalez | Residential | Large kitchen              |
| C012        | Ethan Perez       | Commercial  | Parking availability       |
| C013        | Mia Turner        | Residential | Garden space               |
| C014        | Lucas Flores      | Commercial  | High visibility            |
| C015        | Ava Collins       | Residential | Near public park           |
| C016        | Liam Stewart      | Residential | Good school district       |
| C017        | Zoe Sanchez       | Commercial  | Storefront location        |
| C018        | Benjamin Rivera   | Residential | Family-friendly            |
| C019        | Emma Hill         | Commercial  | Accessibility features     |
| C020        | James Reed        | Residential | Fitness facilities         |
| C021        | Harper Baker      | Commercial  | Outdoor seating            |
| C022        | Logan Ward        | Residential | Mountain view              |
| C023        | Abigail Murphy    | Commercial  | Main street exposure       |
| C024        | Michael Wright    | Residential | Walking distance to shops  |
| C025        | Avery Nelson      | Commercial  | Renovated interior         |
| C026        | Charlotte Evans   | Residential | Close to public transit    |
| C027        | Matthew Cooper    | Residential | Scenic views               |
| C028        | Grace Long        | Commercial  | Office space layout        |
| C029        | Daniel Hill       | Residential | Private backyard           |
| C030        | Evelyn Hughes     | Commercial  | High ceiling               |
+-------------+-------------------+-------------+----------------------------+
30 rows in set (0.04 sec)
```

For the Interface, the python designer PyQT5 is used.

Main User Interface –



Real Estate Management System

Welcome Users !

Enter you username and Password :

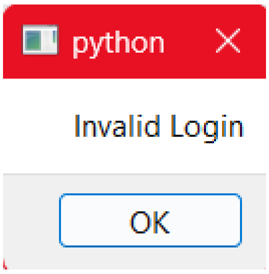
Username :

Password :

Now, what type of details do you want to enter in your database :

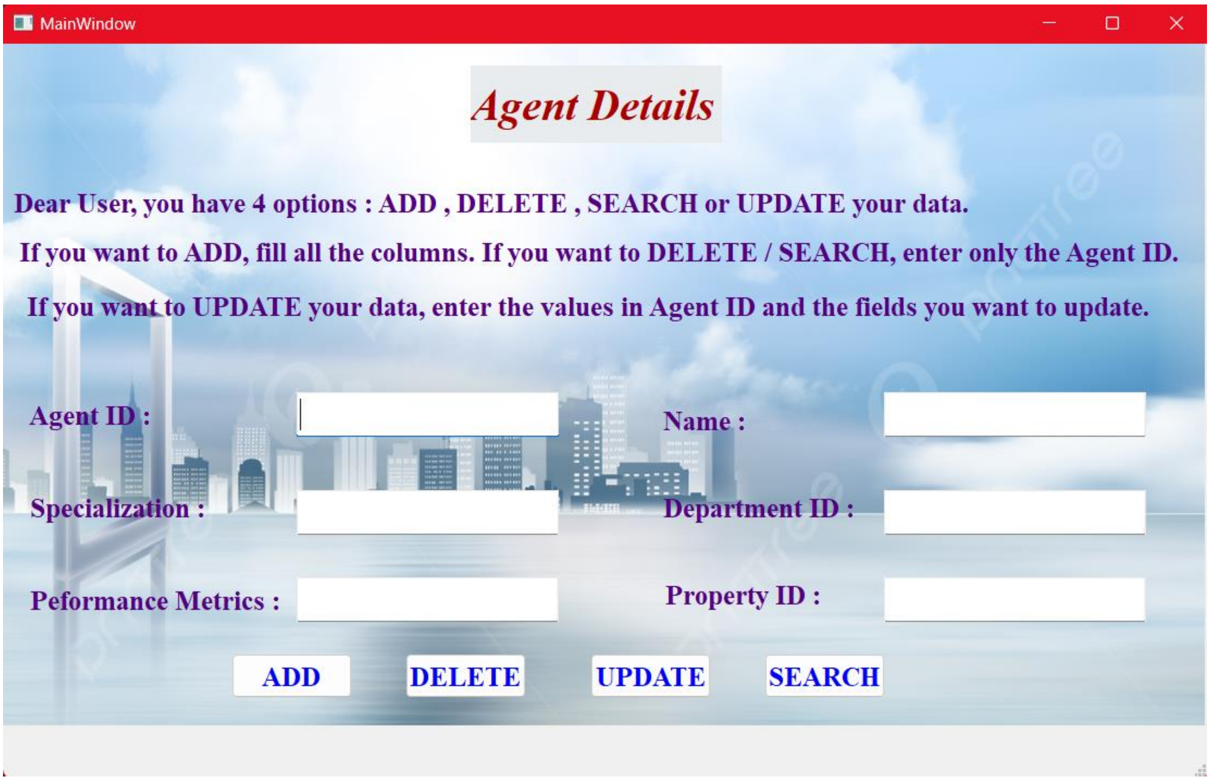
In this interface, unless the right username and password is given you cannot access the buttons present underneath for different tables.

If wrong password is given, this prompt occurs –



Invalid Login

After login, there are 6 separate UIs for every table in the MySQL database.



Agent Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only the Agent ID.

If you want to UPDATE your data, enter the values in Agent ID and the fields you want to update.

Agent ID :

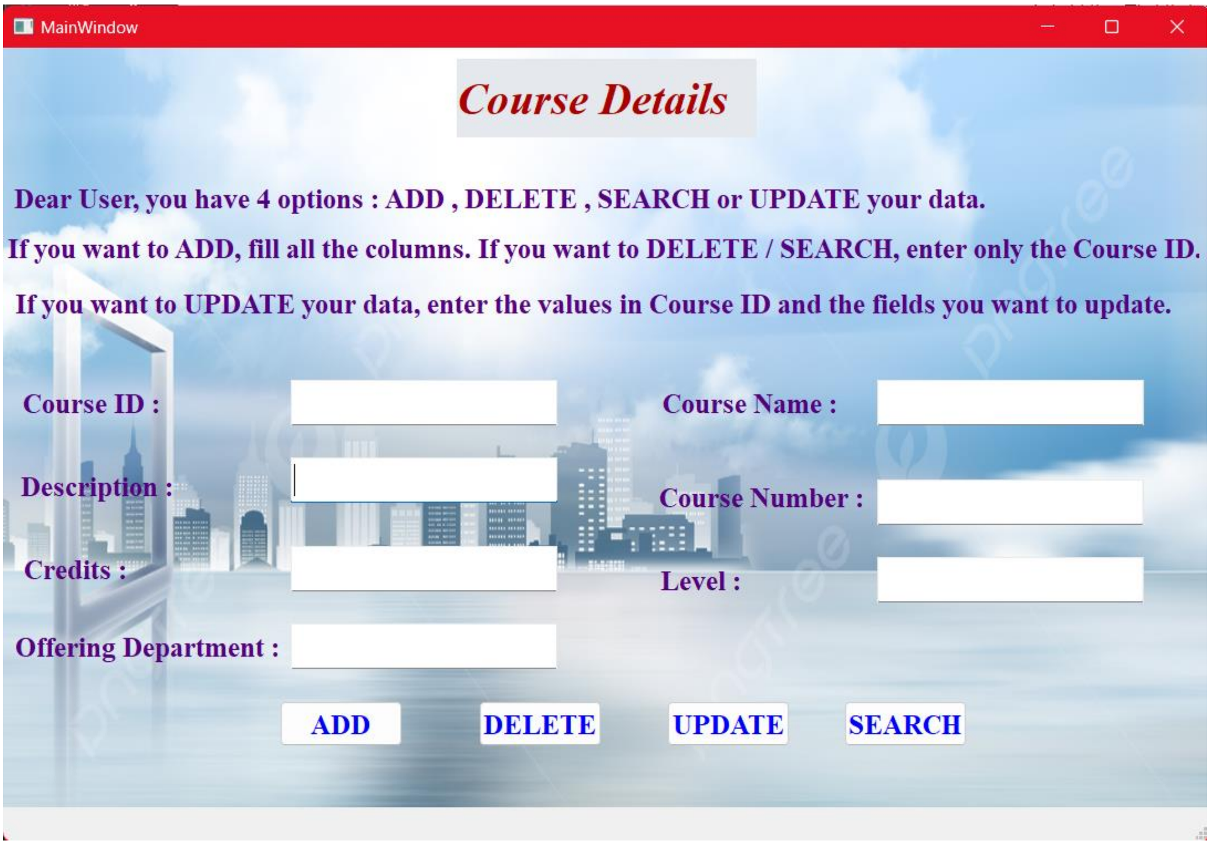
Name :

Specialization :

Department ID :

Peformance Metrics :

Property ID :



Course Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only the Course ID.

If you want to UPDATE your data, enter the values in Course ID and the fields you want to update.

Course ID :

Course Name :

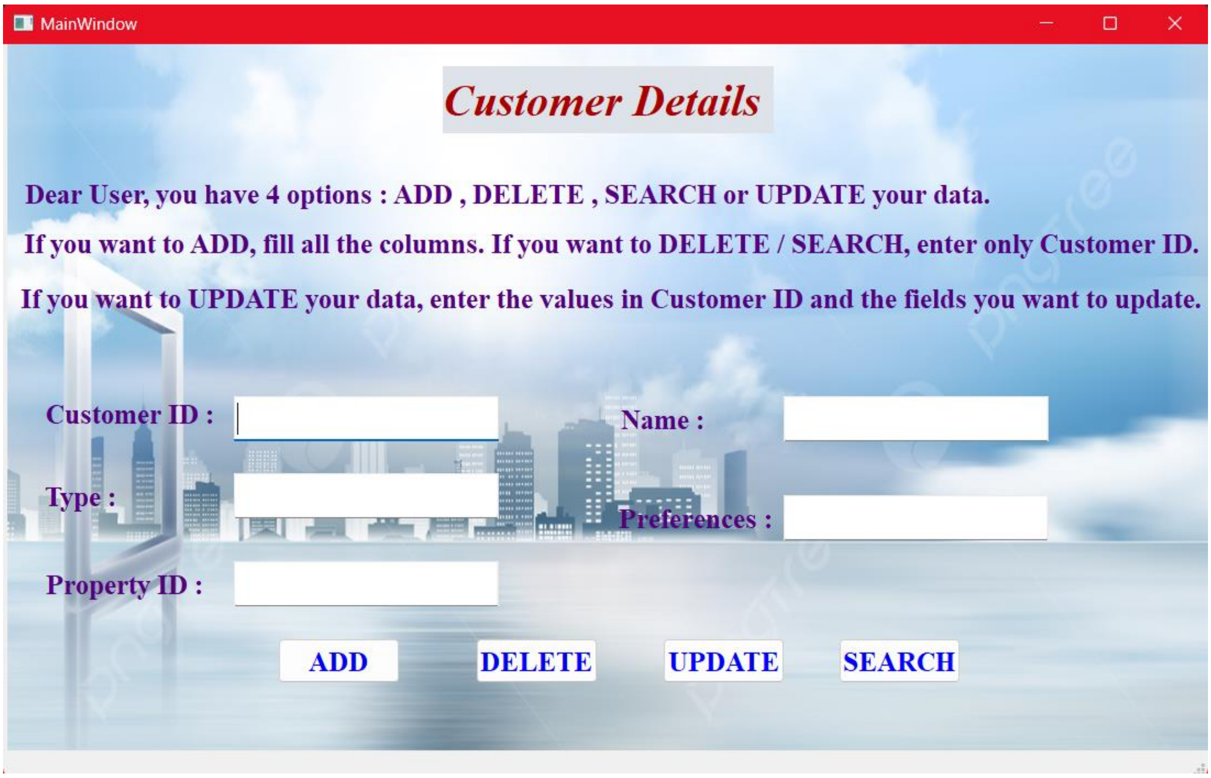
Description :

Course Number :

Credits :

Level :

Offering Department :



Customer Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Customer ID.

If you want to UPDATE your data, enter the values in Customer ID and the fields you want to update.

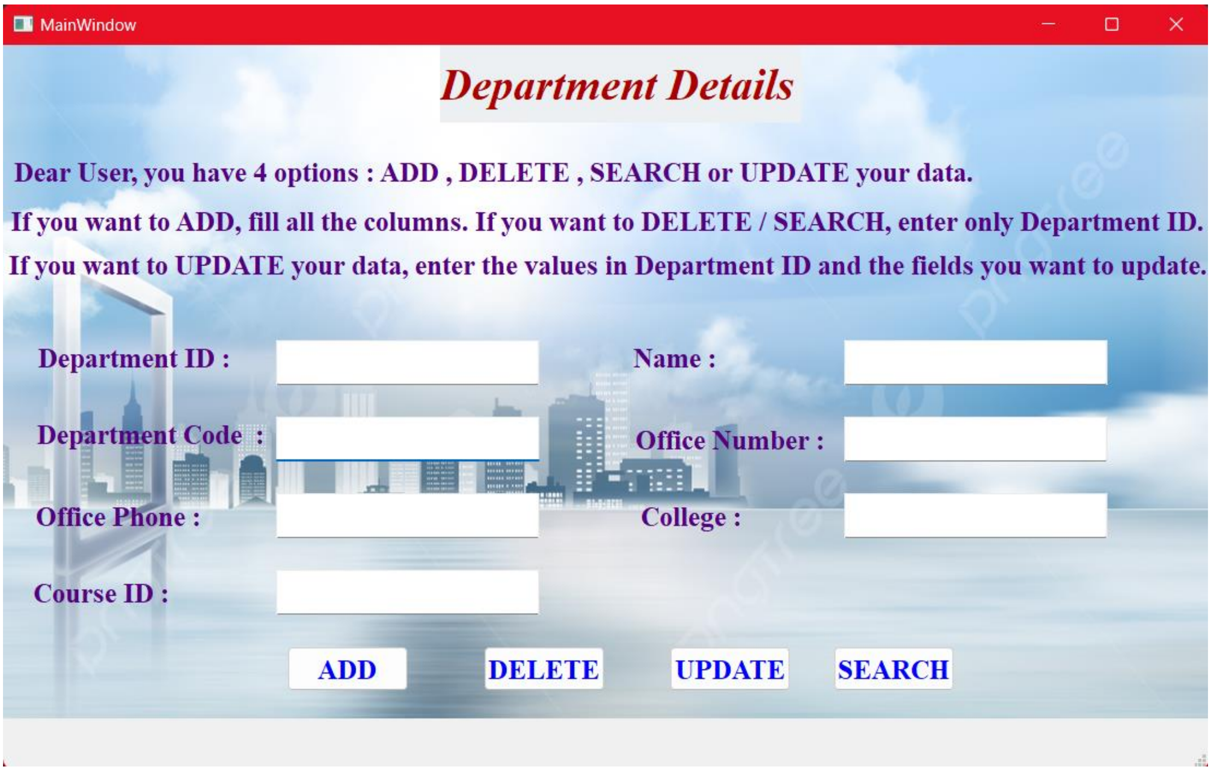
Customer ID :

Name :

Type :

Preferences :

Property ID :



Department Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Department ID.

If you want to UPDATE your data, enter the values in Department ID and the fields you want to update.

Department ID :

Name :

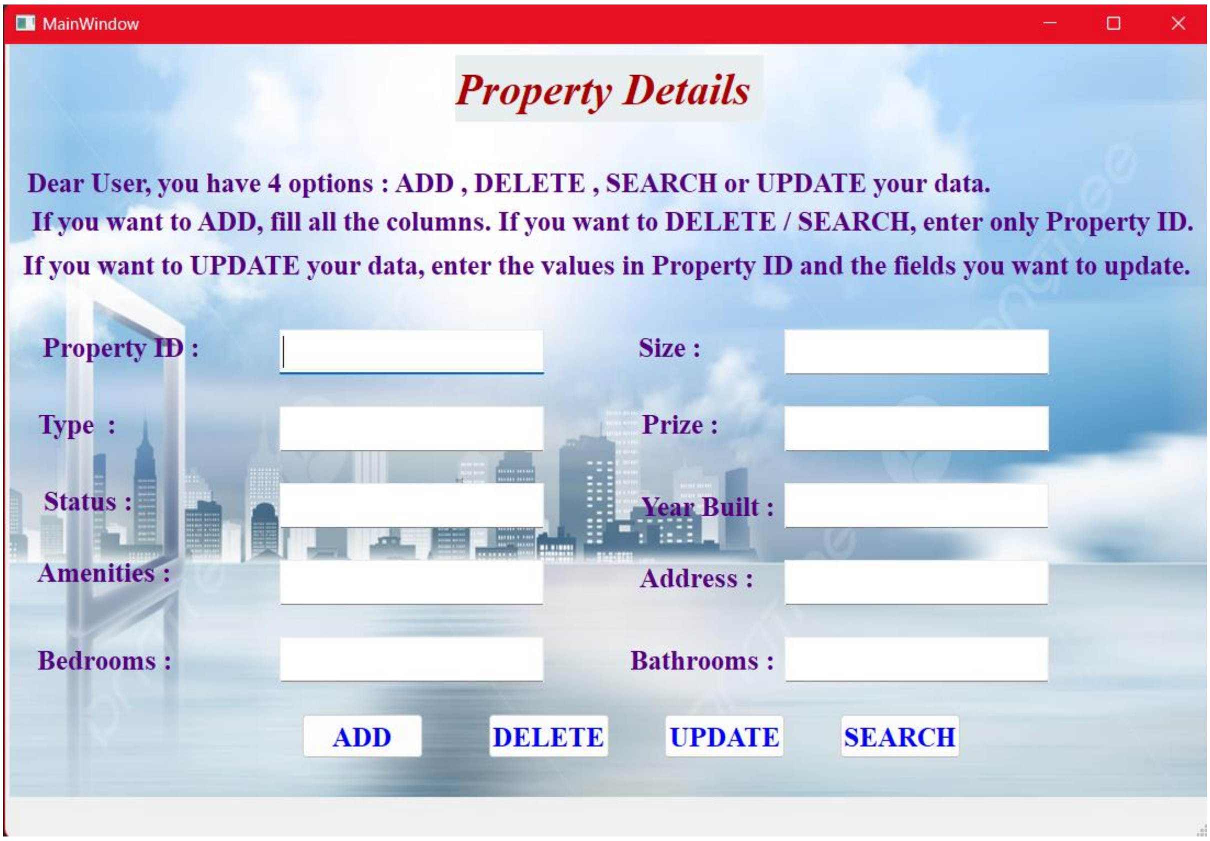
Department Code :

Office Number :

Office Phone :

College :

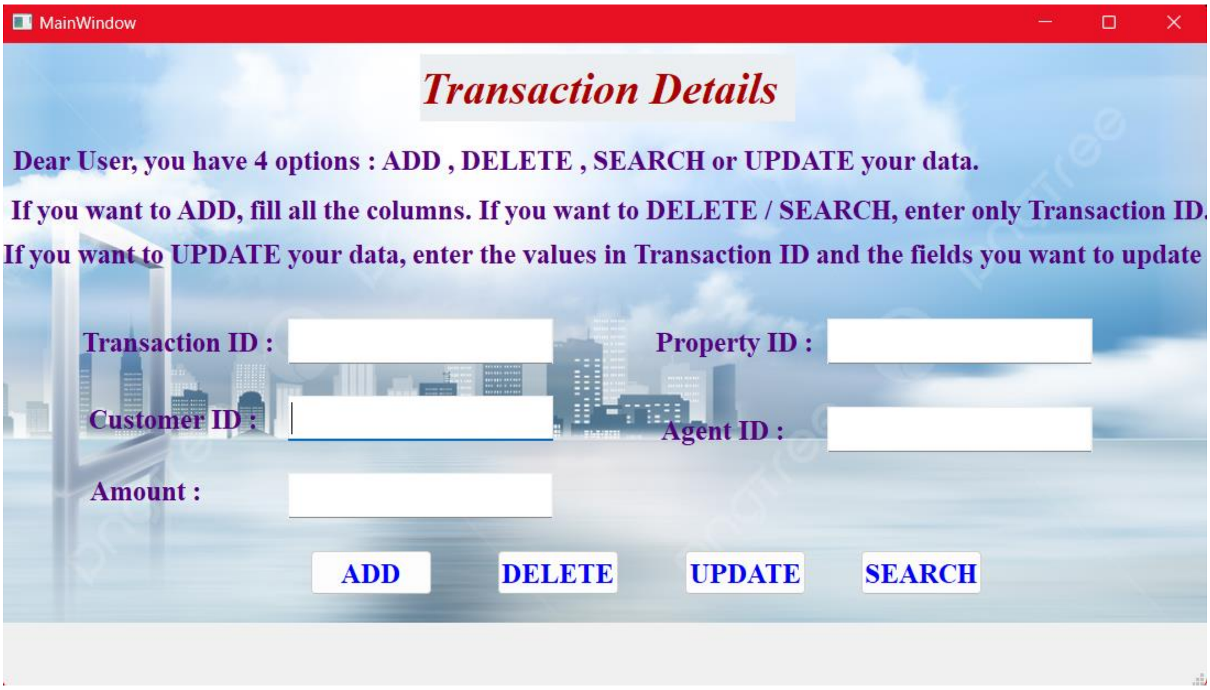
Course ID :



Property Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.
If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Property ID.
If you want to UPDATE your data, enter the values in Property ID and the fields you want to update.

Property ID :
Type :
Status :
Amenities :
Bedrooms :
Size :
Prize :
Year Built :
Address :
Bathrooms :



Transaction Details

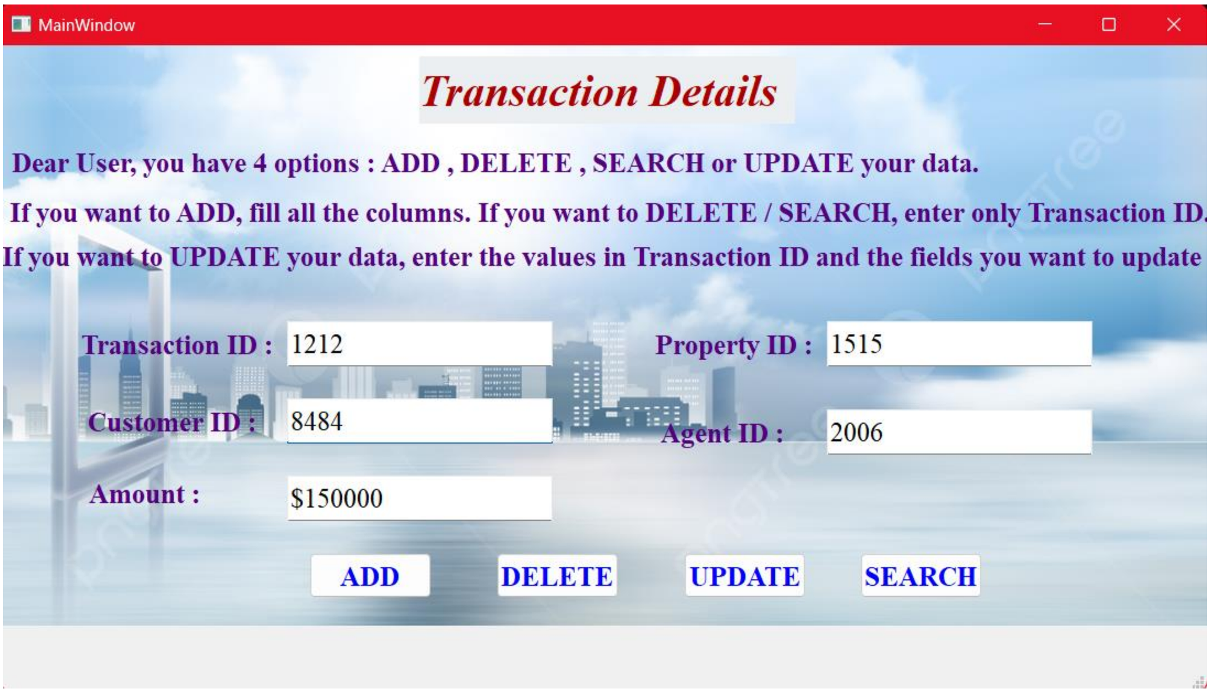
Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.
If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Transaction ID.
If you want to UPDATE your data, enter the values in Transaction ID and the fields you want to update

Transaction ID :
Customer ID :
Amount :
Property ID :
Agent ID :

As the interface suggests, there are total of 4 operations which can be performed, i.e., Insertion, Deletion, Updating and Searching.

Insertion –

For Insertion, an example in the UI –



Transaction Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Transaction ID.

If you want to UPDATE your data, enter the values in Transaction ID and the fields you want to update

Transaction ID : 1212

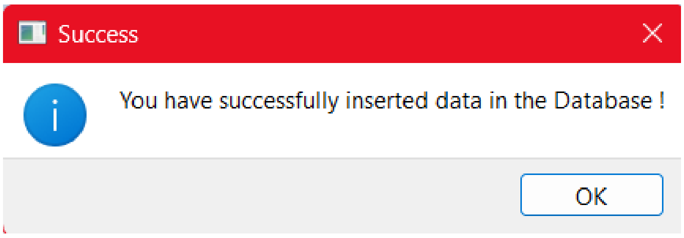
Property ID : 1515

Customer ID : 8484

Agent ID : 2006

Amount : $150000

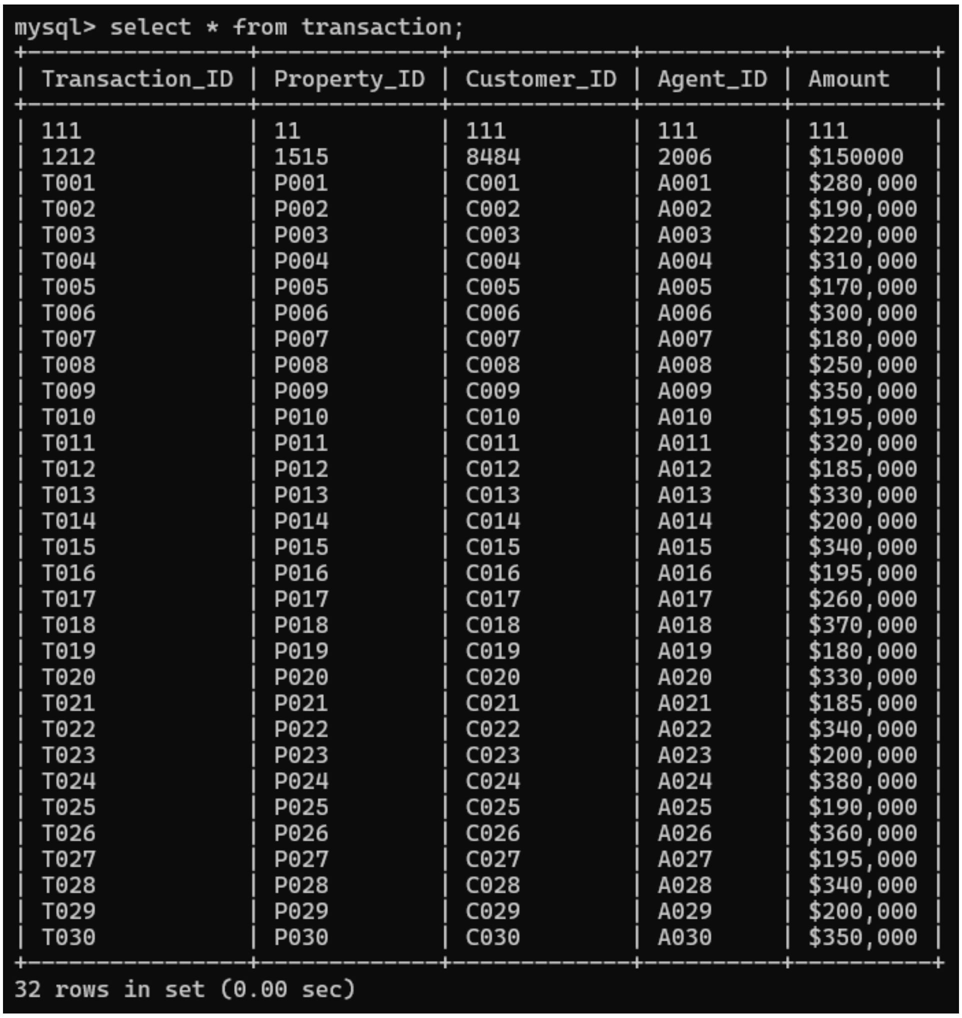
After insertion is successfully completed, a prompt occurs –



You have successfully inserted data in the Database !

After this the data is added in the Transaction table present in the database.

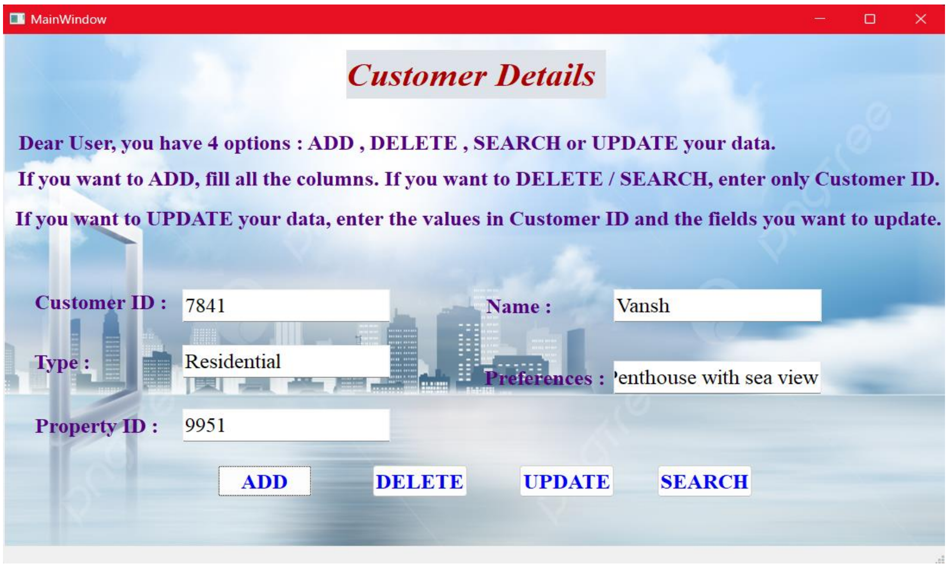
Transaction Table –



```
mysql> select * from transaction;
+----------------+-------------+-------------+----------+----------+
| Transaction_ID | Property_ID | Customer_ID | Agent_ID | Amount   |
+----------------+-------------+-------------+----------+----------+
| 111            | 11          | 111         | 111      | 111      |
| 1212           | 1515        | 8484        | 2006     | $150000  |
| T001           | P001        | C001        | A001     | $280,000 |
| T002           | P002        | C002        | A002     | $190,000 |
| T003           | P003        | C003        | A003     | $220,000 |
| T004           | P004        | C004        | A004     | $310,000 |
| T005           | P005        | C005        | A005     | $170,000 |
| T006           | P006        | C006        | A006     | $300,000 |
| T007           | P007        | C007        | A007     | $180,000 |
| T008           | P008        | C008        | A008     | $250,000 |
| T009           | P009        | C009        | A009     | $350,000 |
| T010           | P010        | C010        | A010     | $195,000 |
| T011           | P011        | C011        | A011     | $320,000 |
| T012           | P012        | C012        | A012     | $185,000 |
| T013           | P013        | C013        | A013     | $330,000 |
| T014           | P014        | C014        | A014     | $200,000 |
| T015           | P015        | C015        | A015     | $340,000 |
| T016           | P016        | C016        | A016     | $195,000 |
| T017           | P017        | C017        | A017     | $260,000 |
| T018           | P018        | C018        | A018     | $370,000 |
| T019           | P019        | C019        | A019     | $180,000 |
| T020           | P020        | C020        | A020     | $330,000 |
| T021           | P021        | C021        | A021     | $185,000 |
| T022           | P022        | C022        | A022     | $340,000 |
| T023           | P023        | C023        | A023     | $200,000 |
| T024           | P024        | C024        | A024     | $380,000 |
| T025           | P025        | C025        | A025     | $190,000 |
| T026           | P026        | C026        | A026     | $360,000 |
| T027           | P027        | C027        | A027     | $195,000 |
| T028           | P028        | C028        | A028     | $340,000 |
| T029           | P029        | C029        | A029     | $200,000 |
| T030           | P030        | C030        | A030     | $350,000 |
+----------------+-------------+-------------+----------+----------+
32 rows in set (0.00 sec)
```

Inserted Data

Another data insertion example –



Customer Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Customer ID.

If you want to UPDATE your data, enter the values in Customer ID and the fields you want to update.

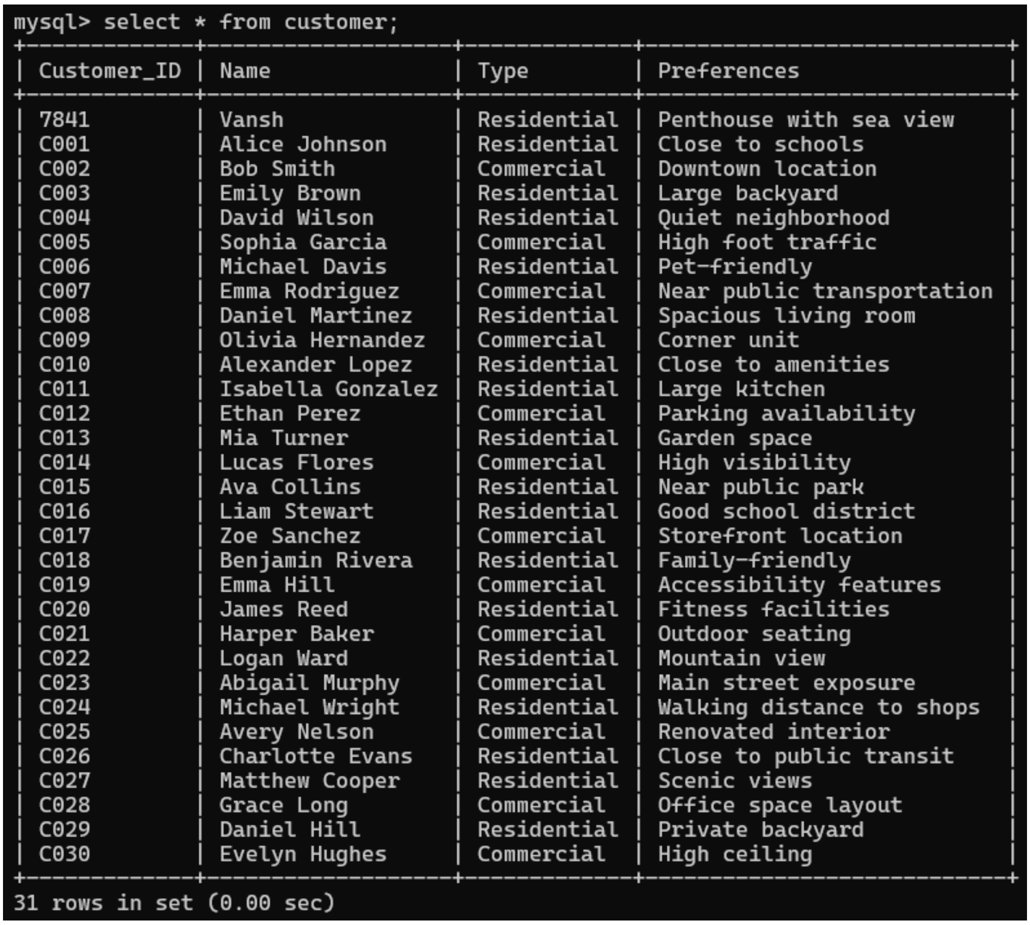
Customer ID : 7841

Name : Vansh

Type : Residential

Preferences : 'enthouse with sea view

Property ID : 9951

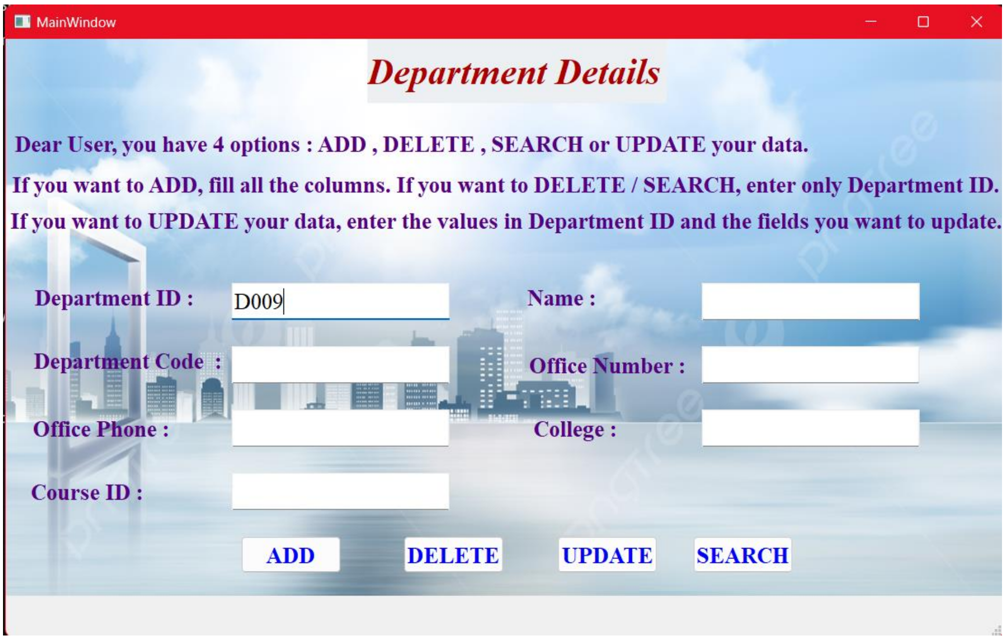


```
mysql> select * from customer;
+-------------+-------------------+-------------+----------------------------+
| Customer_ID | Name              | Type        | Preferences                |
+-------------+-------------------+-------------+----------------------------+
| 7841        | Vansh             | Residential | Penthouse with sea view    |
| C001        | Alice Johnson     | Residential | Close to schools           |
| C002        | Bob Smith         | Commercial  | Downtown location          |
| C003        | Emily Brown       | Residential | Large backyard             |
| C004        | David Wilson      | Residential | Quiet neighborhood         |
| C005        | Sophia Garcia     | Commercial  | High foot traffic          |
| C006        | Michael Davis     | Residential | Pet-friendly               |
| C007        | Emma Rodriguez    | Commercial  | Near public transportation |
| C008        | Daniel Martinez   | Residential | Spacious living room       |
| C009        | Olivia Hernandez  | Commercial  | Corner unit                |
| C010        | Alexander Lopez   | Residential | Close to amenities         |
| C011        | Isabella Gonzalez | Residential | Large kitchen              |
| C012        | Ethan Perez       | Commercial  | Parking availability       |
| C013        | Mia Turner        | Residential | Garden space               |
| C014        | Lucas Flores      | Commercial  | High visibility            |
| C015        | Ava Collins       | Residential | Near public park           |
| C016        | Liam Stewart      | Residential | Good school district       |
| C017        | Zoe Sanchez       | Commercial  | Storefront location        |
| C018        | Benjamin Rivera   | Residential | Family-friendly            |
| C019        | Emma Hill         | Commercial  | Accessibility features     |
| C020        | James Reed        | Residential | Fitness facilities         |
| C021        | Harper Baker      | Commercial  | Outdoor seating            |
| C022        | Logan Ward        | Residential | Mountain view              |
| C023        | Abigail Murphy    | Commercial  | Main street exposure       |
| C024        | Michael Wright    | Residential | Walking distance to shops  |
| C025        | Avery Nelson      | Commercial  | Renovated interior         |
| C026        | Charlotte Evans   | Residential | Close to public transit    |
| C027        | Matthew Cooper    | Residential | Scenic views               |
| C028        | Grace Long        | Commercial  | Office space layout        |
| C029        | Daniel Hill       | Residential | Private backyard           |
| C030        | Evelyn Hughes     | Commercial  | High ceiling               |
+-------------+-------------------+-------------+----------------------------+
31 rows in set (0.00 sec)
```

Inserted Data

Deletion –

Now for Data Deletion Example, for data deletion, only the first input has to be given in the input field (which is the primary key of the table).



Department Details

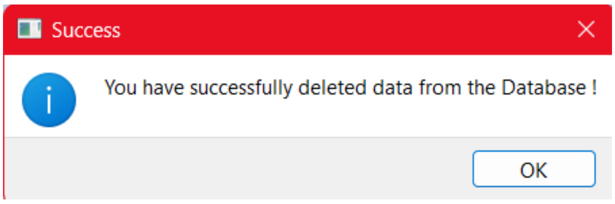
Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Department ID.

If you want to UPDATE your data, enter the values in Department ID and the fields you want to update.

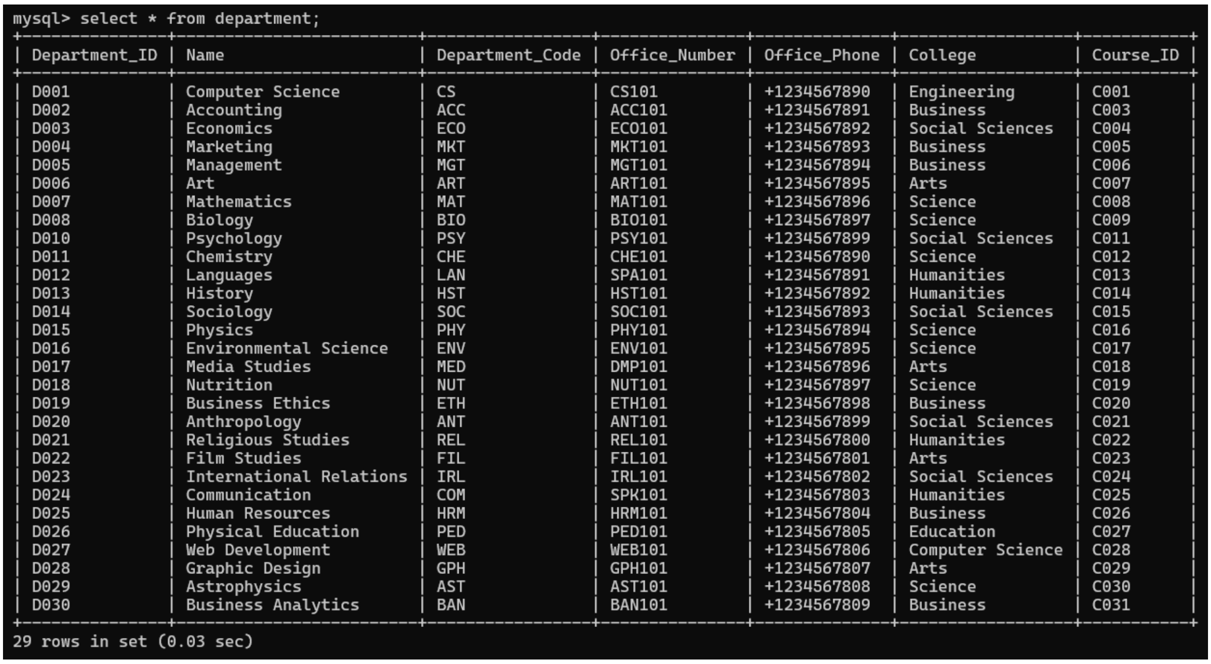
Department ID :	D009	Name :	
Department Code :		Office Number :	
Office Phone :		College :	
Course ID :			

After successful deletion, this prompt occurs –



You have successfully deleted data from the Database !

Department Table –

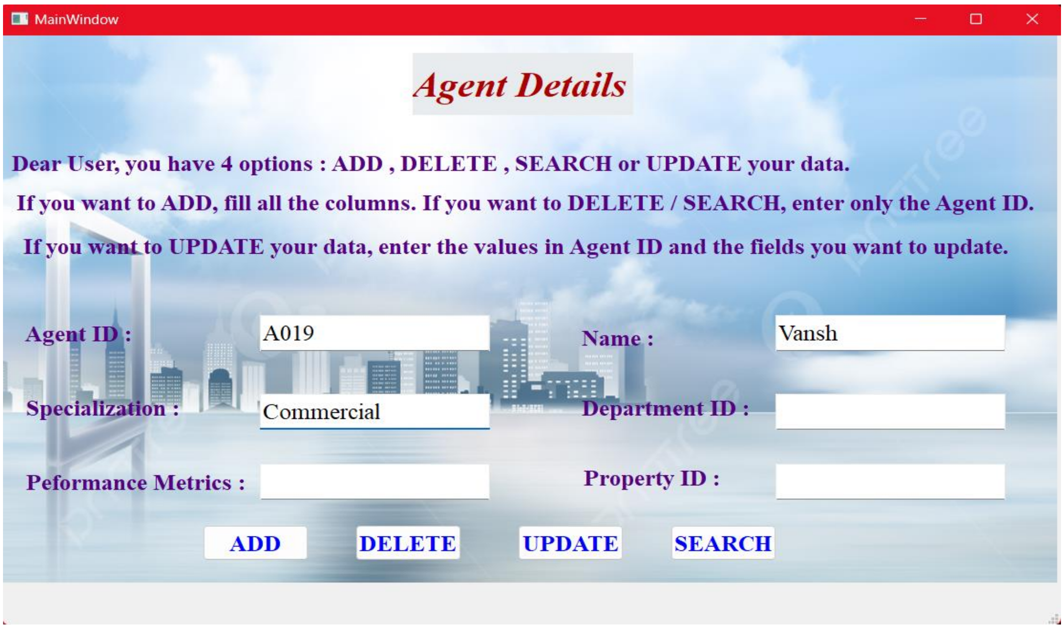


```
mysql> select * from department;
+---------------+-------------------------+-----------------+---------------+--------------+------------------+-----------+
| Department_ID | Name                    | Department_Code | Office_Number | Office_Phone | College          | Course_ID |
+---------------+-------------------------+-----------------+---------------+--------------+------------------+-----------+
| D001          | Computer Science        | CS              | CS101         | +1234567890  | Engineering      | C001      |
| D002          | Accounting              | ACC             | ACC101        | +1234567891  | Business         | C003      |
| D003          | Economics               | ECO             | ECO101        | +1234567892  | Social Sciences  | C004      |
| D004          | Marketing               | MKT             | MKT101        | +1234567893  | Business         | C005      |
| D005          | Management              | MGT             | MGT101        | +1234567894  | Business         | C006      |
| D006          | Art                     | ART             | ART101        | +1234567895  | Arts             | C007      |
| D007          | Mathematics             | MAT             | MAT101        | +1234567896  | Science          | C008      |
| D008          | Biology                 | BIO             | BIO101        | +1234567897  | Science          | C009      |
| D010          | Psychology              | PSY             | PSY101        | +1234567899  | Social Sciences  | C011      |
| D011          | Chemistry               | CHE             | CHE101        | +1234567890  | Science          | C012      |
| D012          | Languages               | LAN             | SPA101        | +1234567891  | Humanities       | C013      |
| D013          | History                 | HST             | HST101        | +1234567892  | Humanities       | C014      |
| D014          | Sociology               | SOC             | SOC101        | +1234567893  | Social Sciences  | C015      |
| D015          | Physics                 | PHY             | PHY101        | +1234567894  | Science          | C016      |
| D016          | Environmental Science   | ENV             | ENV101        | +1234567895  | Science          | C017      |
| D017          | Media Studies           | MED             | DMP101        | +1234567896  | Arts             | C018      |
| D018          | Nutrition               | NUT             | NUT101        | +1234567897  | Science          | C019      |
| D019          | Business Ethics         | ETH             | ETH101        | +1234567898  | Business         | C020      |
| D020          | Anthropology            | ANT             | ANT101        | +1234567899  | Social Sciences  | C021      |
| D021          | Religious Studies       | REL             | REL101        | +1234567800  | Humanities       | C022      |
| D022          | Film Studies            | FIL             | FIL101        | +1234567801  | Arts             | C023      |
| D023          | International Relations | IRL             | IRL101        | +1234567802  | Social Sciences  | C024      |
| D024          | Communication           | COM             | SPK101        | +1234567803  | Humanities       | C025      |
| D025          | Human Resources         | HRM             | HRM101        | +1234567804  | Business         | C026      |
| D026          | Physical Education      | PED             | PED101        | +1234567805  | Education        | C027      |
| D027          | Web Development         | WEB             | WEB101        | +1234567806  | Computer Science | C028      |
| D028          | Graphic Design          | GPH             | GPH101        | +1234567807  | Arts             | C029      |
| D029          | Astrophysics            | AST             | AST101        | +1234567808  | Science          | C030      |
| D030          | Business Analytics      | BAN             | BAN101        | +1234567809  | Business         | C031      |
+---------------+-------------------------+-----------------+---------------+--------------+------------------+-----------+
29 rows in set (0.03 sec)
```

– **D009 is missing** – **(successfully deleted)**

Updating –

Now, an example for updating the records, the first input field should be filled and whichever field that you want to update should be filled –



Agent Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only the Agent ID.

If you want to UPDATE your data, enter the values in Agent ID and the fields you want to update.

Agent ID : A019

Name : Vansh

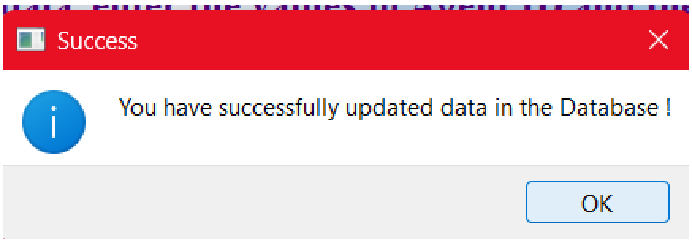
Specialization : Commercial

Department ID :

Peformance Metrics :

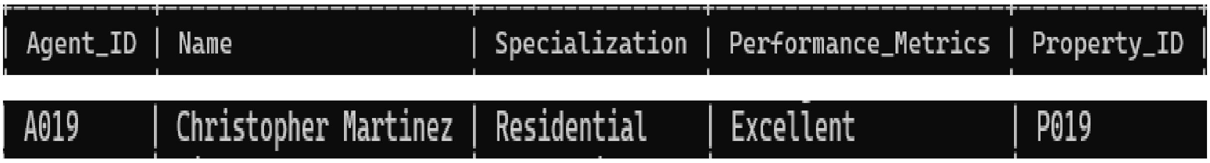
Property ID :

After successfully updating, this prompt occurs –



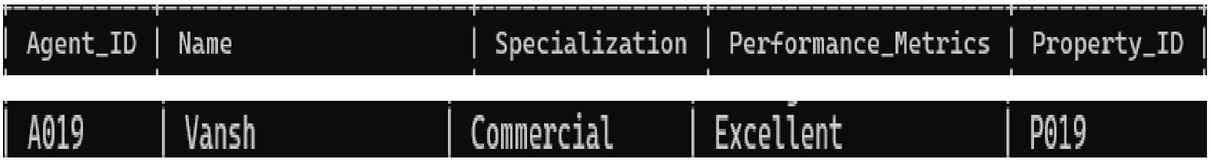
You have successfully updated data in the Database !

Record before updating –



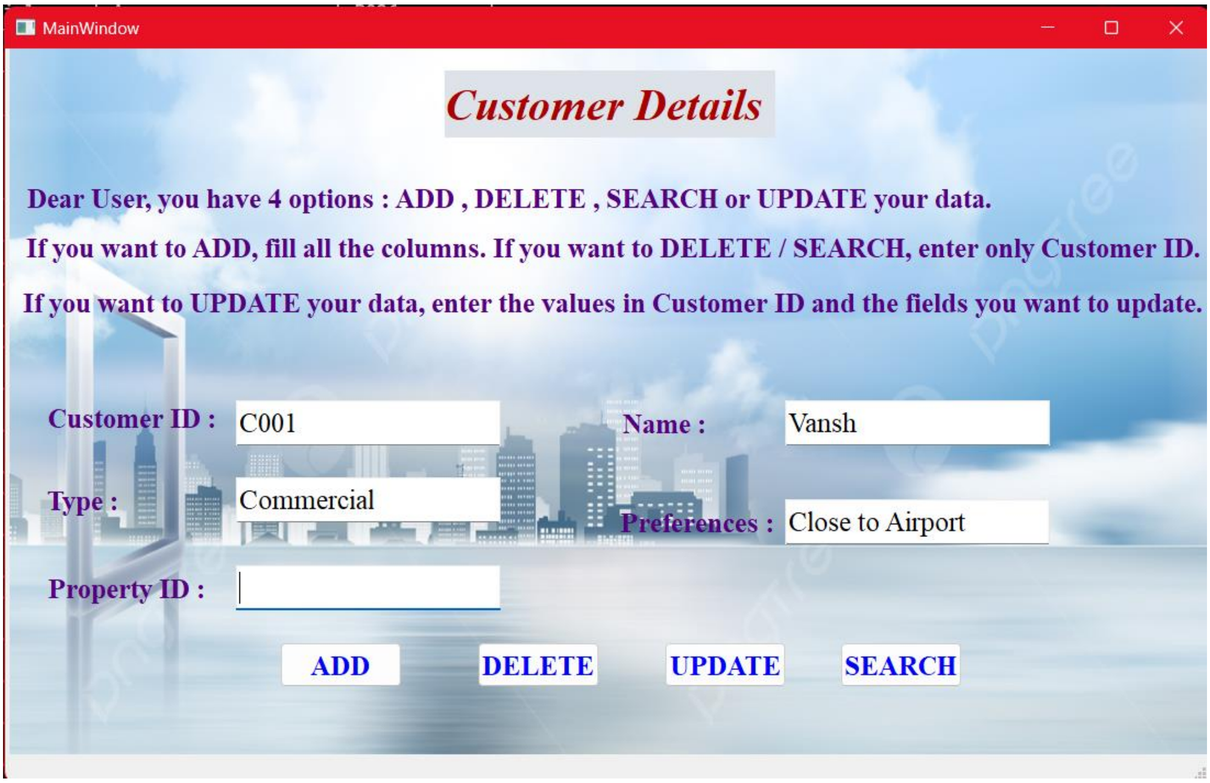
```
| Agent_ID | Name                 | Specialization | Performance_Metrics | Property_ID |
  A019     | Christopher Martinez | Residential    | Excellent           | P019
```

After updating –



```
| Agent_ID | Name                 | Specialization | Performance_Metrics | Property_ID |
| A019     | Vansh                | Commercial     | Excellent           | P019
```

Another example,



Customer Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Customer ID.

If you want to UPDATE your data, enter the values in Customer ID and the fields you want to update.

Customer ID : C001

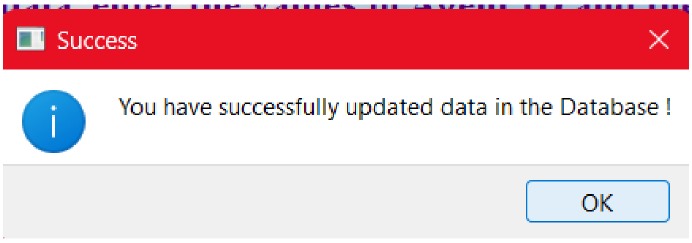
Name : Vansh

Type : Commercial

Preferences : Close to Airport

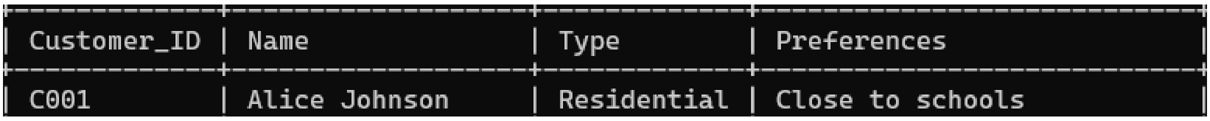
Property ID :

After successfully updating, this prompt occurs –



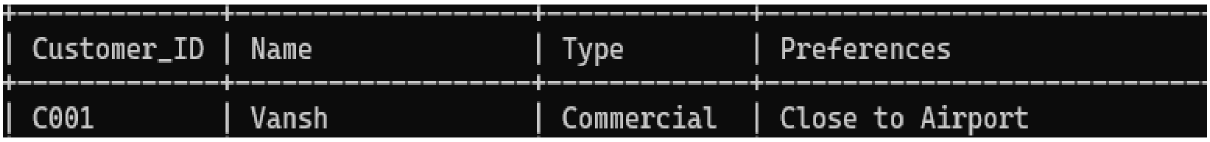
You have successfully updated data in the Database !

Record before updating –



```
+-------------+-------------------+-------------+------------------------+
| Customer_ID | Name              | Type        | Preferences            |
+-------------+-------------------+-------------+------------------------+
| C001        | Alice Johnson     | Residential | Close to schools       |
```

After updating –

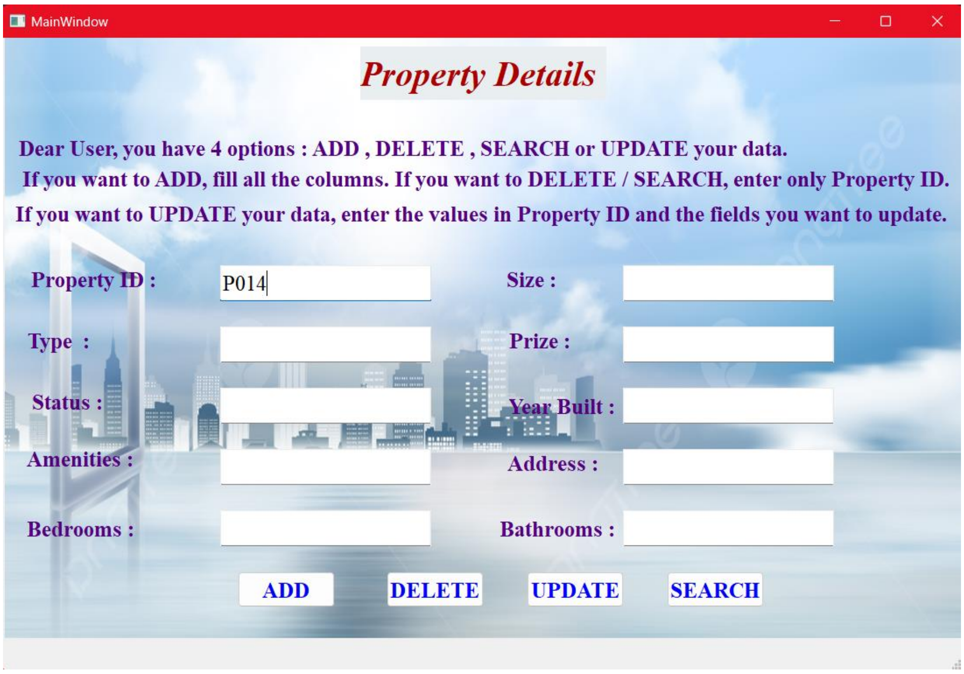


```
+-------------+-------------------+-------------+------------------------+
| Customer_ID | Name              | Type        | Preferences            |
+-------------+-------------------+-------------+------------------------+
| C001        | Vansh             | Commercial  | Close to Airport       |
```

Searching –

Now, for searching any record in the database, you just have to fill the first input field and the resultant record will be displayed.

Example –



Property Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.
If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Property ID.
If you want to UPDATE your data, enter the values in Property ID and the fields you want to update.

Property ID : P014

Type :

Status :

Amenities :

Bedrooms :

Size :

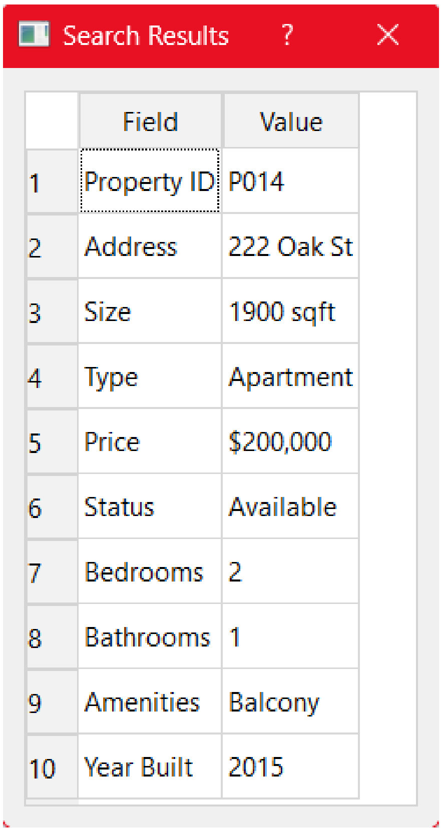
Prize :

Year Built :

Address :

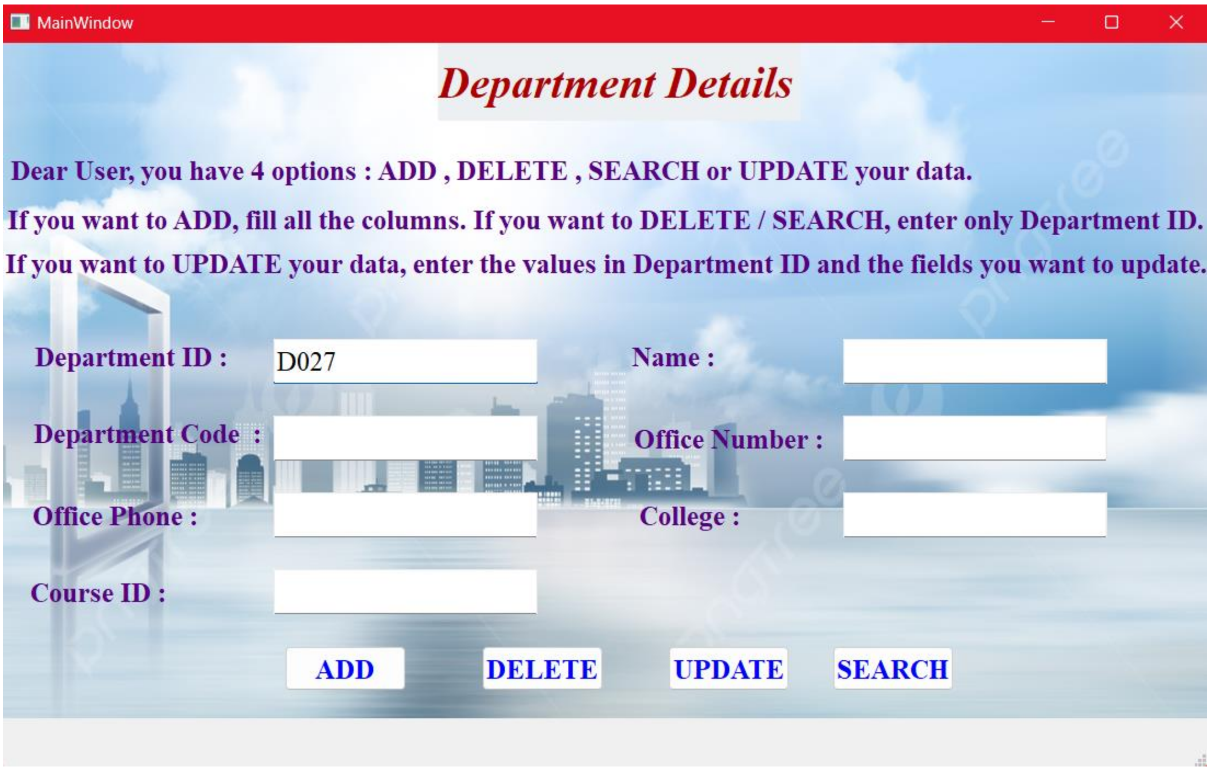
Bathrooms :

After successful searching,



	Field	Value
1	Property ID	P014
2	Address	222 Oak St
3	Size	1900 sqft
4	Type	Apartment
5	Price	$200,000
6	Status	Available
7	Bedrooms	2
8	Bathrooms	1
9	Amenities	Balcony
10	Year Built	2015

Another Example –



Department Details

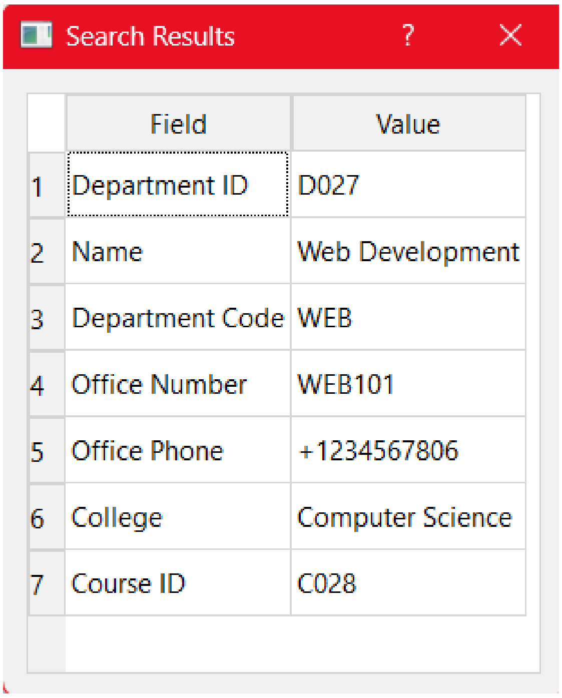
Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Department ID.

If you want to UPDATE your data, enter the values in Department ID and the fields you want to update.

Department ID :	D027	Name :	
Department Code :		Office Number :	
Office Phone :		College :	
Course ID :			

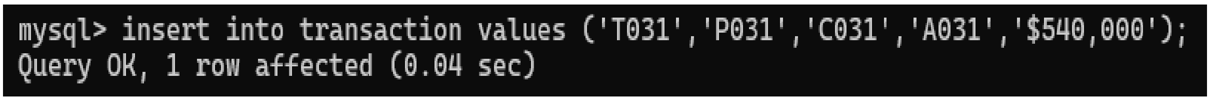
After successful searching,



	Field	Value
1	Department ID	D027
2	Name	Web Development
3	Department Code	WEB
4	Office Number	WEB101
5	Office Phone	+1234567806
6	College	Computer Science
7	Course ID	C028

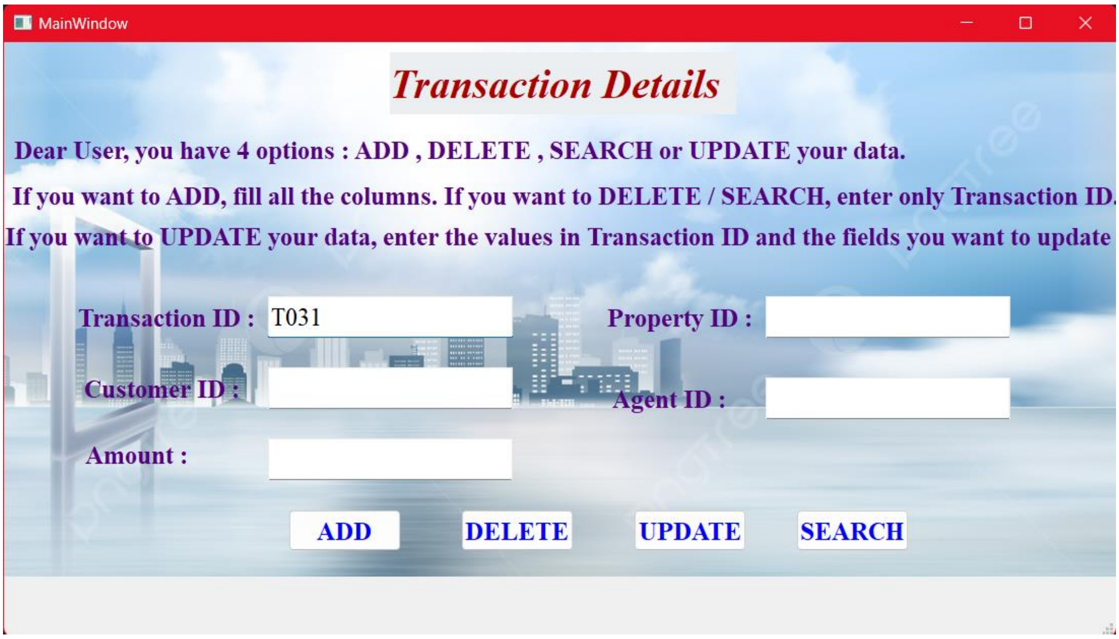
Also, if there is a change in the database, then also we are able to search the record through the UI –

Change in table –



```
mysql> insert into transaction values ('T031','P031','C031','A031','$540,000');
Query OK, 1 row affected (0.04 sec)
```

Record shown through UI –



Transaction Details

Dear User, you have 4 options : ADD , DELETE , SEARCH or UPDATE your data.

If you want to ADD, fill all the columns. If you want to DELETE / SEARCH, enter only Transaction ID.

If you want to UPDATE your data, enter the values in Transaction ID and the fields you want to update

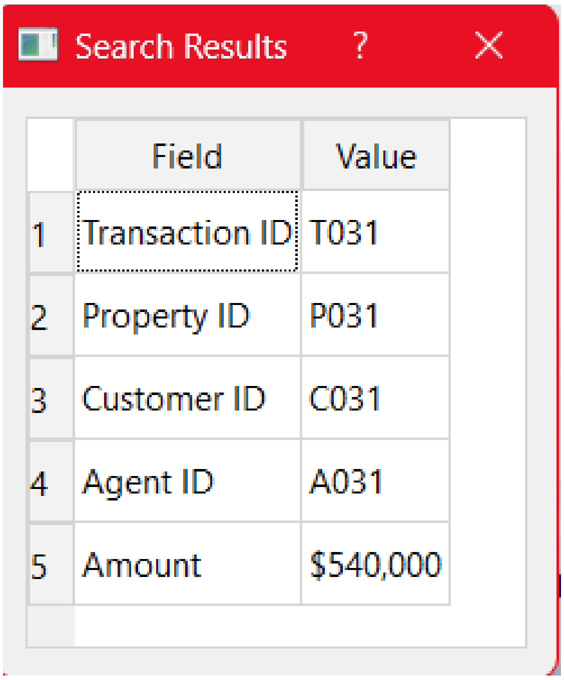
Transaction ID : T031

Property ID :

Customer ID :

Agent ID :

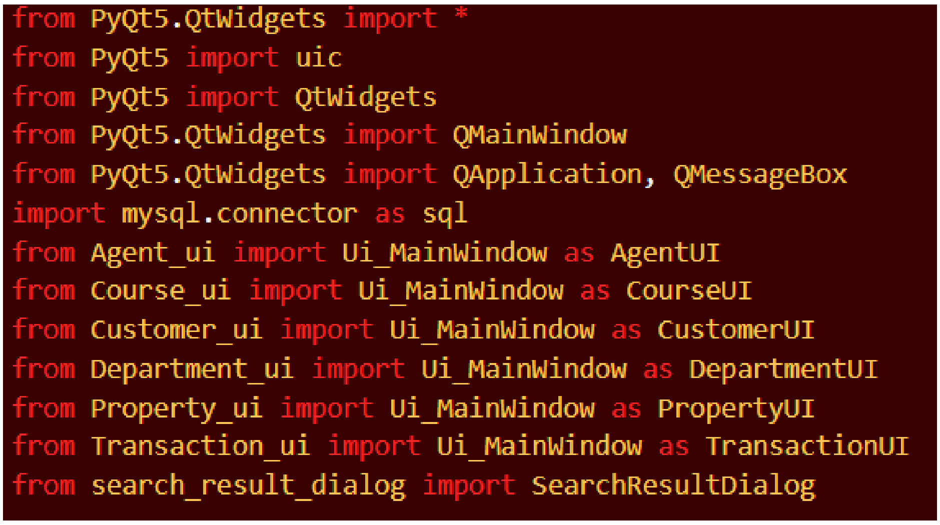
Amount :



	Field	Value
1	Transaction ID	T031
2	Property ID	P031
3	Customer ID	C031
4	Agent ID	A031
5	Amount	$540,000

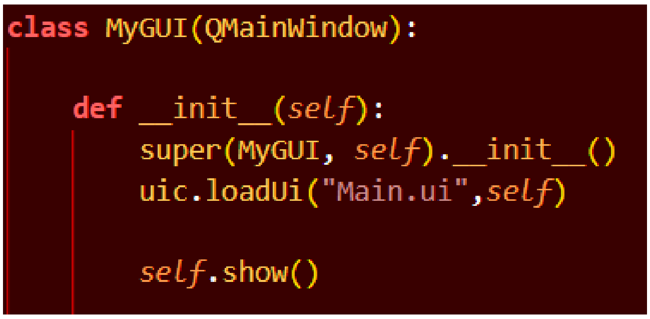
Code –

In the project code, I have imported many libraries and their elements which include MySQL connector with python, PyQT5 designer and the code files of the all the User Interfaces, below is code snippet of the same –



```
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QMainWindow
from PyQt5.QtWidgets import QApplication, QMessageBox
import mysql.connector as sql
from Agent_ui import Ui_MainWindow as AgentUI
from Course_ui import Ui_MainWindow as CourseUI
from Customer_ui import Ui_MainWindow as CustomerUI
from Department_ui import Ui_MainWindow as DepartmentUI
from Property_ui import Ui_MainWindow as PropertyUI
from Transaction_ui import Ui_MainWindow as TransactionUI
from search_result_dialog import SearchResultDialog
```

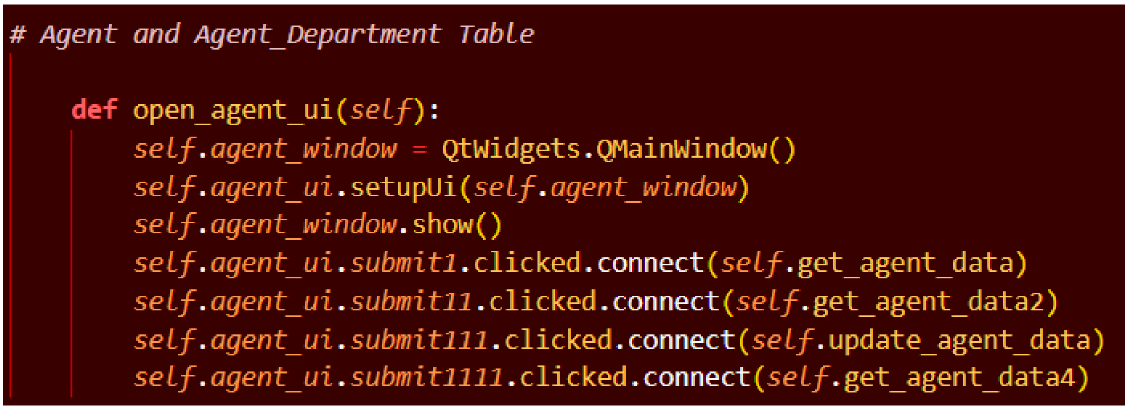
Now, code snippet for opening the main UI at the start –



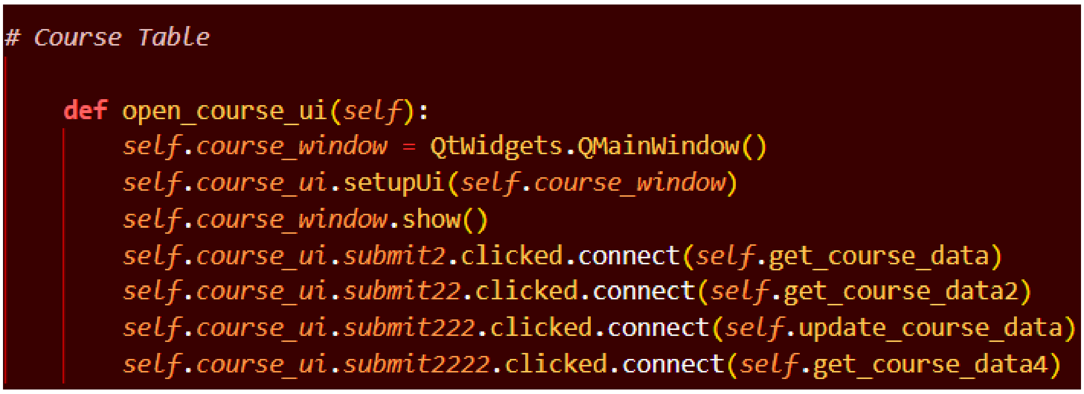
```
class MyGUI(QMainWindow):

    def __init__(self):
        super(MyGUI, self).__init__()
        uic.loadUi("Main.ui",self)

        self.show()
```

where self is the Main UI.

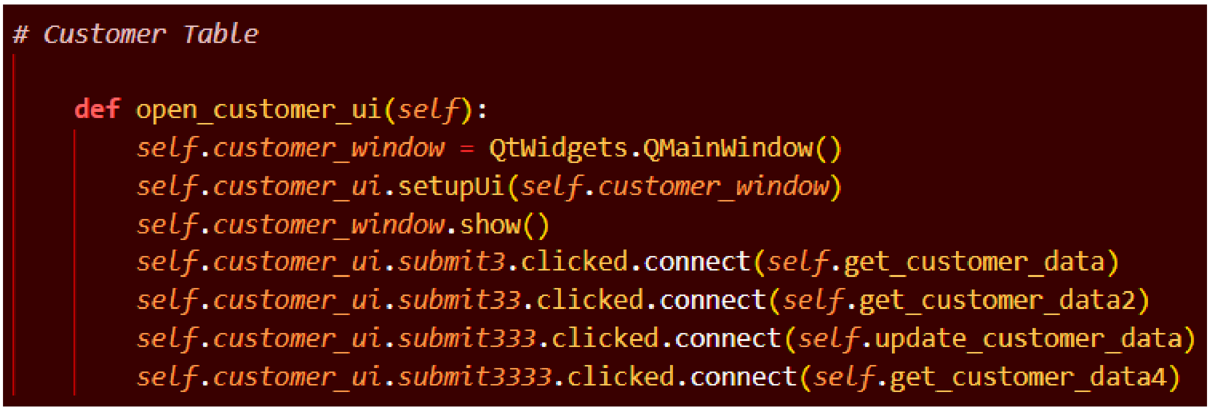
Now below are code snippets of all the UI openings after clicking the buttons for the same –



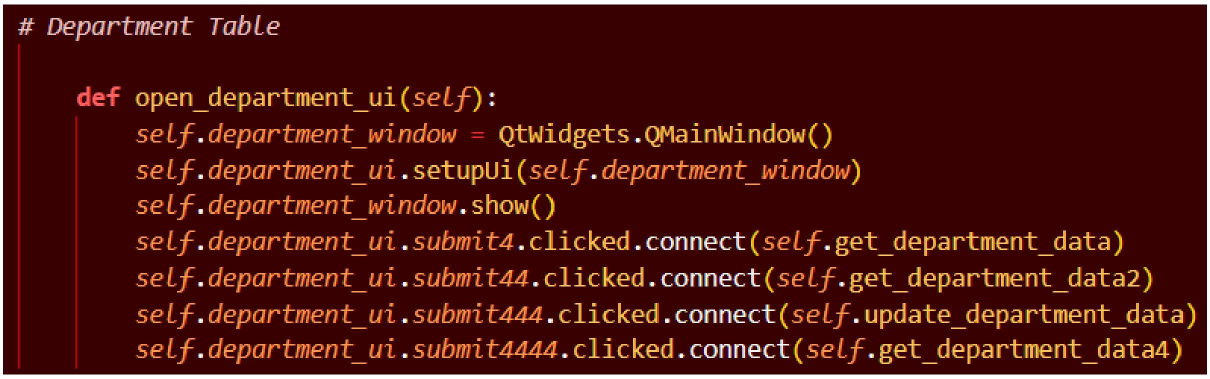
```
# Agent and Agent_Department Table

    def open_agent_ui(self):
        self.agent_window = QtWidgets.QMainWindow()
        self.agent_ui.setupUi(self.agent_window)
        self.agent_window.show()
        self.agent_ui.submit1.clicked.connect(self.get_agent_data)
        self.agent_ui.submit11.clicked.connect(self.get_agent_data2)
        self.agent_ui.submit111.clicked.connect(self.update_agent_data)
        self.agent_ui.submit1111.clicked.connect(self.get_agent_data4)
```



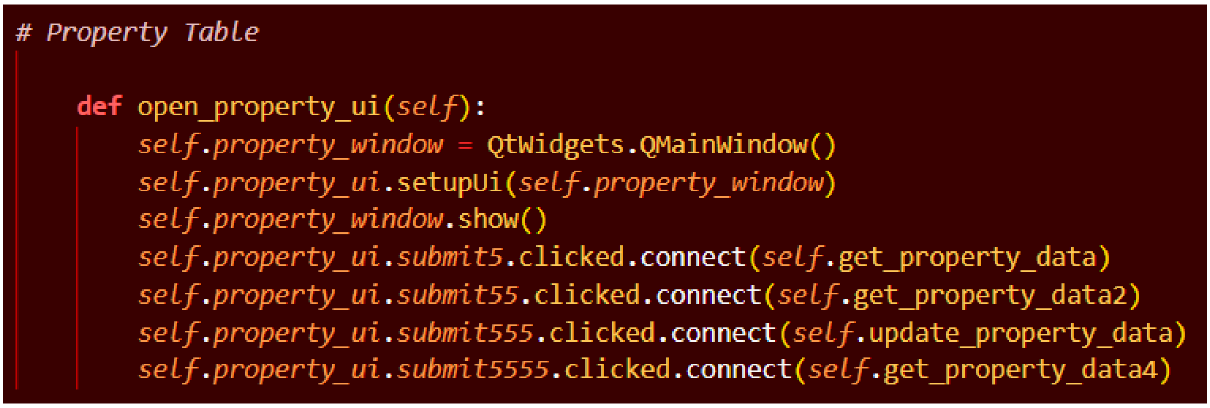
```
# Course Table

    def open_course_ui(self):
        self.course_window = QtWidgets.QMainWindow()
        self.course_ui.setupUi(self.course_window)
        self.course_window.show()
        self.course_ui.submit2.clicked.connect(self.get_course_data)
        self.course_ui.submit22.clicked.connect(self.get_course_data2)
        self.course_ui.submit222.clicked.connect(self.update_course_data)
        self.course_ui.submit2222.clicked.connect(self.get_course_data4)
```



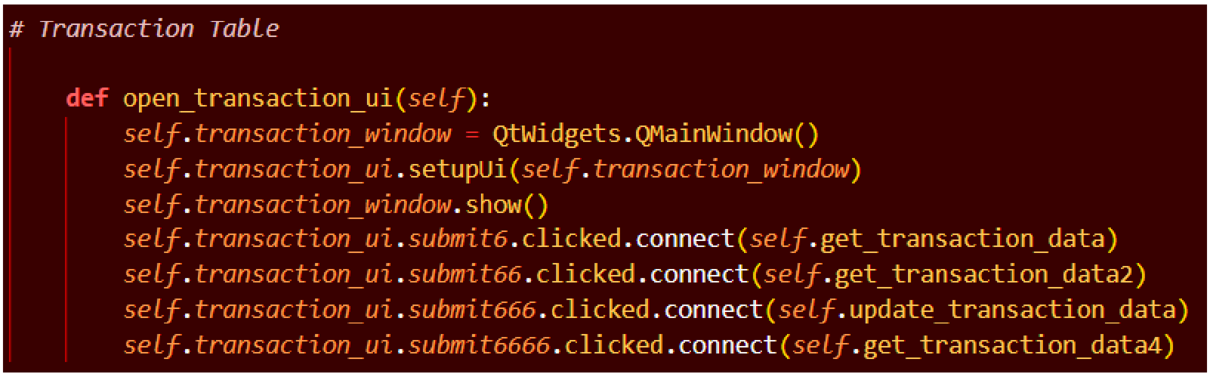
```
# Customer Table

    def open_customer_ui(self):
        self.customer_window = QtWidgets.QMainWindow()
        self.customer_ui.setupUi(self.customer_window)
        self.customer_window.show()
        self.customer_ui.submit3.clicked.connect(self.get_customer_data)
        self.customer_ui.submit33.clicked.connect(self.get_customer_data2)
        self.customer_ui.submit333.clicked.connect(self.update_customer_data)
        self.customer_ui.submit3333.clicked.connect(self.get_customer_data4)
```



```
# Department Table

    def open_department_ui(self):
        self.department_window = QtWidgets.QMainWindow()
        self.department_ui.setupUi(self.department_window)
        self.department_window.show()
        self.department_ui.submit4.clicked.connect(self.get_department_data)
        self.department_ui.submit44.clicked.connect(self.get_department_data2)
        self.department_ui.submit444.clicked.connect(self.update_department_data)
        self.department_ui.submit4444.clicked.connect(self.get_department_data4)
```



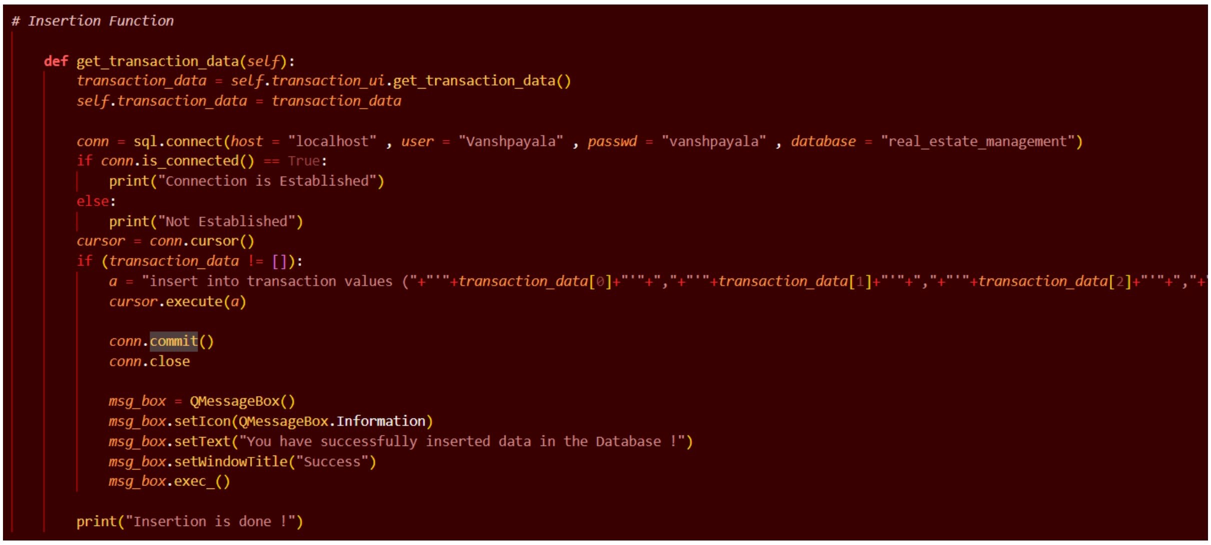
```
# Property Table

    def open_property_ui(self):
        self.property_window = QtWidgets.QMainWindow()
        self.property_ui.setupUi(self.property_window)
        self.property_window.show()
        self.property_ui.submit5.clicked.connect(self.get_property_data)
        self.property_ui.submit55.clicked.connect(self.get_property_data2)
        self.property_ui.submit555.clicked.connect(self.update_property_data)
        self.property_ui.submit5555.clicked.connect(self.get_property_data4)
```



```python
# Transaction Table

    def open_transaction_ui(self):
        self.transaction_window = QtWidgets.QMainWindow()
        self.transaction_ui.setupUi(self.transaction_window)
        self.transaction_window.show()
        self.transaction_ui.submit6.clicked.connect(self.get_transaction_data)
        self.transaction_ui.submit66.clicked.connect(self.get_transaction_data2)
        self.transaction_ui.submit666.clicked.connect(self.update_transaction_data)
        self.transaction_ui.submit6666.clicked.connect(self.get_transaction_data4)
```

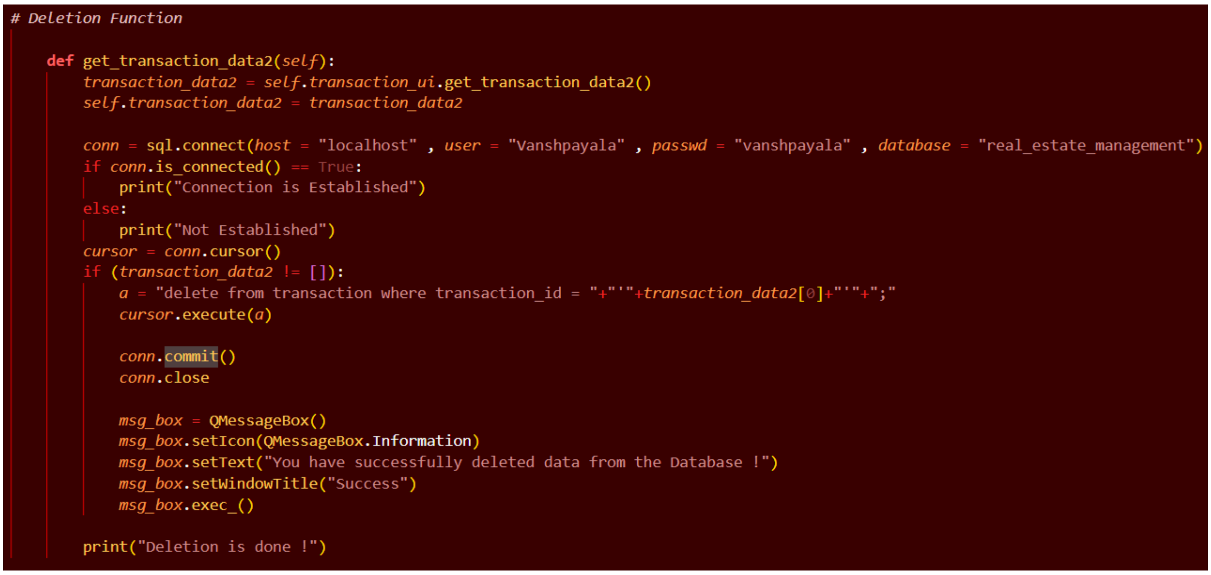
Now, below are the code snippets for the different operation which are performed by the UI –

Insertion Function –



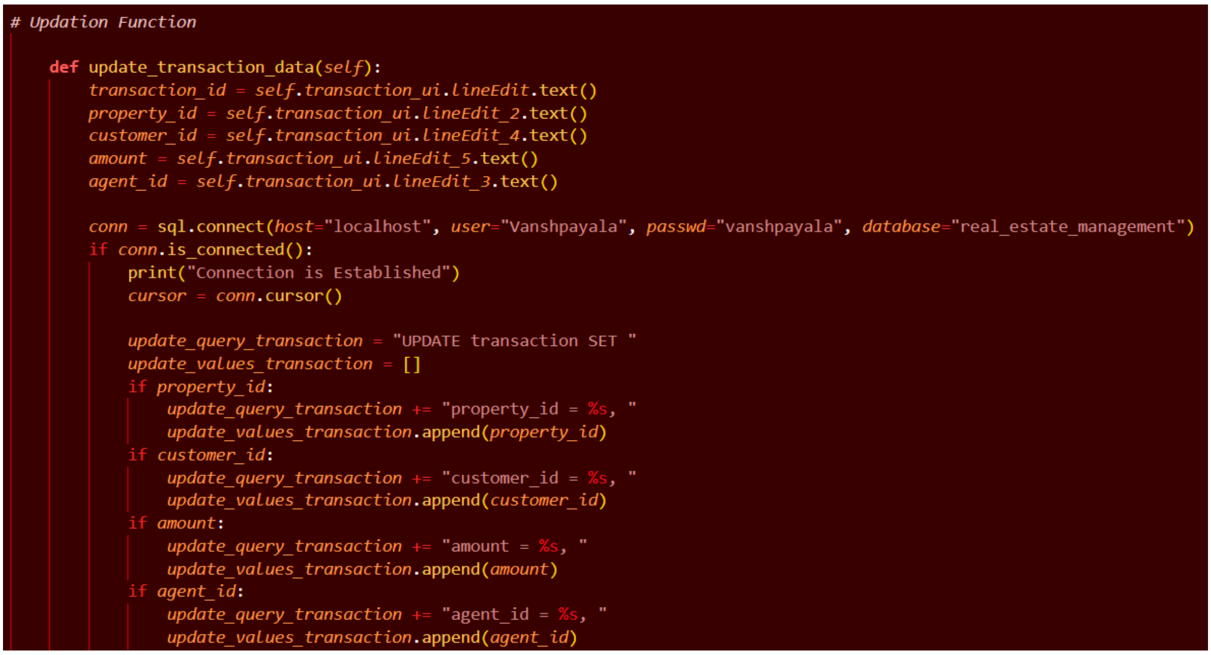
```python
# Insertion Function

    def get_transaction_data(self):
        transaction_data = self.transaction_ui.get_transaction_data()
        self.transaction_data = transaction_data

        conn = sql.connect(host = "localhost" , user = "Vanshpayala" , passwd = "vanshpayala" , database = "real_estate_management")
        if conn.is_connected() == True:
            print("Connection is Established")
        else:
            print("Not Established")
        cursor = conn.cursor()
        if (transaction_data != []):
            a = "insert into transaction values ("+"'"+transaction_data[0]+"'"+","+"'"+transaction_data[1]+"'"+","+"'"+transaction_data[2]+"'"+","+
            cursor.execute(a)

            conn.commit()
            conn.close

            msg_box = QMessageBox()
            msg_box.setIcon(QMessageBox.Information)
            msg_box.setText("You have successfully inserted data in the Database !")
            msg_box.setWindowTitle("Success")
            msg_box.exec_()

        print("Insertion is done !")
```

Deletion Function –

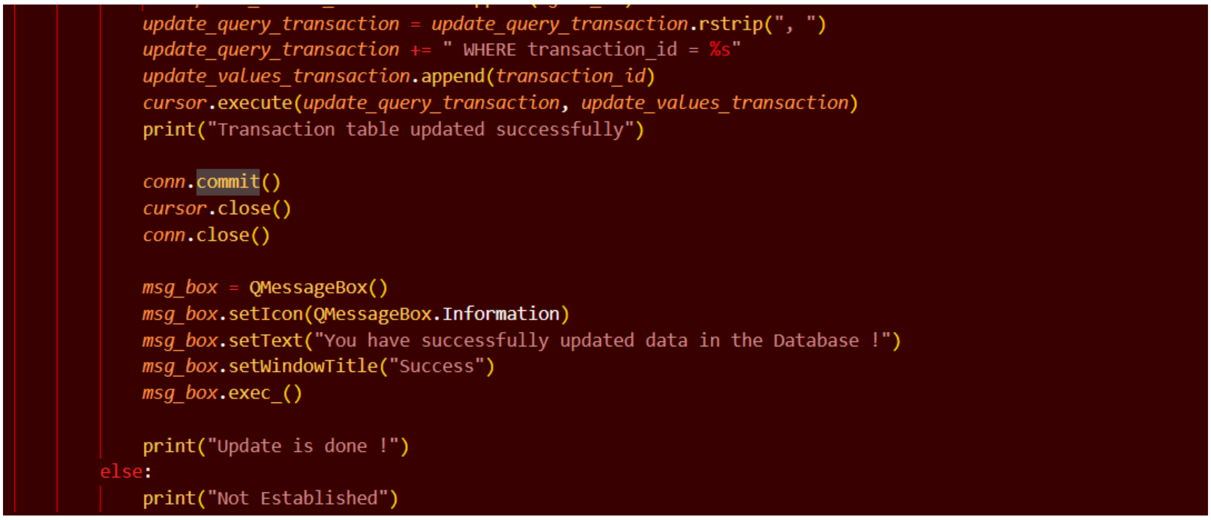


```
# Deletion Function

    def get_transaction_data2(self):
        transaction_data2 = self.transaction_ui.get_transaction_data2()
        self.transaction_data2 = transaction_data2

        conn = sql.connect(host = "localhost" , user = "Vanshpayala" , passwd = "vanshpayala" , database = "real_estate_management")
        if conn.is_connected() == True:
            print("Connection is Established")
        else:
            print("Not Established")
        cursor = conn.cursor()
        if (transaction_data2 != []):
            a = "delete from transaction where transaction_id = "+"'"+transaction_data2[0]+"'"+";"
            cursor.execute(a)

            conn.commit()
            conn.close

            msg_box = QMessageBox()
            msg_box.setIcon(QMessageBox.Information)
            msg_box.setText("You have successfully deleted data from the Database !")
            msg_box.setWindowTitle("Success")
            msg_box.exec_()

        print("Deletion is done !")
```

Update Function –

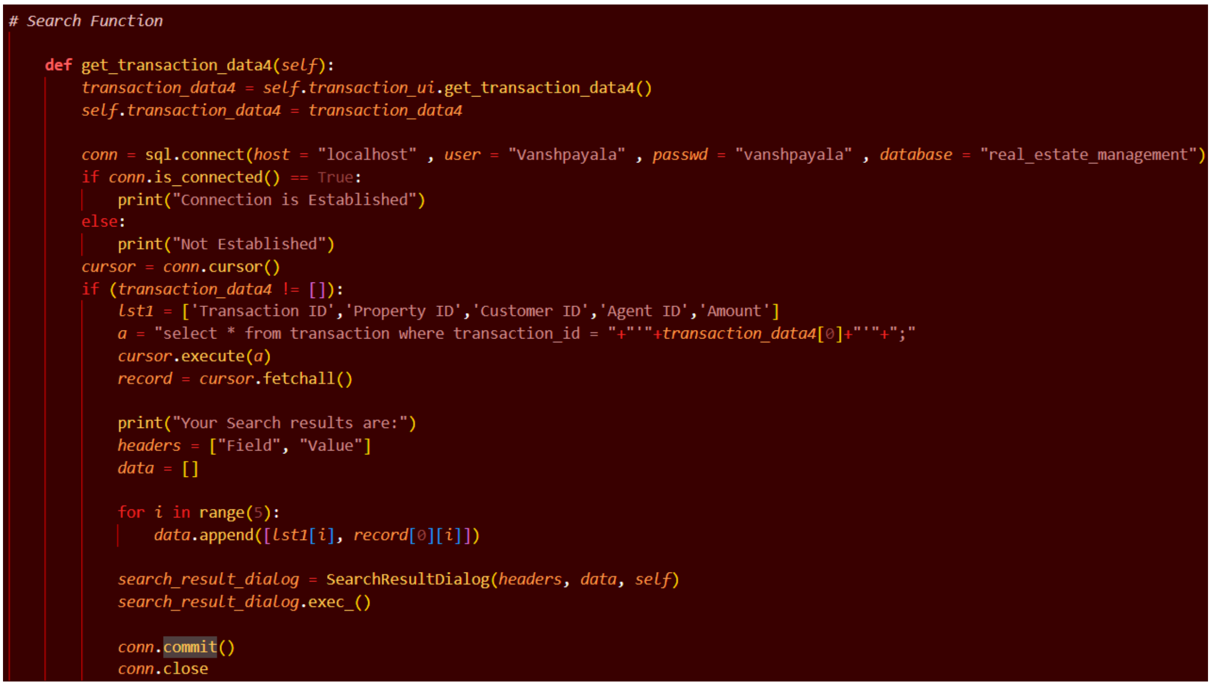


```
# Updation Function

    def update_transaction_data(self):
        transaction_id = self.transaction_ui.lineEdit.text()
        property_id = self.transaction_ui.lineEdit_2.text()
        customer_id = self.transaction_ui.lineEdit_4.text()
        amount = self.transaction_ui.lineEdit_5.text()
        agent_id = self.transaction_ui.lineEdit_3.text()

        conn = sql.connect(host="localhost", user="Vanshpayala", passwd="vanshpayala", database="real_estate_management")
        if conn.is_connected():
            print("Connection is Established")
            cursor = conn.cursor()

            update_query_transaction = "UPDATE transaction SET "
            update_values_transaction = []
            if property_id:
                update_query_transaction += "property_id = %s, "
                update_values_transaction.append(property_id)
            if customer_id:
                update_query_transaction += "customer_id = %s, "
                update_values_transaction.append(customer_id)
            if amount:
                update_query_transaction += "amount = %s, "
                update_values_transaction.append(amount)
            if agent_id:
                update_query_transaction += "agent_id = %s, "
                update_values_transaction.append(agent_id)
```



```
        update_query_transaction = update_query_transaction.rstrip(", ")
        update_query_transaction += " WHERE transaction_id = %s"
        update_values_transaction.append(transaction_id)
        cursor.execute(update_query_transaction, update_values_transaction)
        print("Transaction table updated successfully")

        conn.commit()
        cursor.close()
        conn.close()

        msg_box = QMessageBox()
        msg_box.setIcon(QMessageBox.Information)
        msg_box.setText("You have successfully updated data in the Database !")
        msg_box.setWindowTitle("Success")
        msg_box.exec_()

        print("Update is done !")
    else:
        print("Not Established")
```

Search Function –



```
# Search Function

    def get_transaction_data4(self):
        transaction_data4 = self.transaction_ui.get_transaction_data4()
        self.transaction_data4 = transaction_data4

        conn = sql.connect(host = "localhost" , user = "Vanshpayala" , passwd = "vanshpayala" , database = "real_estate_management")
        if conn.is_connected() == True:
            print("Connection is Established")
        else:
            print("Not Established")
        cursor = conn.cursor()
        if (transaction_data4 != []):
            lst1 = ['Transaction ID','Property ID','Customer ID','Agent ID','Amount']
            a = "select * from transaction where transaction_id = "+"'"+transaction_data4[0]+"'"+";"
            cursor.execute(a)
            record = cursor.fetchall()

            print("Your Search results are:")
            headers = ["Field", "Value"]
            data = []

            for i in range(5):
                data.append([lst1[i], record[0][i]])

            search_result_dialog = SearchResultDialog(headers, data, self)
            search_result_dialog.exec_()

            conn.commit()
            conn.close
```

Conclusion –

So, through this project, we can use the database (MySQL) using a User Interface (PyQT5) and can perform many operations such as inserting the data, updating the data, deleting the data and searching the desired data from the database.