CODING ASSESMENT – 1

PETPALS

1.Create and implement the mentioned class and the structure in your application.

Create and implement the following tasks in your application.

Create and implement the following exceptions in your application.

```
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):
        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

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

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

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)
```

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

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" Name :{pet[1]} , Age: {pet[2]}, Breed: {pet[3]}")

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

OUTPUT

```
Available Pets:
 Name :Jack , Age: 3, Breed: Afollie
 Name :Jill , Age: 2, Breed: Maine Coon
 Name :Kitto , Age: 1, Breed: Doberman
 Name :Siva , Age: 3, Breed: Persian cat
 Name :Tiger , Age: 1, Breed: Siberian Huskey
 Name :Shilly , Age: 3, Breed: Siberian cat
 Name :Tommy , Age: 1, Breed: Golden retriener
 Name :Julie , Age: 2, Breed: Himalayan Cat
 Name :Viba , Age: 4, Breed: German sherpard
 Name :Rosy , Age: 1, Breed: Bombay Cat
 Name :Reeva , Age: 3, Breed: Bombay cat
```

```
from abc import ABC, abstractmethod
from datetime import datetime
from Pet import Pet
class Donation(ABC):
    def __init__(self,Donor_name,Amount):
      self.Donor_name  = Donor_name
      self.Amount = Amount

    def RecordDonation(self):
        pass

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}")

class FileReader:
    def ReadFromFile(self, filename):
        try:
```

```
            with open(filename, 'r') as file:
                data = file.read()
                print(f"Data read from file: {data}")
        except FileNotFoundError:
            print(f"Error: File '{filename}' not found.")
        except Exception as e:
            print(f"Error: {e}")

shelter = Pet()
try:
    pet_name = input("Enter the name of the pet: ")
    pet_age = int(input("Enter the age of the pet: "))
    if pet_age <= 0:
        raise ValueError("Invalid age. Please enter a positive integer.")
    pet = Pet(pet_name, pet_age)
    shelter.AddPet(pet)
except ValueError as e:
    print(f"Error: {e}")

shelter.ListAvailablePets()

try:
    donation_amount = float(input("Enter the donation amount: $"))
    cash_donation = CashDonation(donation_amount)
    cash_donation.ProcessDonation()
except ValueError as e:
    print(f"Error: {e}")

file_reader = FileReader()
file_reader.ReadFromFile("nonexistent_file.txt")
```

```
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_date = datetime.now().strftime("%Y-%m-%d")
```

```
            cursor = connection.cursor()
            cursor.execute("INSERT INTO donations (donation_id, donor_name,
donation_amount, donation_date) VALUES (%s, %s, %s, %s)",
                           (donation_number,donor_name, amount,
donation_date))
            connection.commit()

            print("Donation recorded successfully!")

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

OUTPUT

```
PROBLEMS 1  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

/Python/Python312/python.exe "c:/python/Coding Assesment1/databasefile.py"
Enter donor name: Kelvin
Enter donation amount: 1000
Error recording donation: 1054 (42S22): Unknown column 'donar_name' in 'field list'
PS C:\python\Coding Assesment1> & C:/Users/welcome/AppData/Local/Programs/Python/Python312/python.exe "c:/python/Coding Assesment1/databasefile.py"
Enter donor name: Kelvin
Enter donation amount: 1000
Donation recorded successfully!
PS C:\python\Coding Assesment1>
```

```
mysql> select * from donations;
+-------------+------------+---------------+-----------------+----------------------+---------------------+
| Donation_id | Donor_name | Donation_type | Donation_amount | Donation_item        | Donation_date       |
+-------------+------------+---------------+-----------------+----------------------+---------------------+
|           1 | Martin     | cash          |          500.00 | null                 | 2023-05-10 09:30:00 |
|           2 | Charlie    | item          |            NULL | pet toys             | 2023-05-12 14:45:00 |
|           3 | David      | item          |            NULL | pet food             | 2023-05-15 18:20:00 |
|           4 | Knia       | cash          |         5000.00 | NULL                 | 2023-05-18 11:10:00 |
|           5 | John       | item          |            NULL | pet foods            | 2023-05-20 16:55:00 |
|           6 | Samthan    | item          |            NULL | Pet Bedding          | 2023-05-22 20:30:00 |
|           7 | Jane       | cash          |          700.00 | NULL                 | 2023-05-25 13:45:00 |
|           8 | Frankie    | item          |            NULL | Pet Grooming Supplies | 2023-05-28 09:15:00 |
|           9 | Churcil    | cash          |          200.00 | NULL                 | 2023-06-01 14:00:00 |
|          10 | Niti       | item          |            NULL | Pet Medications      | 2023-06-05 17:40:00 |
|          11 | Kelvin     | NULL          |         1000.00 | NULL                 | 2023-12-22 00:00:00 |
+-------------+------------+---------------+-----------------+----------------------+---------------------+
11 rows in set (0.00 sec)
```

```
participant_enter = 10

def generate_participant_number():
    global participant_enter
```

```
    participant_enter += 1
    return participant_enter

def manage_adoption_event():
    connection = create_connection()
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute("SELECT * FROM adoption_events")
            events = cursor.fetchall()

            print("Upcoming Adoption Events:")
            for event in events:
                print(f"Event ID: {event[0]}, Date: {event[1]}, Location:
{event[2]}")

            participant_no=generate_participant_number()
            event_id = int(input("Enter the Event ID to register: "))
            participant_name = input("Enter your name: ")

            cursor.execute("INSERT INTO participants (participant_id,
event_id, participant_name ) VALUES (%s, %s, %s)",
                           (participant_no, event_id, participant_name))
            connection.commit()

            print("Registration successful!")

        except (Error, ValueError) as e:
            print(f"Error managing adoption event: {e}")
        finally:
            connection.close()
manage_adoption_event()
```

OUTPUT

```
PS C:\python\Coding Assesment1> & C:/Users/welcome/AppData/Local/Programs/Python/Python312/python.exe "c:/python/Coding Assesment1/databasefile.py"
Upcoming Adoption Events:
Event ID: 1, Date: Kolkata Canine Carnival, Location: 2023-07-08 09:30:00
Event ID: 2, Date: Pune Paw Fest, Location: 2023-08-12 13:15:00
Event ID: 3, Date: Jaipur Feline Fiesta, Location: 2023-10-05 16:45:00
Event ID: 4, Date: Goa Doggie Day Out, Location: 2023-11-18 10:00:00
Event ID: 5, Date: Pet Carnival Mumbai, Location: 2023-06-15 10:00:00
Event ID: 6, Date: Bangalore Pet Expo, Location: 2023-09-20 12:30:00
Event ID: 7, Date: Delhi Adoption Drive, Location: 2023-12-10 14:00:00
Event ID: 8, Date: Hyderabad Pet Fair, Location: 2024-03-25 11:45:00
Event ID: 9, Date: Chennai Mega Adoption Event, Location: 2024-12-15 15:20:00
Event ID: 10, Date: Chennai Meow Mixer, Location: 2024-04-02 16:10:00
Enter the Event ID to register: 9
Enter your name: Vishnu
Registration successful!
PS C:\python\Coding Assesment1>
```

```
ysql> select * from participants;
+----------------+-----------------------+------------------+----------+
| Participant_id | Participant_Name      | Participant_Type | Event_id |
+----------------+-----------------------+------------------+----------+
|              1 | Pet Lovers Club       | Adopter          |        1 |
|              2 | Happy Tails Foundation | Adopter          |        2 |
|              3 | Doggy Delight Rescue  | Adopter          |        3 |
|              4 | Cat Haven             | Shelter          |        4 |
|              5 | Pet Educators Network | Shelter          |        1 |
|              6 | Stray Angels          | Shelter          |        5 |
|              7 | Rescue Rangers        | Shelter          |        7 |
|              8 | Bark Buddies Team     | Adopter          |        9 |
|              9 | Caring Canines Club   | Adopter          |        8 |
|             10 | Meow Manor Volunteers | Shelter          |        6 |
|             11 | Vishnu                | NULL             |        9 |
+----------------+-----------------------+------------------+----------+
1 rows in set (0.00 sec)
```