Coding Challenges: PetPals, The Pet Adoption Platform

Pets.py

Pet class:

```
from abc import ABC, abstractmethod
import mysql.connector as con
import Exception_pet_pals
from datetime import datetime

class Pet:
    def _init_(self, name, age, breed):
        self.name = name
        self.age = age
        self.breed = breed

    def str(self):
        return f"{self.name} - Age: {self.age}, Breed: {self.breed}"

    @property
    def name(self):
        return self.name

    @name.setter
    def name(self, name):
        self.name = name

    @property
    def age(self):
        if(self.age<0)
            raise Exception_pet_pals.InvalidPetAgeException("invalid age")
        return self.age

    @age.setter
    def age(self, age):
        self.age = age

    @property
    def breed(self):
        return self.breed

    @breed.setter
    def breed(self, breed):
        self.breed = breed
```

Dog class:

```
class Dog(Pet):
    def init(self, name, age, breed, dog_breed):
        super().__init__(name, age, breed)
        self.dog_breed = dog_breed

    def str(self):
        return f"{super().str()}, Dog Breed: {self.dog_breed}"

    @property
    def dog_breed(self):
        return self.dog_breed

    @dog_breed.setter
    def set_dog_breed(self, dog_breed):
        self.dog_breed = dog_breed
```

Cat class:

```
class Cat(Pet):
    def init(self, name, age, breed, cat_color):
        super().__init__(name, age, breed)
        self.cat_color = cat_color

    def str(self):
        return f"{super().str()}, Cat Color: {self.cat_color}"

    @property
    def cat_color(self):
        return self.cat_color

    @cat_color.setter
    def set_cat_color(self, cat_color):
        self.cat_color = cat_color
```

PetShelter class:

```
class PetShelter:
    def init(self):
        self.available_pets = []

    def add_pet(self, pet):
        self.available_pets.append(pet)

    def remove_pet(self, pet):
        if pet in self.available_pets:
            self.available_pets.remove(pet)

    def list_available_pets(self):
```

```
        for pet in self.available_pets:
            print(pet)
```

Donation class:

```
class Donation(ABC):
    def __init__(self, donor_name, amount):
        self.DonorName = donor_name
        self.Amount = amount

    @abstractmethod
    def RecordDonation(self):
        pass

# Derived class: CashDonation
```

CashDonation class:

```
class CashDonation(Donation):
    def __init__(self, donor_name, amount, donation_date):
        super().__init__(donor_name, amount)
        self.DonationDate = donation_date

    def RecordDonation(self):
        print(f"Cash donation recorded on {self.DonationDate}: {self.Amount}
from {self.DonorName}")

# Derived class: ItemDonation
```

ItemDonation class:

```
class ItemDonation(Donation):
    def __init__(self, donor_name, amount, item_type):
        super().__init__(donor_name, amount)
        self.ItemType = item_type

    def RecordDonation(self):
        print(f"Item donation recorded: {self.Amount} from {self.DonorName},
Type: {self.ItemType}")

# Interface/Abstract class: IAdoptable
```

IAdoptable class:

```
class IAdoptable(ABC):
    @abstractmethod
    def Adopt(self):
        pass

# AdoptionEvent class
```

AdoptionEvent class:

```
class AdoptionEvent:
    def _init_(self):
        self.Participants = []

    def HostEvent(self):
        print("Adoption event hosted!")

    def RegisterParticipant(self, participant):
        self.Participants.append(participant)
        print(f"{participant} registered for the adoption event.")

# Example usage
cash_donation = CashDonation("John Doe", 500.00, datetime.now())
item_donation = ItemDonation("Jane Smith", 10.00, "Toys")

cash_donation.RecordDonation()
item_donation.RecordDonation()
```

AnimalShelter class:

```
class AnimalShelter(IAdoptable):
    def Adopt(self):
        print("Animal adopted from the shelter.")
```

Adopter class:

```
class Adopter(IAdoptable):
    def Adopt(self):
        print("Adopter has adopted a pet.")

shelter = AnimalShelter()
adopter = Adopter()

event = AdoptionEvent()
event.RegisterParticipant(shelter)
event.RegisterParticipant(adopter)

event.HostEvent()

shelter.Adopt()
adopter.Adopt()
```

Exception Handling:

```
class NullReferenceException(Exception):
    def __init__(self, message="given infromation is missing"):
        self.message = message
        super().__init__(self.message)

class InsufficientFundException(Exception):
    def __init__(self, message="Donation amount tis less than 10$"):
        self.message = message
        super().__init__(self.message)

class FileHandlingException(Exception):
    def __init__(self, message=" file  not found"):
        self.message = message
        super().__init__(self.message)
```

DataBase connectivity:

```
import mysql.connector
from mysql.connector import Error
from datetime import datetime

# Database connection
def create_connection():
    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="Aparna@1234",
            port='3306',
            database="petpals"
        )
        return connection
    except Error as e:
        print(f"Error connecting to the database: {e}")
        return None
```

Implemented Functions:

```
def display_pet_listings():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM pets")
            pets = cursor.fetchall()

            print("Available Pets:")
            for pet in pets:
                print(f"{pet[1]} - Age: {pet[2]}, Breed: {pet[3]}")


        except Error as e:
            print(f"Error retrieving pet listings: {e}")
        finally:
            connection.close()

####
def display_adoptionEvents():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM adoptionevents;")
            event = cursor.fetchall()

            print("Upcoming adoption events:")
            for e in event:
                print(f" AdoptionEvent: {e[1]}            Location: {e[3]}")

        except Error as e:
            print(f"Error retrieving pet listings: {e}")
        finally:
            connection.close()

###

###
def display_adoptionEvents():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM shelters;")
            event = cursor.fetchall()

            print("All nearby shelters :")
            for e in event:
                print(f" Shelter: {e[1]}            Location: {e[2]}")
```

```
        except Error as e:
            print(f"Error retrieving pet listings: {e}")
        finally:
            connection.close()

###
donation_counter = 10
def generate_donation_number():
    global donation_counter
    donation_counter += 1
    return donation_counter

def record_cash_donation():
    connection = create_connection()
    if connection:
        try:
            donation_number=generate_donation_number()
            donor_name = input("Enter donor name: ")
            amount = float(input("Enter donation amount: "))
            Donation_type  =  input("Enter donation type: ")
            Donation_item  = input("Enter donation item: ")
            donation_date = datetime.now().strftime("%Y-%m-%d")

            cursor = connection.cursor()
            cursor.execute("INSERT INTO donations (donationID,
DonorName,DonationType,DonationAmount,DonationItem,DonationDate) VALUES (%s,
%s, %s, %s,%s,%s)",
                           (donation_number,donor_name,Donation_type,
amount,Donation_item , donation_date))
            connection.commit()

            print("Donation recorded successfully!")

        except (Error, ValueError) as e:
            print(f"Error recording donation: {e}")
        finally:
            connection.close()
```

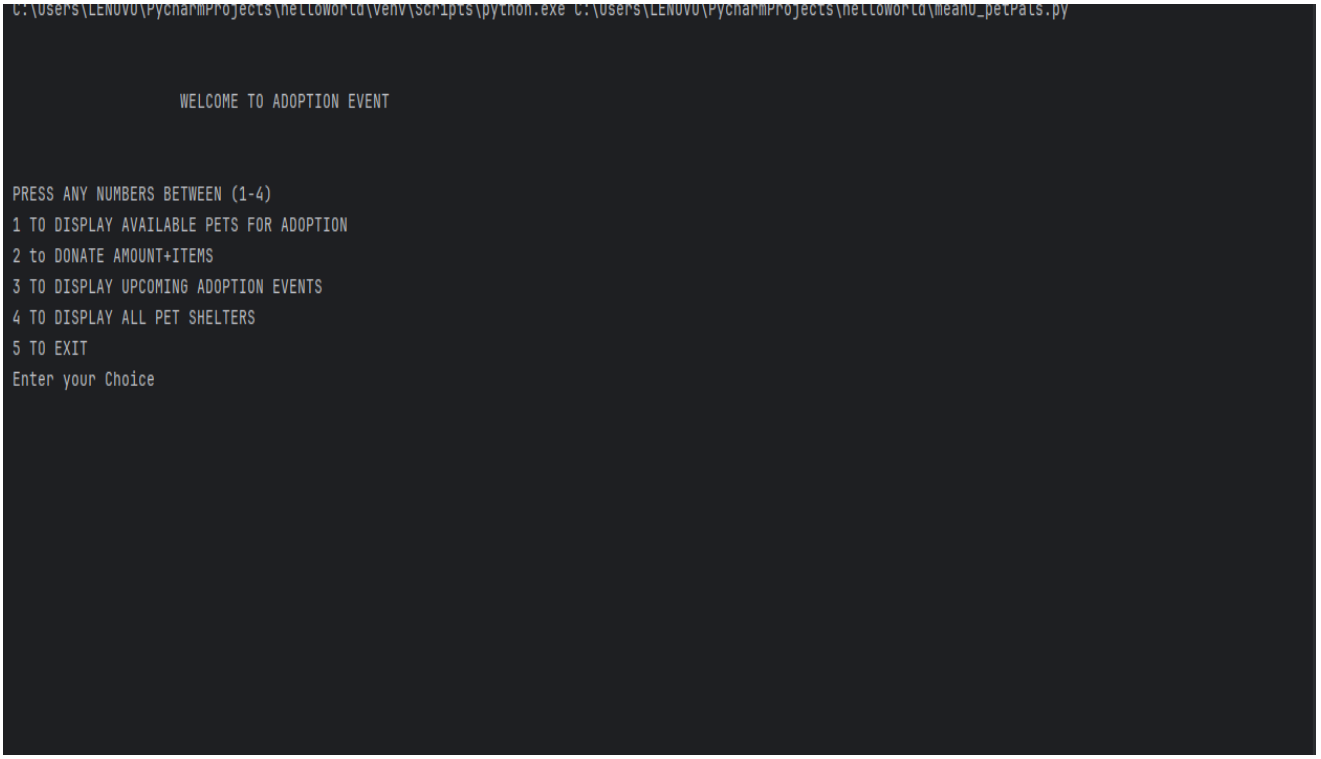
Menu:

```
def menu():
    print("\n\n\t\t\t\t\tWELCOME TO ADOPTION EVENT ")
    print("\n\nPRESS ANY NUMBERS BETWEEN (1-4)")
    print("1 TO DISPLAY AVAILABLE PETS FOR ADOPTION")
    print("2 to DONATE AMOUNT+ITEMS ")
    print("3 TO DISPLAY UPCOMING ADOPTION EVENTS")
    print("4 TO DISPLAY ALL PET SHELTERS ")
    print("5 TO EXIT ")
```

```
    ch = int(input("Enter your Choice "))
    if ch == 1:
        petsDatabaseConnector.display_pet_listings()
    elif ch == 2:
        petsDatabaseConnector.record_cash_donation()
    elif ch == 3:
        petsDatabaseConnector.display_adoptionEvents()
    elif ch == 4:
        petsDatabaseConnector.display_adoptionEvents()


# Calling menu function
menu()
```

Output:



```
                    WELCOME TO ADOPTION EVENT


PRESS ANY NUMBERS BETWEEN (1-4)
1 TO DISPLAY AVAILABLE PETS FOR ADOPTION
2 to DONATE AMOUNT+ITEMS
3 TO DISPLAY UPCOMING ADOPTION EVENTS
4 TO DISPLAY ALL PET SHELTERS
5 TO EXIT
Enter your Choice 1
Available Pets:
Buddy - Age: 3, Breed: Labrador
Whiskers - Age: 5, Breed: Siamese
Rocky - Age: 2, Breed: Bulldog
Mittens - Age: 4, Breed: Persian
Max - Age: 1, Breed: German Shepherd
Oreo - Age: 2, Breed: Maine Coon
Charlie - Age: 4, Breed: Poodle
Luna - Age: 3, Breed: Ragdoll
Bailey - Age: 2, Breed: Golden Retriever
Simba - Age: 5, Breed: Siberian

Process finished with exit code 0
```

```
C:\Users\LENOVO\PycharmProjects\helloWorld\venv\Scripts\python.exe C:\Users\LENOVO\PycharmProjects\helloWorld\meanu_petPals.py


                    WELCOME TO ADOPTION EVENT


PRESS ANY NUMBERS BETWEEN (1-4)
1 TO DISPLAY AVAILABLE PETS FOR ADOPTION
2 to DONATE AMOUNT+ITEMS
3 TO DISPLAY UPCOMING ADOPTION EVENTS
4 TO DISPLAY ALL PET SHELTERS
5 TO EXIT
Enter your Choice 2
Enter donor name: Aparna Bharti
Enter donation amount: 345
Enter donation type: cash
Enter donation item: food
Error recording donation: 1062 (23000): Duplicate entry '11' for key 'donations.PRIMARY'

Process finished with exit code 0
```

```
                    WELCOME TO ADOPTION EVENT


PRESS ANY NUMBERS BETWEEN (1-4)
1 TO DISPLAY AVAILABLE PETS FOR ADOPTION
2 to DONATE AMOUNT+ITEMS
3 TO DISPLAY UPCOMING ADOPTION EVENTS
4 TO DISPLAY ALL PET SHELTERS
5 TO EXIT
Enter your Choice 3
All nearby shelters :
 Shelter: Happy Paws Shelter            Location: 123 Main Street, City A
 Shelter: Second Chance Shelter            Location: 456 Elm Street, City B
 Shelter: Paws and Claws Rescue            Location: 789 Oak Avenue, City C
 Shelter: Furry Friends Haven            Location: 321 Maple Lane, City D
 Shelter: Rescue Me Shelter            Location: 555 Pine Street, City E
 Shelter: Safe Haven Animal Rescue            Location: 888 Cedar Road, City F
 Shelter: Homeless Pets Sanctuary            Location: 777 Birch Boulevard, City G
 Shelter: Kindness Animal Shelter            Location: 1010 Walnut Drive, City H
 Shelter: Forever Friends Center            Location: 222 Chestnut Court, City I
 Shelter: Hearts of Gold Shelter            Location: 444 Spruce Place, City J

Process finished with exit code 0
```

```
                    WELCOME TO ADOPTION EVENT


PRESS ANY NUMBERS BETWEEN (1-4)
1 TO DISPLAY AVAILABLE PETS FOR ADOPTION
2 to DONATE AMOUNT+ITEMS
3 TO DISPLAY UPCOMING ADOPTION EVENTS
4 TO DISPLAY ALL PET SHELTERS
5 TO EXIT
Enter your Choice 4
All nearby shelters :
 Shelter: Happy Paws Shelter            Location: 123 Main Street, City A
 Shelter: Second Chance Shelter            Location: 456 Elm Street, City B
 Shelter: Paws and Claws Rescue            Location: 789 Oak Avenue, City C
 Shelter: Furry Friends Haven            Location: 321 Maple Lane, City D
 Shelter: Rescue Me Shelter            Location: 555 Pine Street, City E
 Shelter: Safe Haven Animal Rescue            Location: 888 Cedar Road, City F
 Shelter: Homeless Pets Sanctuary            Location: 777 Birch Boulevard, City G
 Shelter: Kindness Animal Shelter            Location: 1010 Walnut Drive, City H
 Shelter: Forever Friends Center            Location: 222 Chestnut Court, City I
 Shelter: Hearts of Gold Shelter            Location: 444 Spruce Place, City J

Process finished with exit code 0
```

```
C:\Users\LENOVO\PycharmProjects\helloWorld\venv\Scripts\python.exe C:\Users\LENOVO\PycharmProjects\helloWorld\meanu_petPals.py


                    WELCOME TO ADOPTION EVENT


PRESS ANY NUMBERS BETWEEN (1-4)
1 TO DISPLAY AVAILABLE PETS FOR ADOPTION
2 to DONATE AMOUNT+ITEMS
3 TO DISPLAY UPCOMING ADOPTION EVENTS
4 TO DISPLAY ALL PET SHELTERS
5 TO EXIT
Enter your Choice 5

Process finished with exit code 0
```