NEST SEEKER

A house renting app

DESCRIPTION:

a simple house renting application using object-oriented programming principles and the pandas library for data management. The house class represents a house with attributes such as address, price, and size. The app class is responsible for managing houses, and provides methods to add and remove houses, search for houses by price or size, and display all houses. The allhouses class is used to create a pandas DataFrame from a list of houses, and provides methods to filter and sort the DataFrame by price or size.

Description about different classes and modules used

1. House Classes:
    - The House class represents a house with attributes such as address, rent, number of bedrooms, and number of bathrooms..
2. User class:
    - The user class represents a user with a name and a list of purchased houses
3. App Class:
    - The app class is the main class responsible for managing houses, users, and purchases.
    - It initializes DataFrames to store information about all houses, purchased houses, available houses, and users.
    - Methods are provided to add houses and users, make purchases, and view purchased and available houses.
    - The rent_distribution method visualizes the rent distribution of available houses using a histogram.
4. User Interaction:
    - The program provides a simple command-line interface for users to interact with the application.
    - Users can add houses 1, create user accounts 2, make purchases 3, view rent distribution 4, view purchased houses 5_1, view available houses 5_2, or stop the program 6.
5. Main Loop:
    - The program runs in an infinite loop where users input commands according to the instructions provided.
    - Depending on the input command, the corresponding method of the `RentingApp` class is called to perform the desired action.
6. Description:
    - The program allows users to add houses, create user accounts, make purchases, and view rent distribution.
    - Users can also view lists of purchased and available houses.
    - It provides a simple way to interact with the house renting application via the command line.

Overall, this code demonstrates the implementation of a basic house renting application using object-oriented programming concepts and pandas for data management. It offers functionality for managing houses, users, and purchases, as well as visualizing rent distribution.

CODE:

```
import pandas as pd
import matplotlib.pyplot as plt
class House:
    def __init__(self, address, rent, bedrooms, bathrooms):
        self.address = address
        self.rent = rent
        self.bedrooms = bedrooms
        self.bathrooms= bathrooms

class User:
    def __init__(self, name):
        self.name = name
        self.purchases = []

    def purchase(self, house):
        self.purchases.append(house)

class App:
    def __init__(self):
        self.allhouses = pd.DataFrame(columns=['ID', 'Address', 'Rent', 'Bedrooms', 'Bathrooms'])
        self.purchasedhouses = pd.DataFrame(columns=['ID', 'Address', 'Rent', 'Bedrooms',
'Bathrooms'])
        self.availablehouses= pd.DataFrame(columns=['ID', 'Address', 'Rent', 'Bedrooms',
'Bathrooms'])
        self.users= pd.DataFrame(columns=['ID', 'Name'])

    def add_house(self, address, rent, bedrooms, bathrooms):
        newhouse_id = len(self.allhouses) + 1
        newhouse = pd.DataFrame([[newhouse_id, address, rent, bedrooms, bathrooms]],
columns=['ID', 'Address', 'Rent', 'Bedrooms', 'Bathrooms'])
        self.allhouses = self.allhouses._append(newhouse, ignore_index=True)
        self.availablehouses = self.availablehouses._append(newhouse, ignore_index=True)

    def add_user(self, name):
        nuser_id = len(self.users) + 1
        nuser = pd.DataFrame([[nuser_id, name]], columns=['ID', 'Name'])
        self.users = self.users._append(nuser, ignore_index=True)

    def make_purchase(self, user_id, house_id):
        purchasedhouse = self.availablehouses[self.availablehouses['ID'] == house_id]
        self.purchasedhouses =self.purchasedhouses._append(purchasedhouse,ignore_index=True)
        self.availablehouses= self.availablehouses[self.availablehouses['ID'] != house_id]
        #newpurchase = pd.DataFrame([[user_id, house_id]], columns=['User_ID', 'House_ID'])
        #self.purchasedhouses= self.purchasedhouses._append(newpurchase, ignore_index=True)
```

```
    def purchased_houses(self):
        print("Purchased Houses:")
        print(self.purchasedhouses)

    def available_houses(self):
        print("Available Houses:")
        print(self.availablehouses)

    def rent_distribution(self):
        rents = self.allhouses['Rent']
        plt.hist(rents, bins=10, color='skyblue', edgecolor='black')
        plt.title('Rent Distribution')
        plt.xlabel('Rent (rupees)')
        plt.ylabel('Frequency')
        plt.show()

obj= App()
obj.add_house("gandipet cross road",25000,2,2)
obj.add_house("hitech city, road 2",30000,2,2)
obj.add_house("char minar road",8000,2,2)
obj.add_house("chandrayan gutta",10000,2,2)
print(" for selling house-1\n for creating account-2 \n for purchasing    house-3\n for vewing
purchases in graph-4\n for getting lists of \n   purchased houses - 5_1\n   available houses -5-2\n
stop-6")
while True:
    n=input('enter the no. according to the instructions given')
    if n=='1':
        add=input("address = ")
        rent=int(input('rent = '))
        bt=input('no. of bathrooms')
        br=input('no. of bedrooms')
        obj.add_house(add,rent,bt,br)
    if n=='2':
        obj.add_user(input('name'))
    if n=='3':
        obj.make_purchase(int(input("user id")),int(input("houseid")))
    if n=='4':
        obj.rent_distribution()
    if n[0]=='5':
        if n[-1]=='1':
            obj.purchased_houses()
        if n[-1]=='2':
            obj.available_houses()
    if n=='6':
        break
```

OUTPUT:

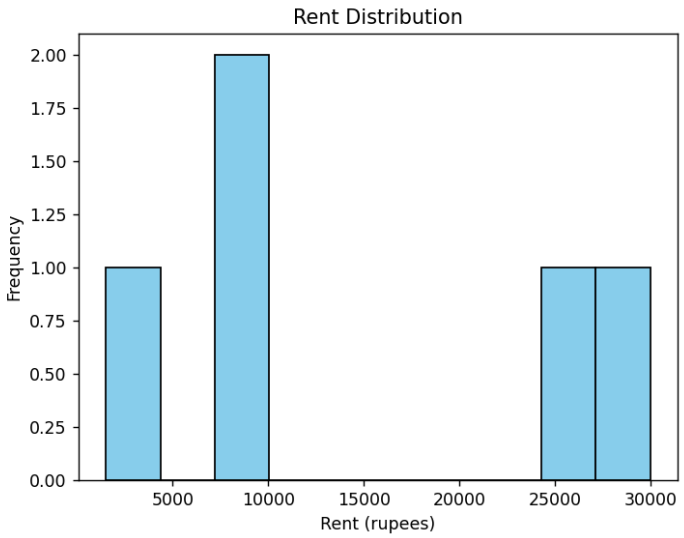
for selling house-1
for creating account
for purchasing house-3
for vewing purchases in graph-4
for getting lists of
purchased houses - 5_1
available houses -5-2
stop-6
enter the no. according to the instructions given2
namecbit
enter the no. according to the instructions given3
user id1
houseid1
enter the no. according to the instructions given5_1
Purchased Houses:

	ID	Address	Rent	Bedrooms	Bathrooms
0	1	gandipet cross road	25000	2	2

enter the no. according to the instructions given5_2
Available Houses:

	ID	Address	Rent	Bedrooms	Bathrooms
1	2	hitech city, road 2	30000	2	2
2	3	char minar road	8000	2	2
3	4	chandrayan gutta	10000	2	2

enter the no. according to the instructions given1
address = balapur
rent = 1500
no. of bathrooms2
no. of bedrooms2
enter the no. according to the instructions given4



DONE BY:
1. Rithika - 160123737072
2. Sahithi - 160123737081
3. Nirupama - 160123737083
4. Sudheeksha - 160123737084
5. Pravalika - 160123737088