

The background of the image is a warm, dimly lit living room. In the center, a framed picture of a landscape hangs on the wall, illuminated by a light fixture above it. On either side of the picture, a hanging lantern provides ambient light. In the foreground, a portion of a sofa with patterned cushions is visible.

FRSS

- KUNAL SINGH - 19CS30025
- ABHINANDAN DE - 19CS10069
- SURYAM ARNAV KALRA - 19CS30050

Your one stop shop for quality furniture !

The problem statement

And Why did we choose this topic ??

An overview

A platform where people can search, rent and loan furniture

The two actors in the system are:

- The admin :
 - Creates accounts and maintain inventory
- The Customer :
 - Loans/rents furniture

Storage of details for furniture:

- Type,
- Relevant images
- Rental price
- Cost price
- Interest rate
- Reviews

Useful for people who are looking to refurbish their house after relocation
and this was our motivation!

Tools used

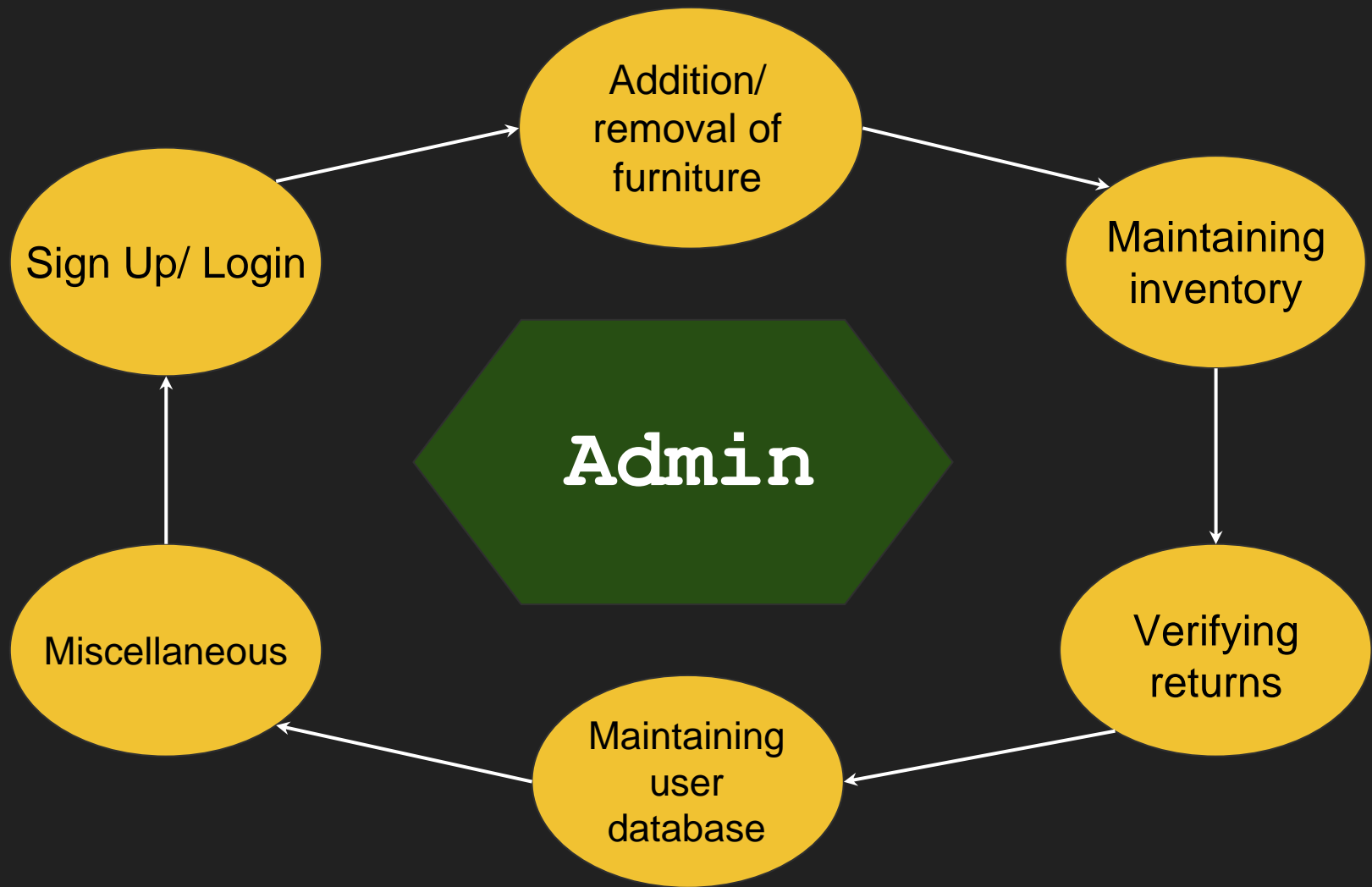
Tkinter

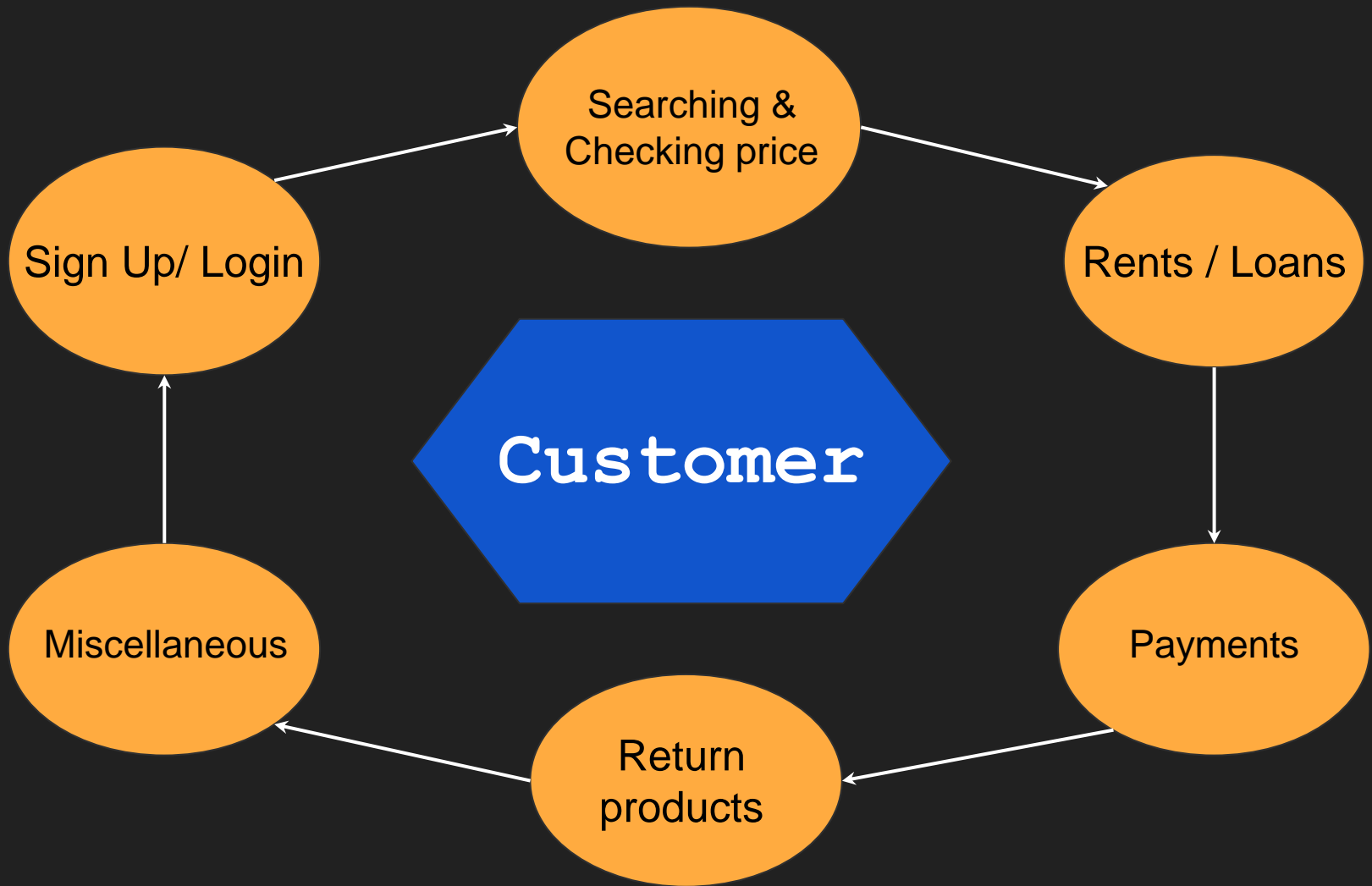


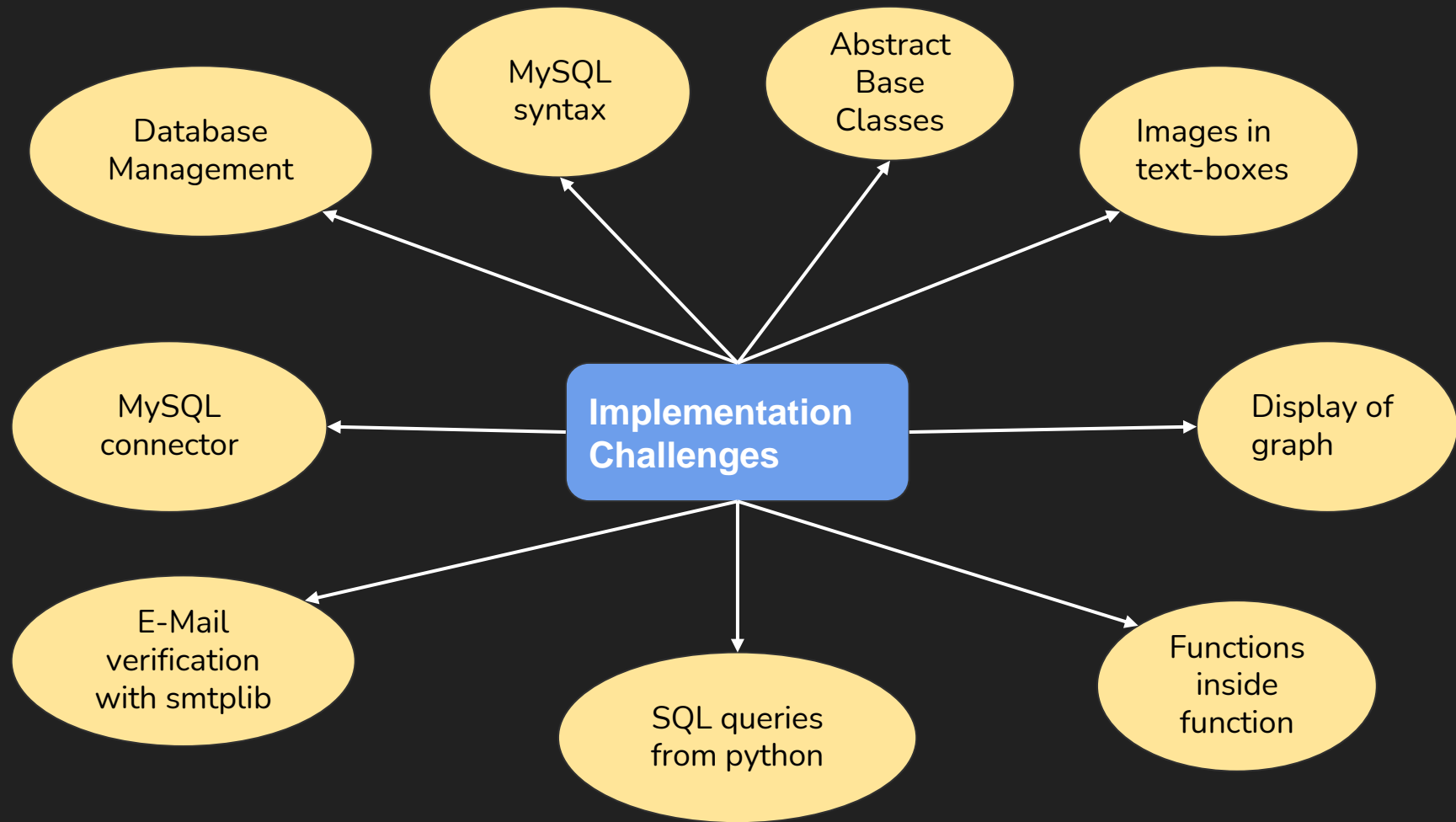
matplotlib

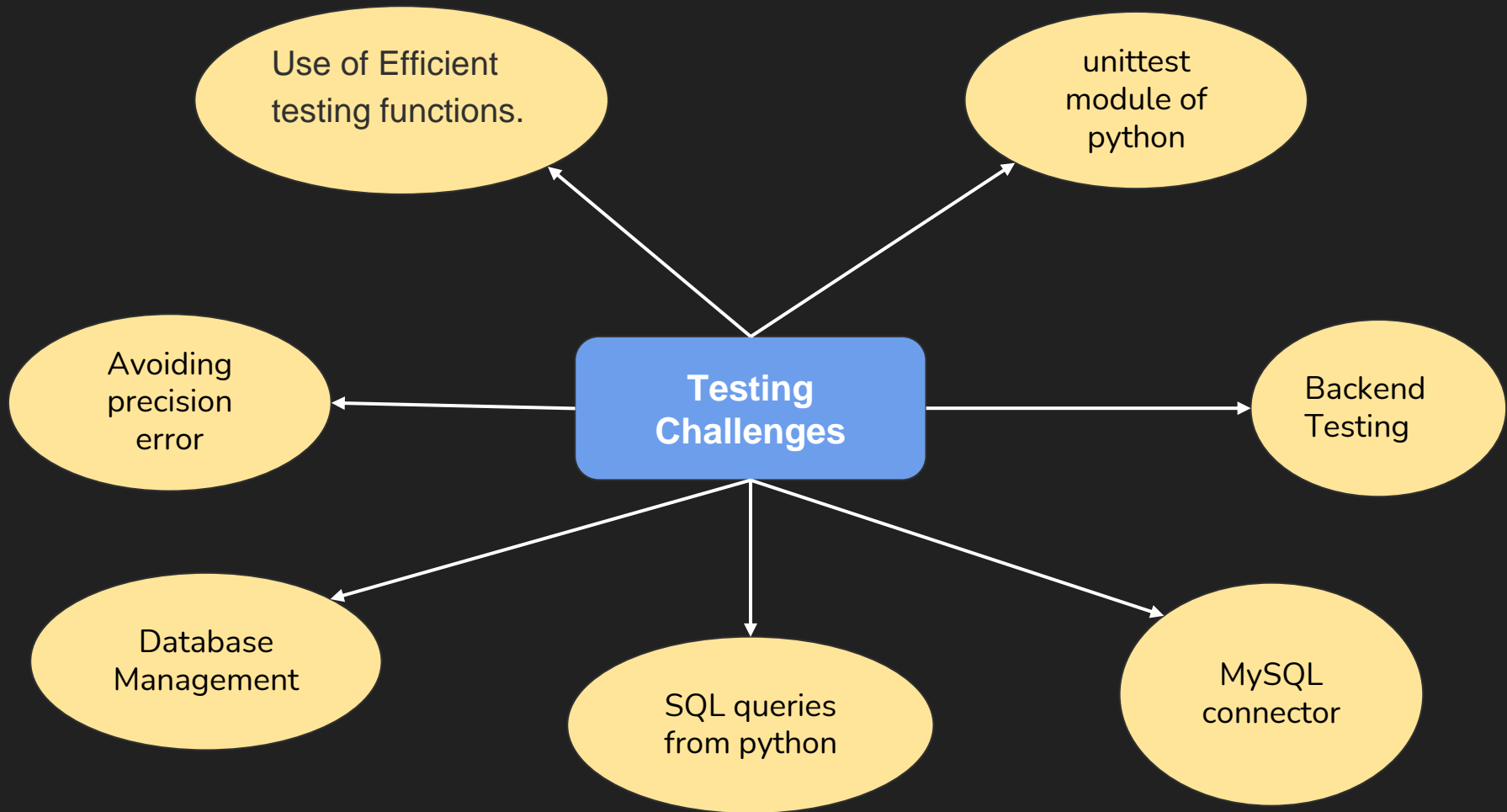


Features and Functionalities



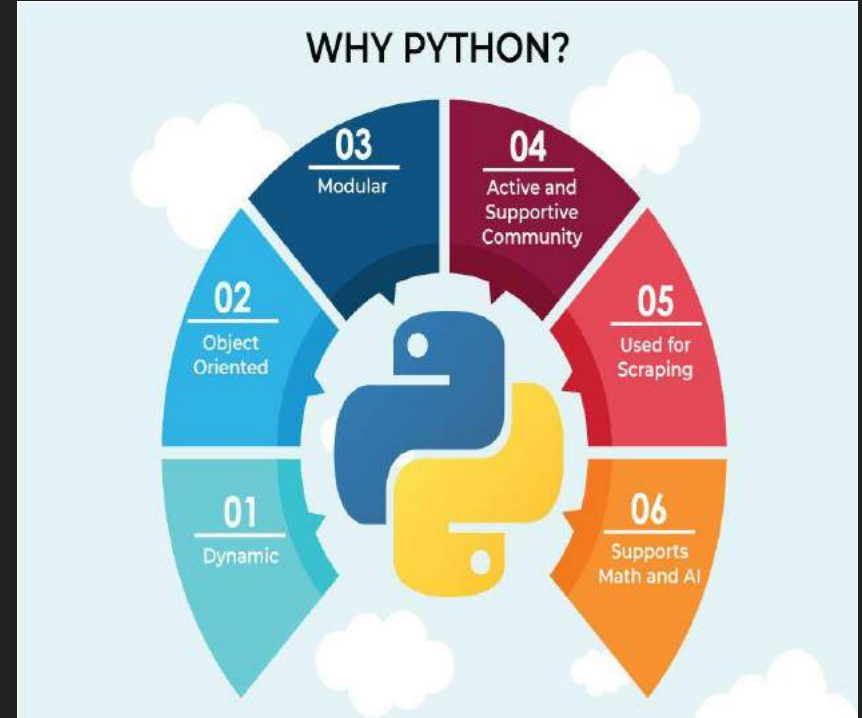






Motivation to Choose Python

- Simplified Syntax and similarity with natural language.
- Excellent libraries which reduce time and effort of development.
- Ultimate tool to play with graphs and data science.
- Great integration of MySQL as a database management tool.
- Easy to use GUI with Tkinter.
- Vast support of error detection with the VSCode editor.



e-mail verification

`ft. smtplib`



frental123@gmail.com

to ▼

OTP for login verification to our Furniture rental store system is 5547

↩ Reply

➡ Forward

Sign Up

Success! Email has been sent. Enter OTP sent to email id in 2 minutes

Name

Username

Enter password

Confirm password

Enter the address

Enter the phone number

Enter email id

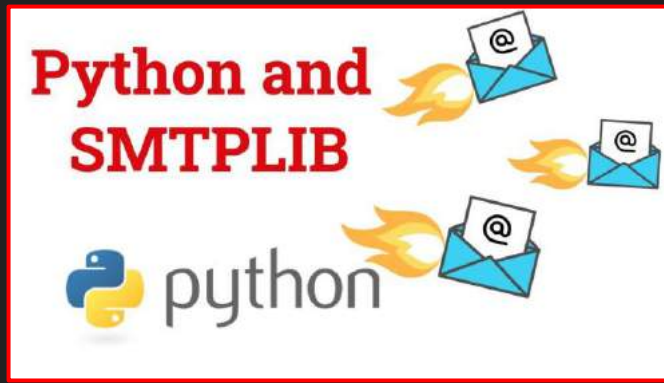
☐ Admin ☒ Customer

Verify details and send email

Enter the OTP here:

Verify OTP!

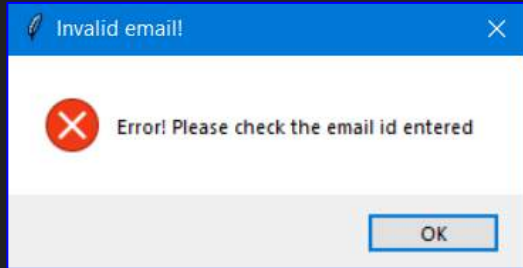
Sign Up



```
import smtplib
import random
import time
```

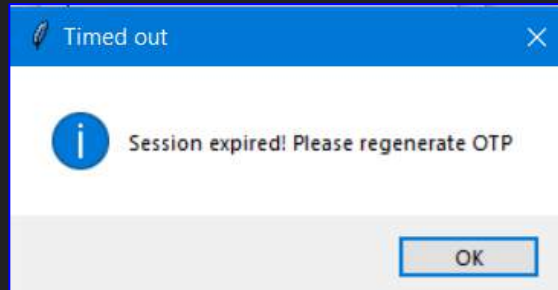
Use of the smtplib library to verify e-mail

Using random library to generate random OTPs.

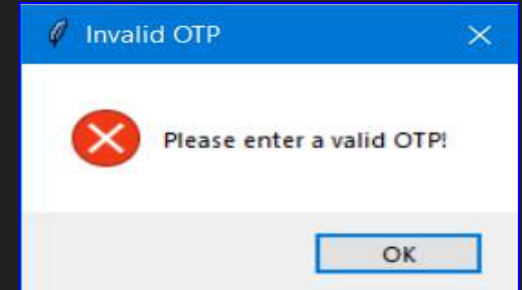


Email id validation

ERRORS!!



2 min limit



Final validation

Price Calculation while Buying Furniture

$$NewInterest = InterestRate \cdot 2^{-count}$$

count = Number of orders in the past

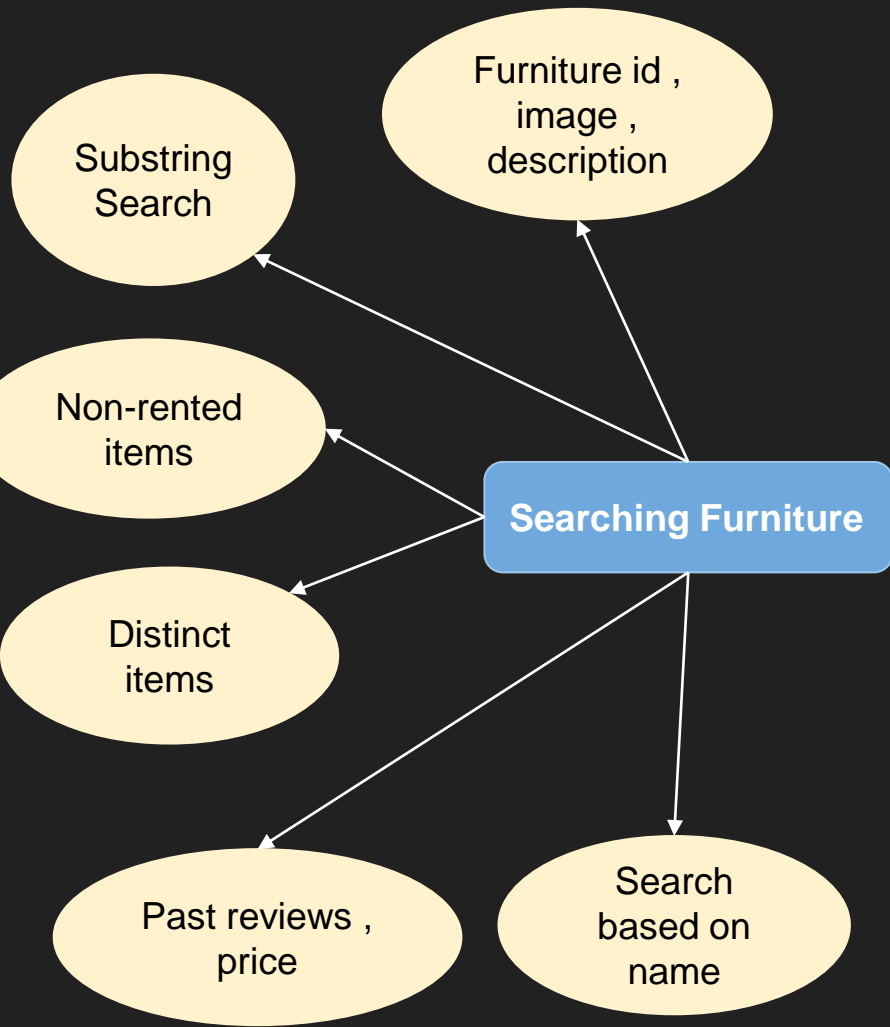
Interest Rate based on past order history

$$Price = Principal \cdot \left(1 + InterestRate \cdot \frac{TimePeriod}{100} \right)$$

Price Calculation while buying on loan

$$NewPrice = OldPrice \cdot \frac{(DaysRented + 9)}{10}$$


Price Calculation while buying on rent



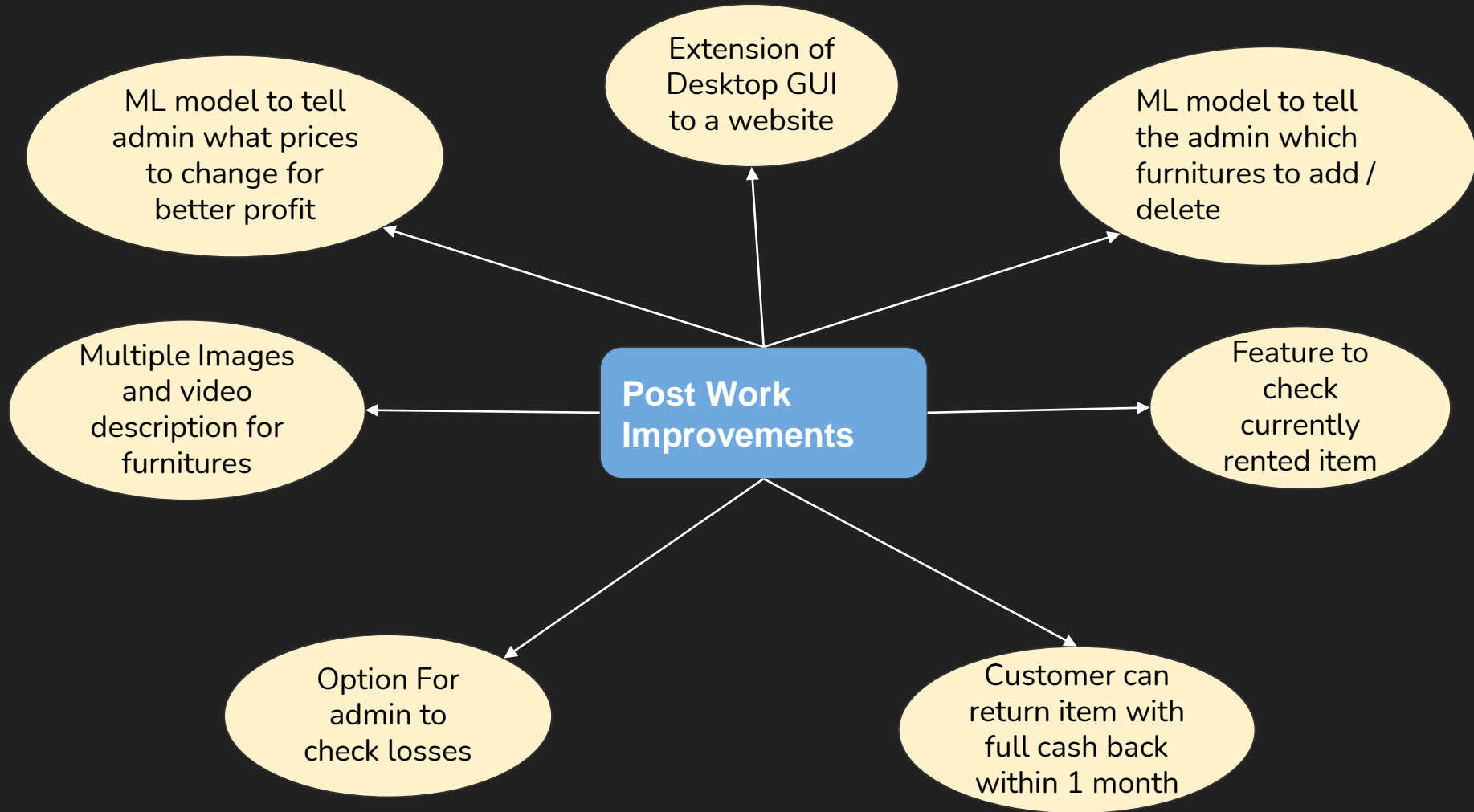
Search for furniture

Enter the name :

Search Now



Id : 5
Name : table
Company : tata
Price : 1111.00
Description : good table
Interest Rate : 12.00
Reviews :



**Post Work
Improvements**

ML model to tell
admin what prices
to change for
better profit

Extension of
Desktop GUI
to a website

ML model to tell
the admin which
furnitures to add /
delete

Feature to
check
currently
rented item

Customer can
return item with
full cash back
within 1 month

Option For
admin to
check losses

Multiple Images
and video
description for
furnitures

unittest module of python

01

UnitTest Module

- Verification of each functionality
- Validation by the QA team

02

Validation and Verification

- First level of software testing
- Unit testing each class
- Integration testing

03

OOP Concepts supported by framework

- Test Fixture
- Test Case
- Test Suite
- Test Runner

04

Possible Outcomes

- OK
- FAIL
- ERROR

How to Use unittest module of python

Process 1

Import unittest module and Define Testcase Subclass

Process 2

Define a function or a class (in our case) that is to be tested.

Process 3

Define a Test class as a subclass of unittest.TestCase

Process 4

In the class defined in Process 3 test each of the functions or methods of the class one by one.

Process 5

Then Execute the tests by the command \$python -m unittest {filename}.py

```
import unittest

class TestFurnitureMethods(unittest.TestCase):
    myfurniture = Furniture(1, "Center Table", "neelkumar")

    def test_id(self):
        self.assertEqual(self.myfurniture.getId(), 1)
```

.....

Ran 16 tests in 0.003s

OK

Backend / Database Testing

test.py is used to Test all the operations that involve any kind of reading and writing to the database.



Made a test.py file

test.py file contains all the function used for testing

Imported test.py in the main program and checked at each step whether the right query is made and the desired changes are being done or not in the main database.

```
def testfurnituredelete(fur_id):  
    exe = "SELECT * FROM furnitures WHERE id = %s"  
    va = (fur_id,)   
    my_cursor1.execute(exe, va)  
    res = my_cursor1.fetchall()  
    mydb1.commit()  
    if len(res) > 0:  
        print("DELETE DAMAGED FURNITURE : FAILED")  
    else:  
        print("DELETE DAMAGED FURNITURE : PASSED")
```

A SHORT DEMO

