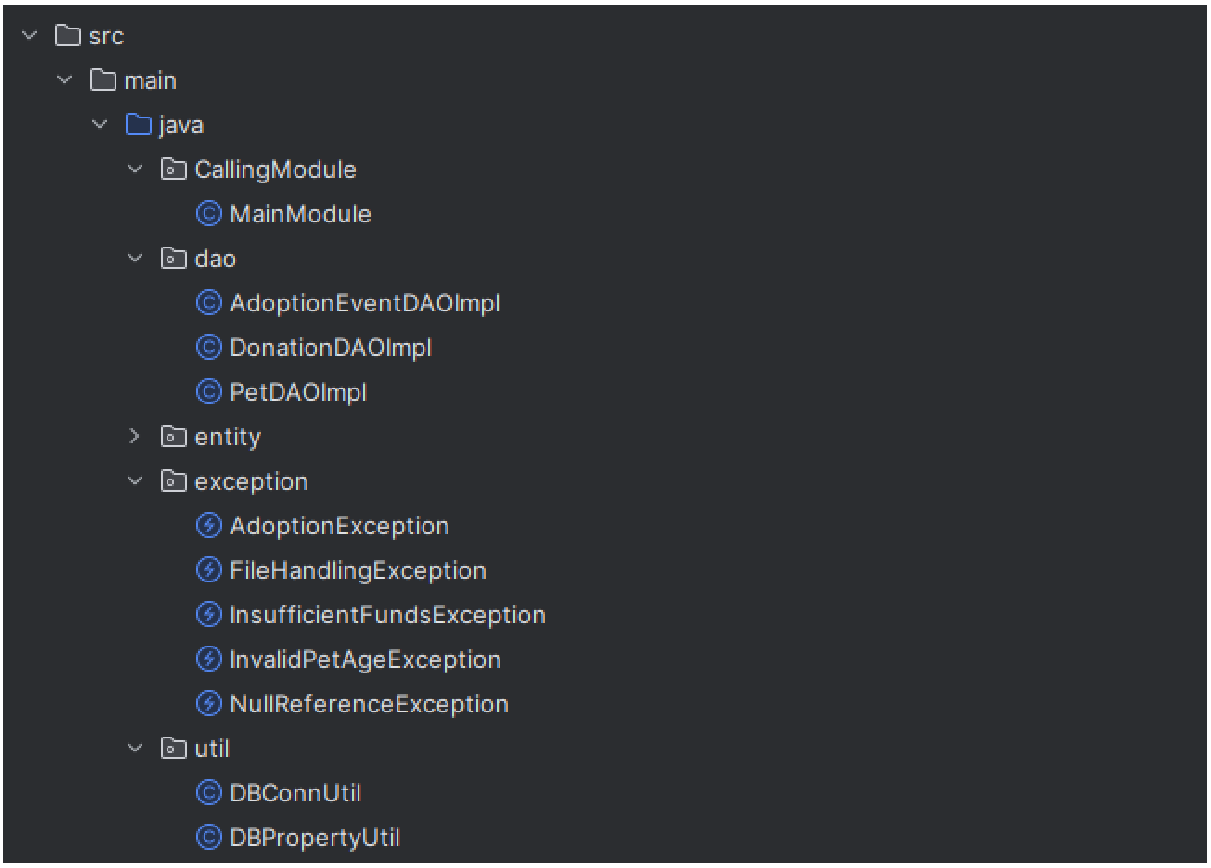
Java coding challenge – Pet Pals

Melvin Jones

Key Features

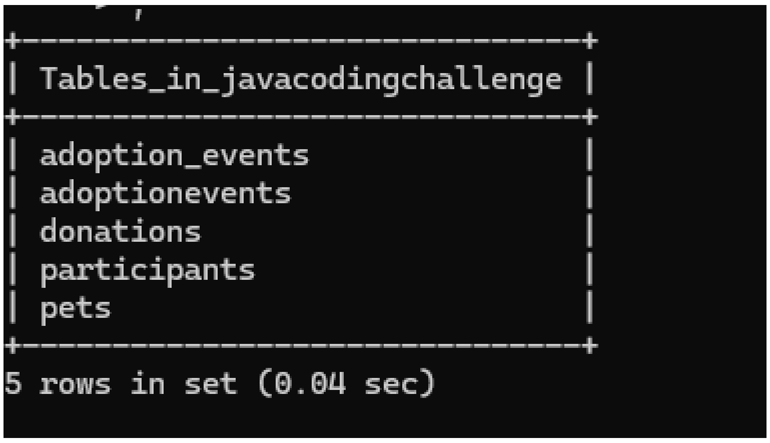
- Listing pets for adoption.
- Managing donations for shelters.
- Hosting and managing adoption events.

My project Structure :



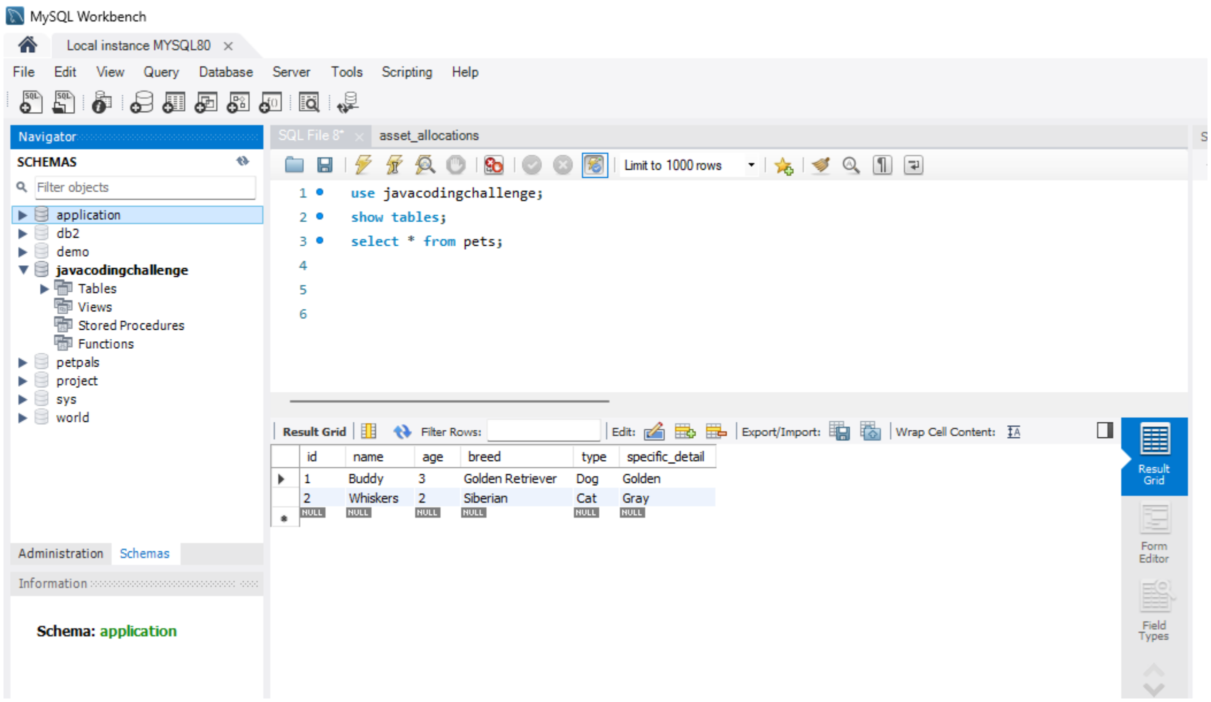
```
src
  main
    java
      CallingModule
        MainModule
      dao
        AdoptionEventDAOImpl
        DonationDAOImpl
        PetDAOImpl
      entity
      exception
        AdoptionException
        FileHandlingException
        InsufficientFundsException
        InvalidPetAgeException
        NullReferenceException
      util
        DBConnUtil
        DBPropertyUtil
```

Created SQL Schema for the tables mentioned in the document



```
+-------------------------------+
| Tables_in_javacodingchallenge |
+-------------------------------+
| adoption_events               |
| adoptionevents                |
| donations                     |
| participants                  |
| pets                          |
+-------------------------------+
5 rows in set (0.04 sec)
```

Inserted values to the tables, such as pets, donations, etc..



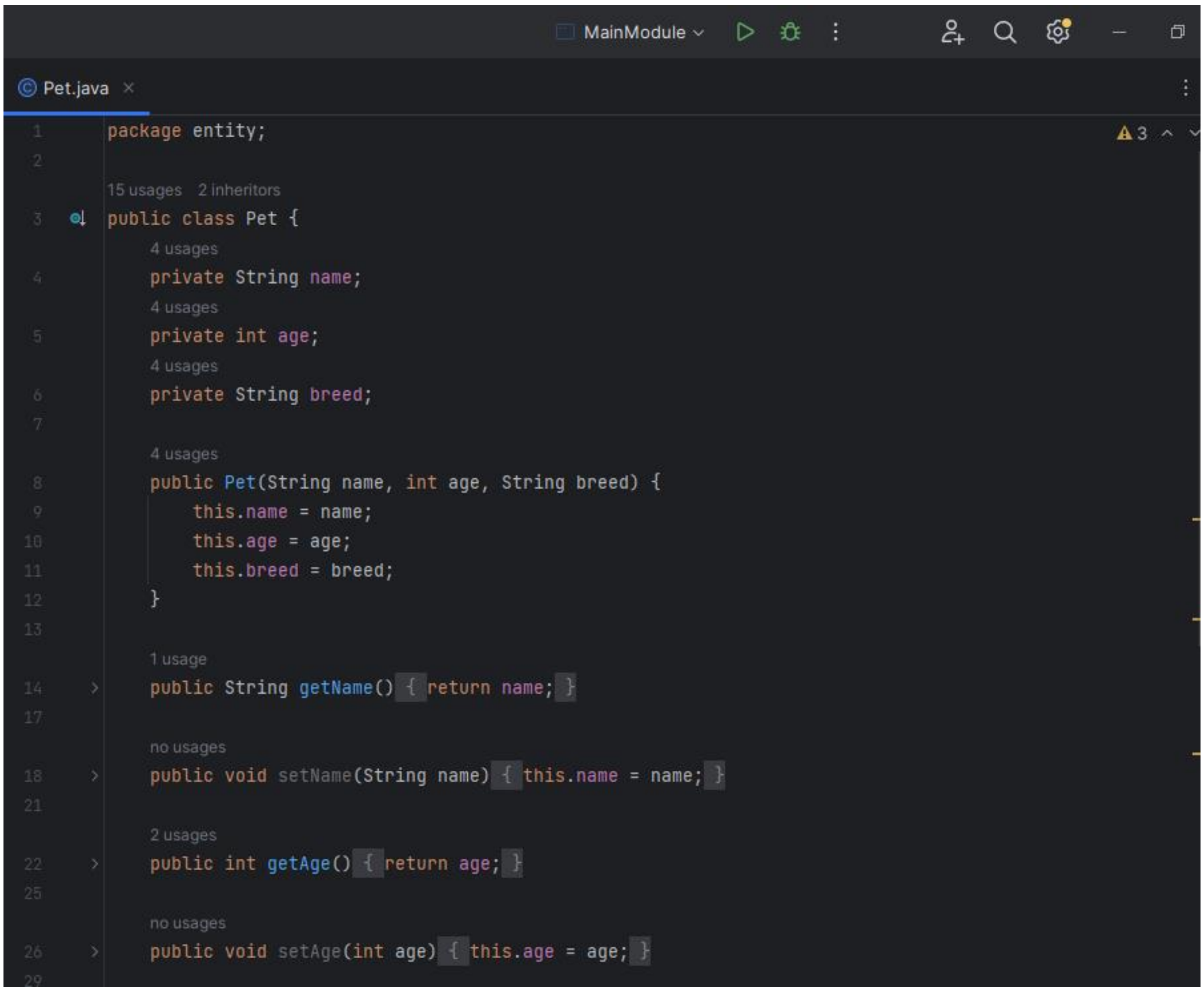
Schema: application

```sql
use javacodingchallenge;
show tables;
select * from pets;
```

id	name	age	breed	type	specific_detail
1	Buddy	3	Golden Retriever	Dog	Golden
2	Whiskers	2	Siberian	Cat	Gray
NULL	NULL	NULL	NULL	NULL	NULL

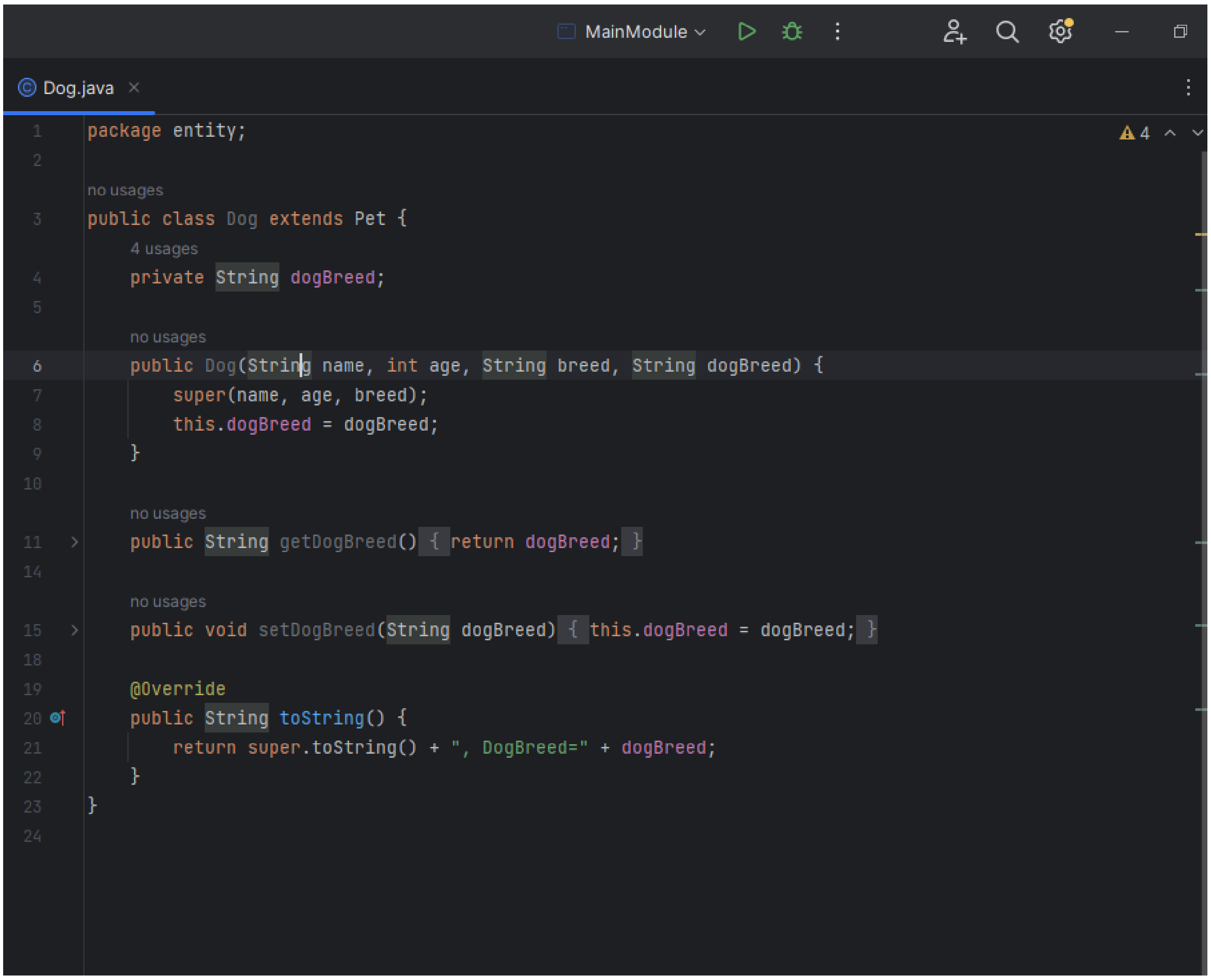
ENTITY PACKAGE :

Created pet.java and implemented the getter setter methods, for it



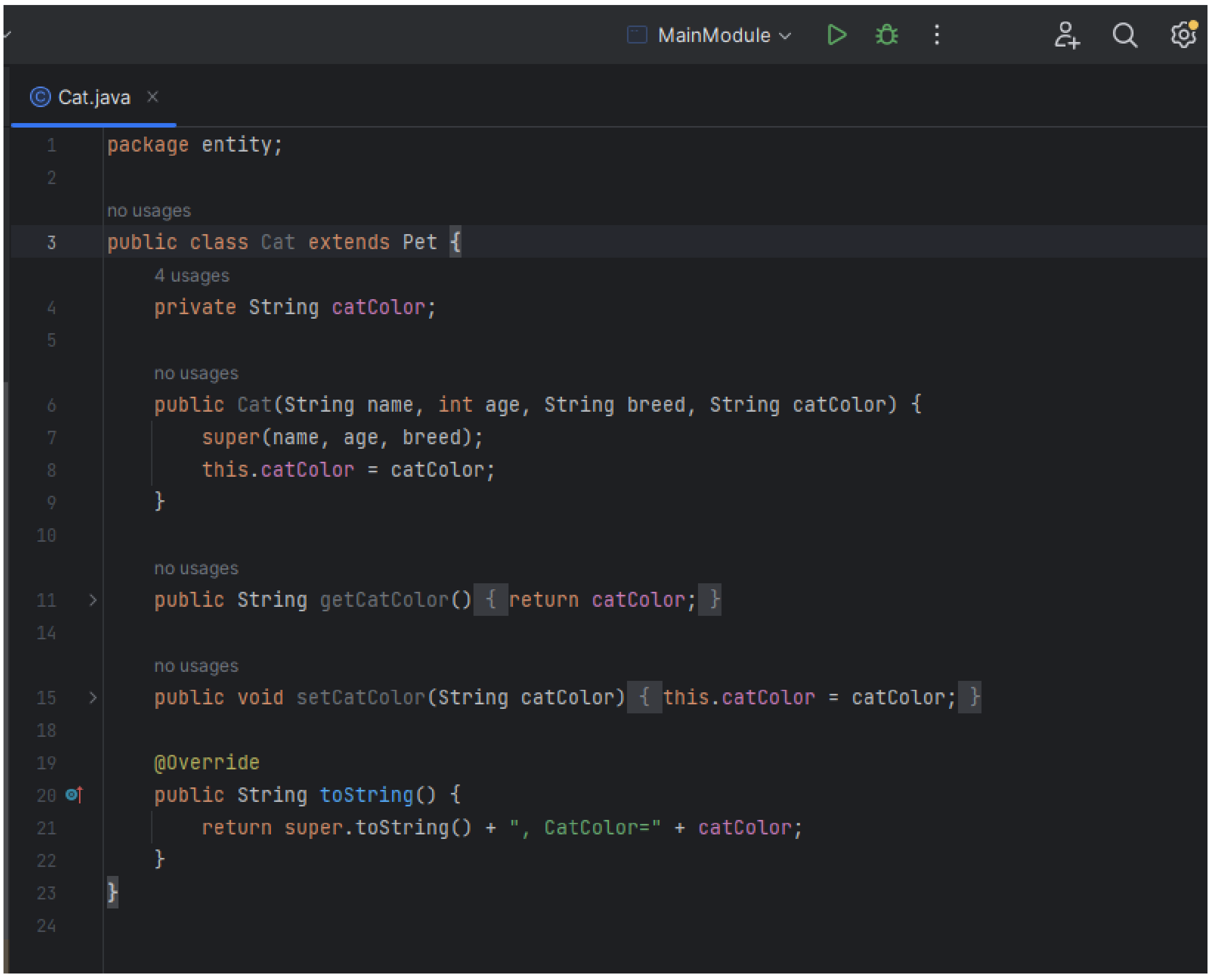
```java
package entity;

15 usages   2 inheritors
public class Pet {
    4 usages
    private String name;
    4 usages
    private int age;
    4 usages
    private String breed;

    4 usages
    public Pet(String name, int age, String breed) {
        this.name = name;
        this.age = age;
        this.breed = breed;
    }

    1 usage
    public String getName() { return name; }

    no usages
    public void setName(String name) { this.name = name; }

    2 usages
    public int getAge() { return age; }

    no usages
    public void setAge(int age) { this.age = age; }
```

Created Dog.java which inherits the properties of pet class



```java
package entity;

public class Dog extends Pet {
    private String dogBreed;

    public Dog(String name, int age, String breed, String dogBreed) {
        super(name, age, breed);
        this.dogBreed = dogBreed;
    }

    public String getDogBreed() { return dogBreed; }

    public void setDogBreed(String dogBreed) { this.dogBreed = dogBreed; }

    @Override
    public String toString() {
        return super.toString() + ", DogBreed=" + dogBreed;
    }
}
```

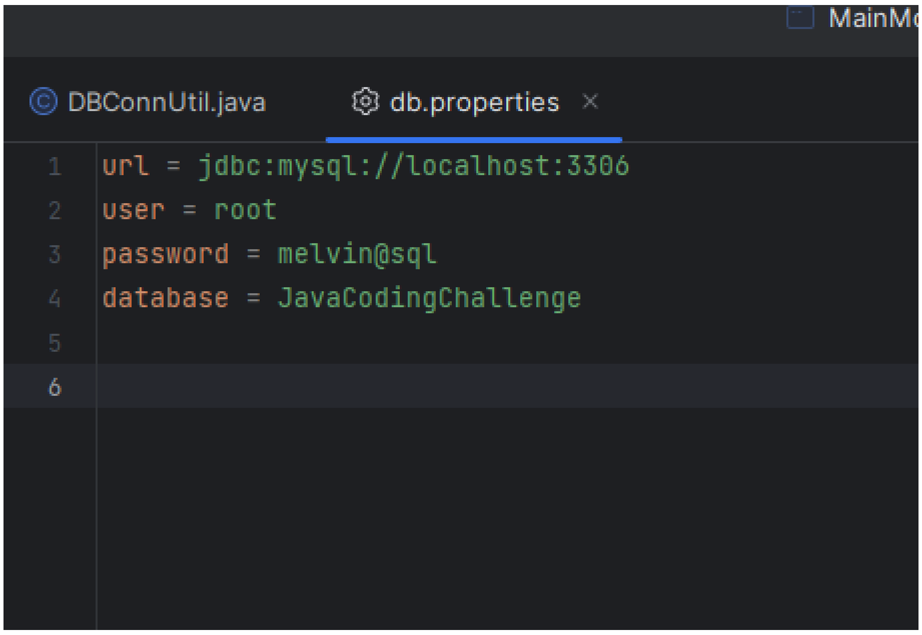
Similarly, for cat.java which inherits the pet class



```java
package entity;

no usages
public class Cat extends Pet {
    4 usages
    private String catColor;

    no usages
    public Cat(String name, int age, String breed, String catColor) {
        super(name, age, breed);
        this.catColor = catColor;
    }

    no usages
    public String getCatColor() { return catColor; }

    no usages
    public void setCatColor(String catColor) { this.catColor = catColor; }

    @Override
    public String toString() {
        return super.toString() + ", CatColor=" + catColor;
    }
}
```

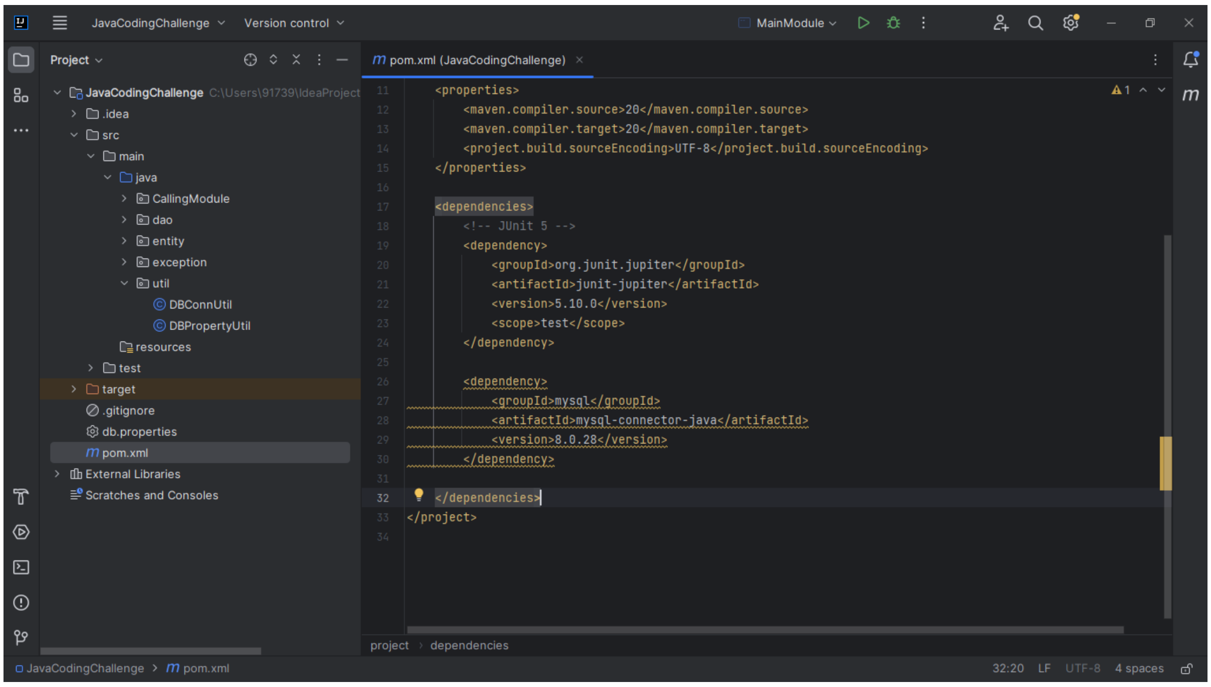
UTIL PACKAGE

Next created Util package to make database connection , then created db.properties file



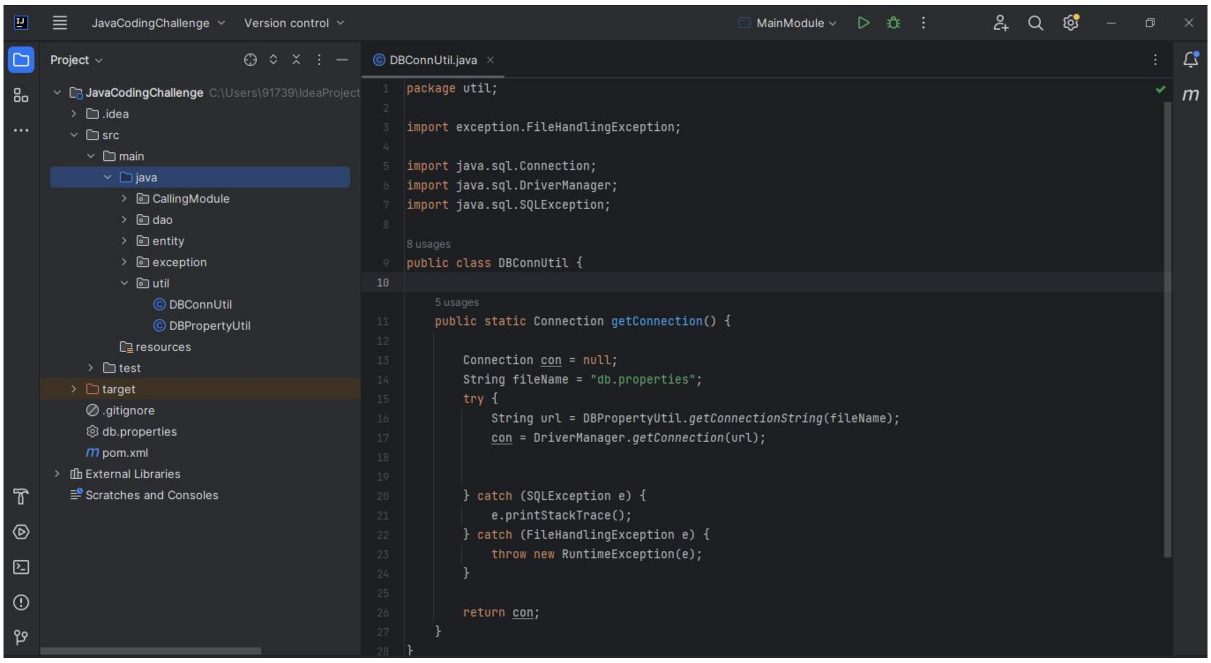
```
url = jdbc:mysql://localhost:3306
user = root
password = melvin@sql
database = JavaCodingChallenge
```

Added Mysql connector with the help of maven , added dependency in pom.xml file



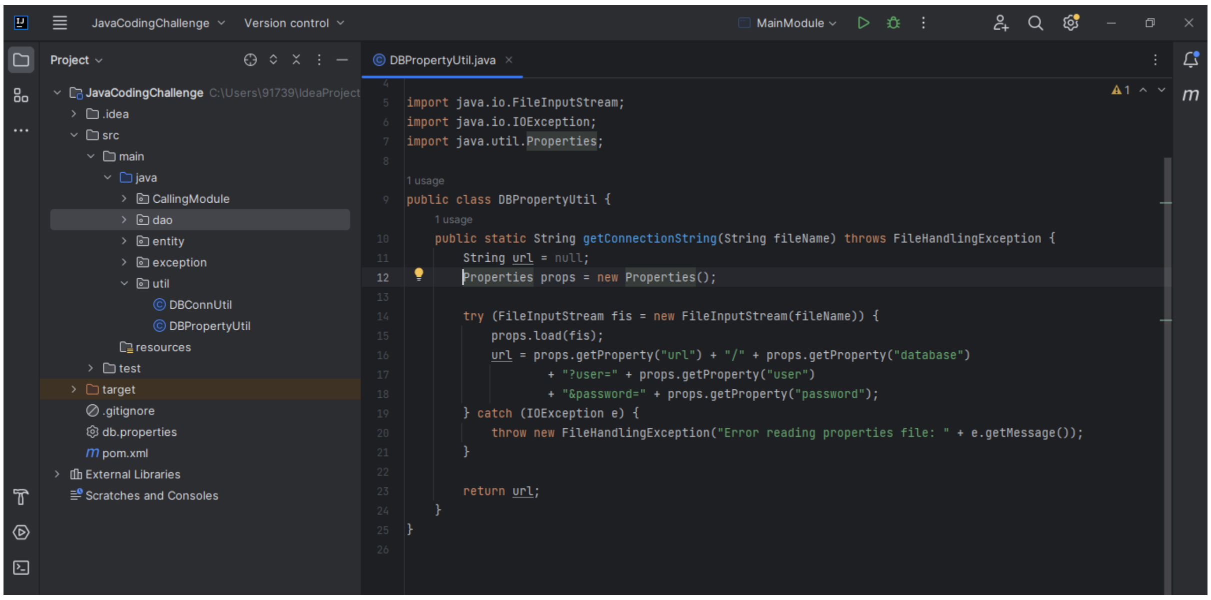
```
    <properties>
        <maven.compiler.source>20</maven.compiler.source>
        <maven.compiler.target>20</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <!-- JUnit 5 -->
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.28</version>
        </dependency>

    </dependencies>
</project>
```

Created DBConnutil.java class to establish a connection



```java
package util;

import exception.FileHandlingException;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnUtil {

    public static Connection getConnection() {

        Connection con = null;
        String fileName = "db.properties";
        try {
            String url = DBPropertyUtil.getConnectionString(fileName);
            con = DriverManager.getConnection(url);


        } catch (SQLException e) {
            e.printStackTrace();
        } catch (FileHandlingException e) {
            throw new RuntimeException(e);
        }

        return con;
    }
}
```

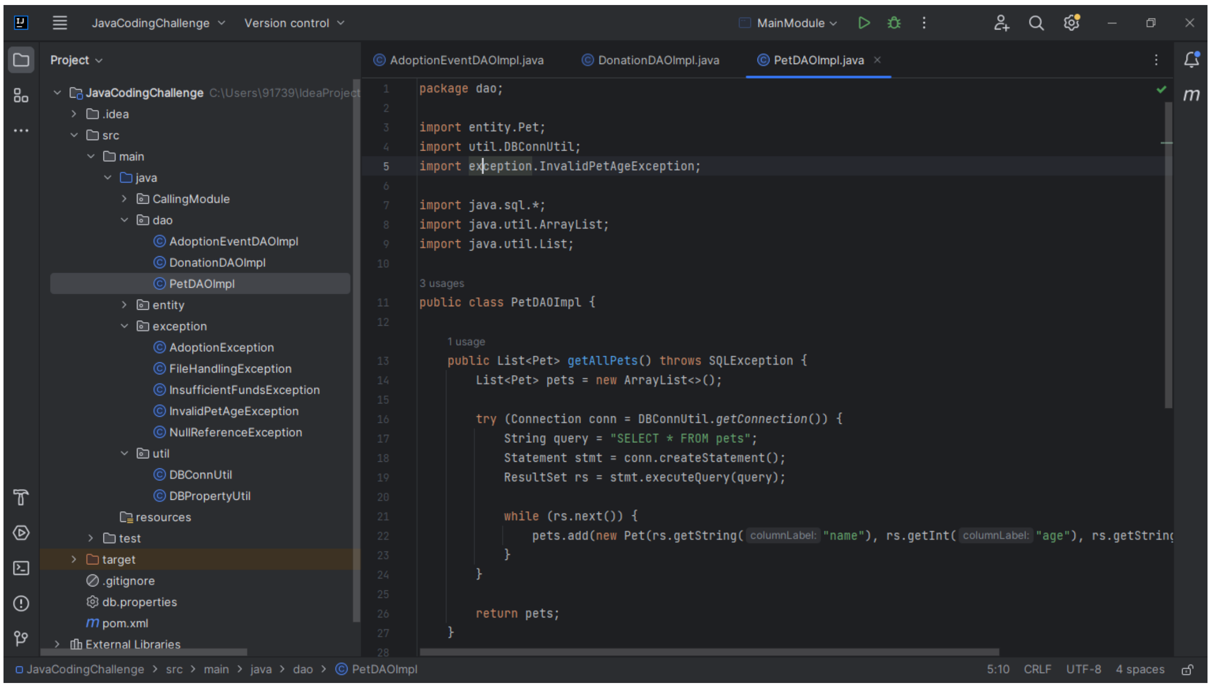
Created the DBproperty.util to create the connection string



```java
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Properties;

public class DBPropertyUtil {
    public static String getConnectionString(String fileName) throws FileHandlingException {
        String url = null;
        Properties props = new Properties();

        try (FileInputStream fis = new FileInputStream(fileName)) {
            props.load(fis);
            url = props.getProperty("url") + "/" + props.getProperty("database")
                    + "?user=" + props.getProperty("user")
                    + "&password=" + props.getProperty("password");
        } catch (IOException e) {
            throw new FileHandlingException("Error reading properties file: " + e.getMessage());
        }

        return url;
    }
}
```

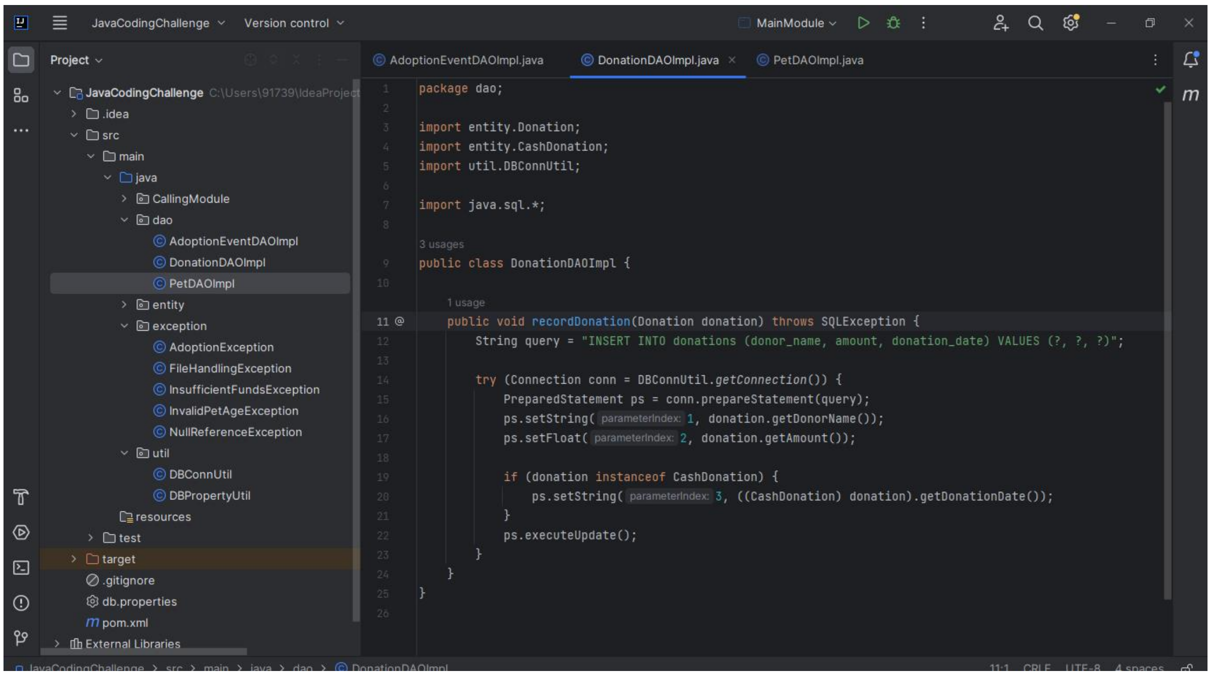
DAO PACKAGE

Created Dao package to implement the methods of List pets, Manage Donation, Manage Events



```java
package dao;

import entity.Pet;
import util.DBConnUtil;
import exception.InvalidPetAgeException;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class PetDAOImpl {

    public List<Pet> getAllPets() throws SQLException {
        List<Pet> pets = new ArrayList<>();

        try (Connection conn = DBConnUtil.getConnection()) {
            String query = "SELECT * FROM pets";
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            while (rs.next()) {
                pets.add(new Pet(rs.getString( columnLabel: "name"), rs.getInt( columnLabel: "age"), rs.getStrin
            }
        }

        return pets;
    }
```

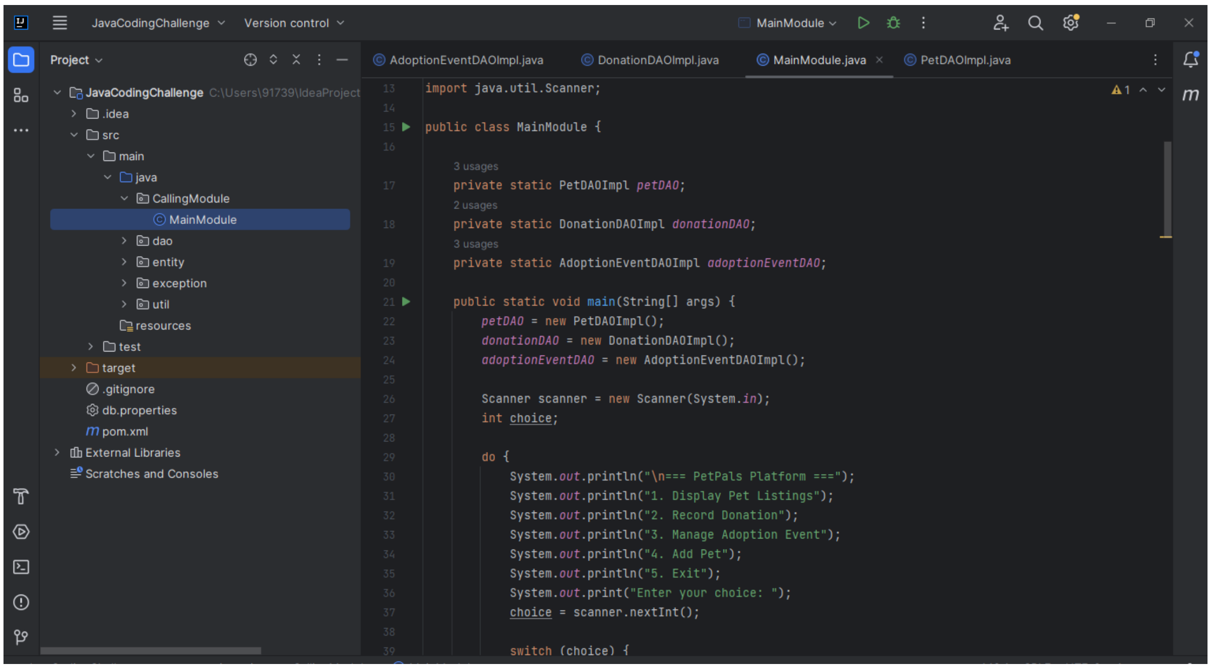
Similarly, for DonationDAImpl



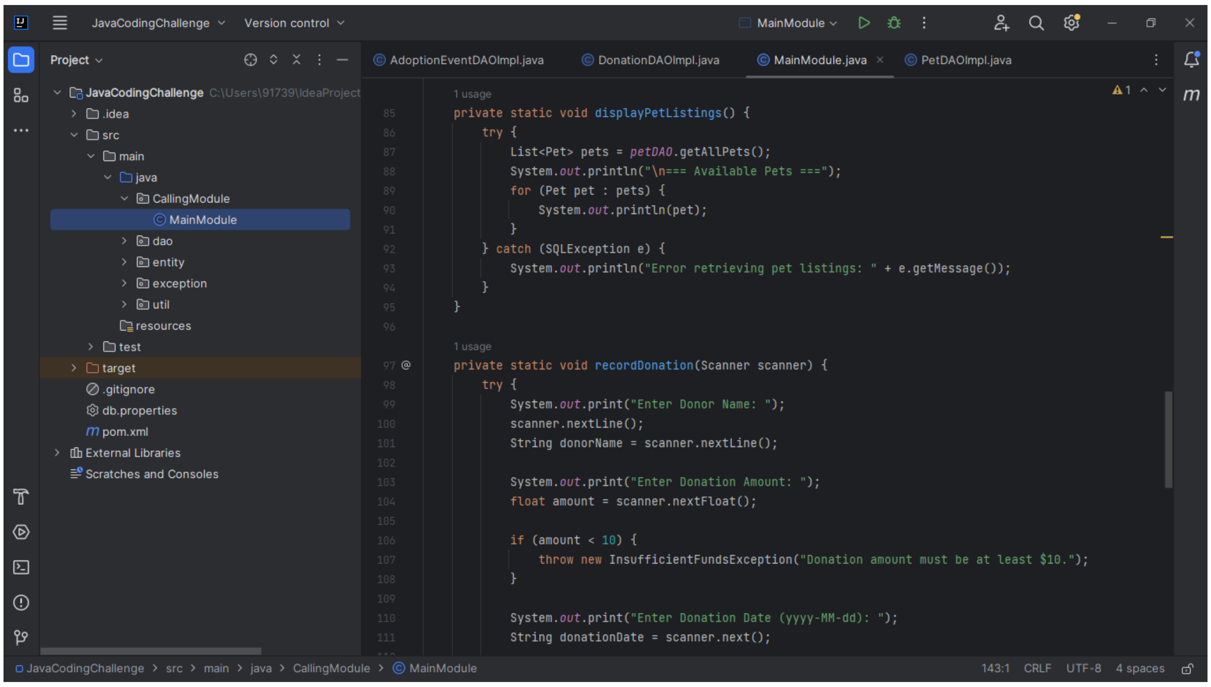
```java
package dao;

import entity.Donation;
import entity.CashDonation;
import util.DBConnUtil;

import java.sql.*;

public class DonationDAOImpl {

    public void recordDonation(Donation donation) throws SQLException {
        String query = "INSERT INTO donations (donor_name, amount, donation_date) VALUES (?, ?, ?)";

        try (Connection conn = DBConnUtil.getConnection()) {
            PreparedStatement ps = conn.prepareStatement(query);
            ps.setString( parameterIndex: 1, donation.getDonorName());
            ps.setFloat( parameterIndex: 2, donation.getAmount());

            if (donation instanceof CashDonation) {
                ps.setString( parameterIndex: 3, ((CashDonation) donation).getDonationDate());
            }
            ps.executeUpdate();
        }
    }
}
```

CALLINGMODULE PACKAGE

Next , Created the Main module to call the methods



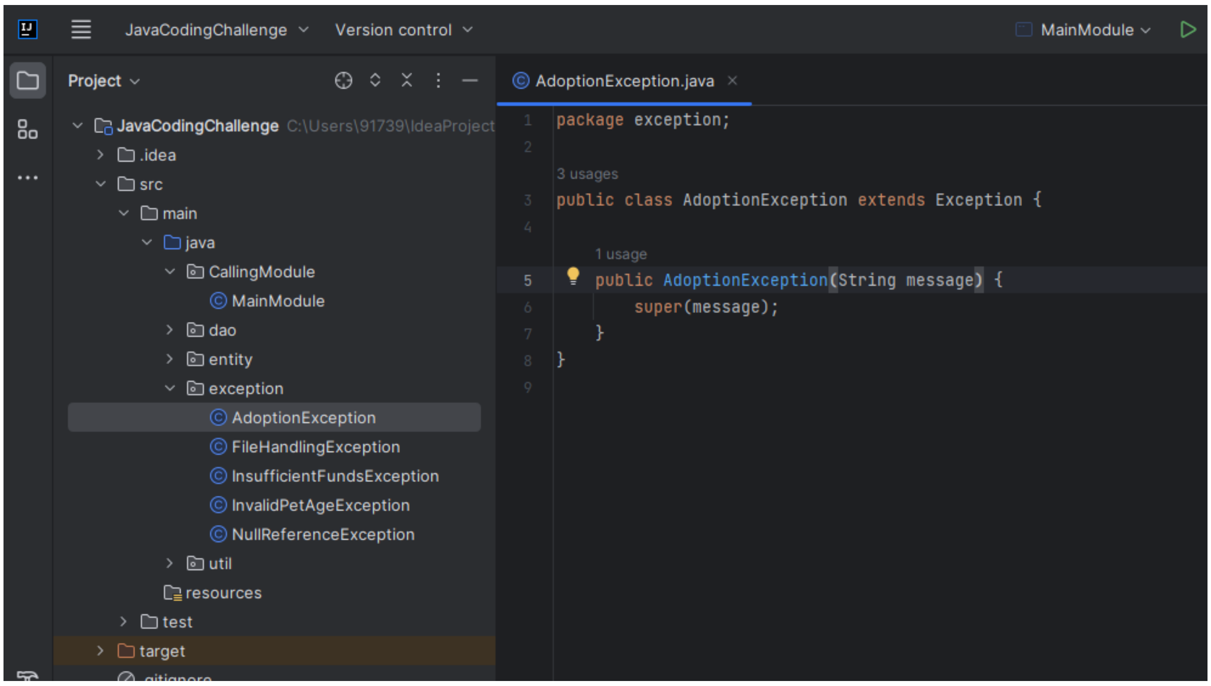
```java
import java.util.Scanner;

public class MainModule {

    3 usages
    private static PetDAOImpl petDAO;
    2 usages
    private static DonationDAOImpl donationDAO;
    3 usages
    private static AdoptionEventDAOImpl adoptionEventDAO;

    public static void main(String[] args) {
        petDAO = new PetDAOImpl();
        donationDAO = new DonationDAOImpl();
        adoptionEventDAO = new AdoptionEventDAOImpl();

        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n=== PetPals Platform ===");
            System.out.println("1. Display Pet Listings");
            System.out.println("2. Record Donation");
            System.out.println("3. Manage Adoption Event");
            System.out.println("4. Add Pet");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
```



```java
    1 usage
    private static void displayPetListings() {
        try {
            List<Pet> pets = petDAO.getAllPets();
            System.out.println("\n=== Available Pets ===");
            for (Pet pet : pets) {
                System.out.println(pet);
            }
        } catch (SQLException e) {
            System.out.println("Error retrieving pet listings: " + e.getMessage());
        }
    }

    1 usage
    private static void recordDonation(Scanner scanner) {
        try {
            System.out.print("Enter Donor Name: ");
            scanner.nextLine();
            String donorName = scanner.nextLine();

            System.out.print("Enter Donation Amount: ");
            float amount = scanner.nextFloat();

            if (amount < 10) {
                throw new InsufficientFundsException("Donation amount must be at least $10.");
            }

            System.out.print("Enter Donation Date (yyyy-MM-dd): ");
            String donationDate = scanner.next();
```

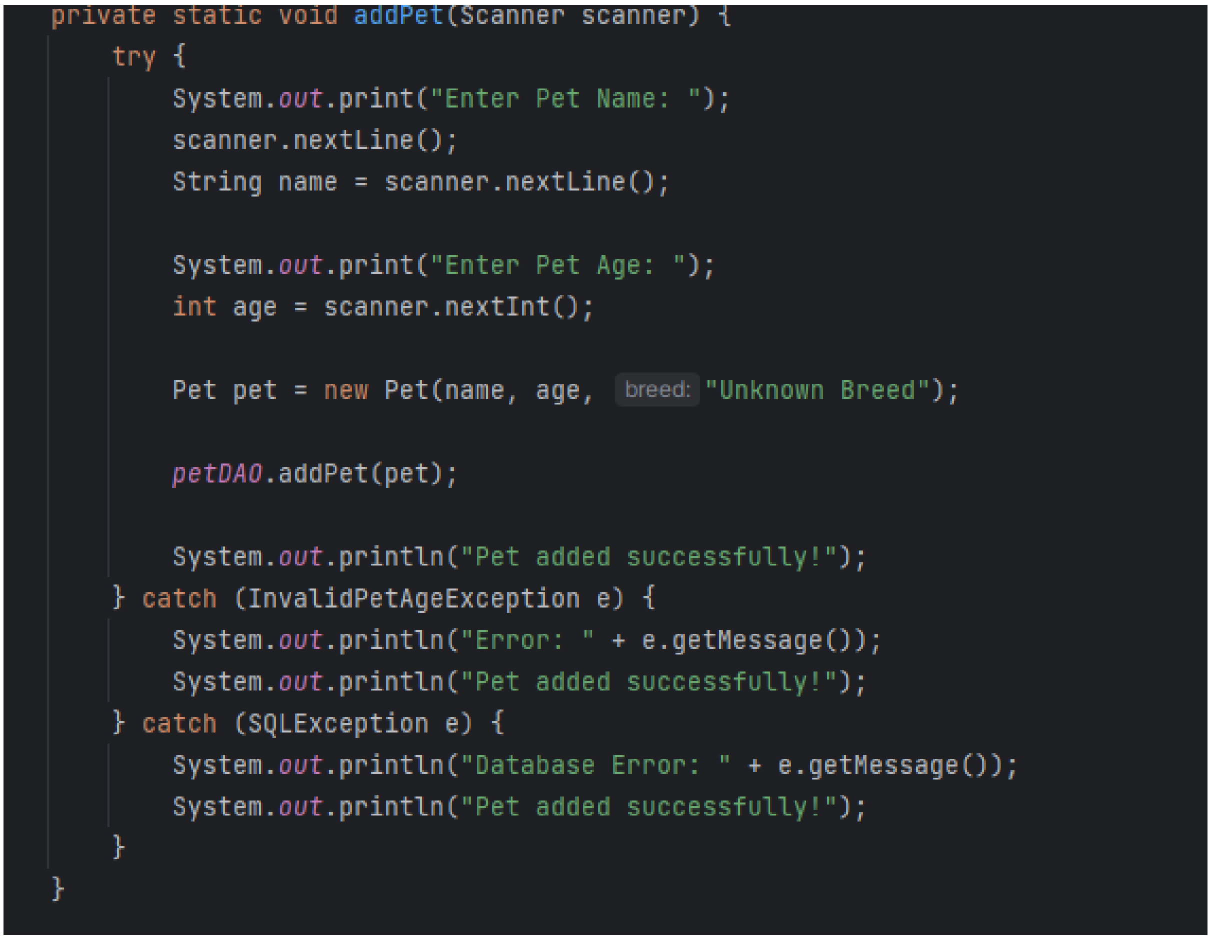
EXCEPTIONS PACKAGE

Then created exceptions for the required class in the exceptions package



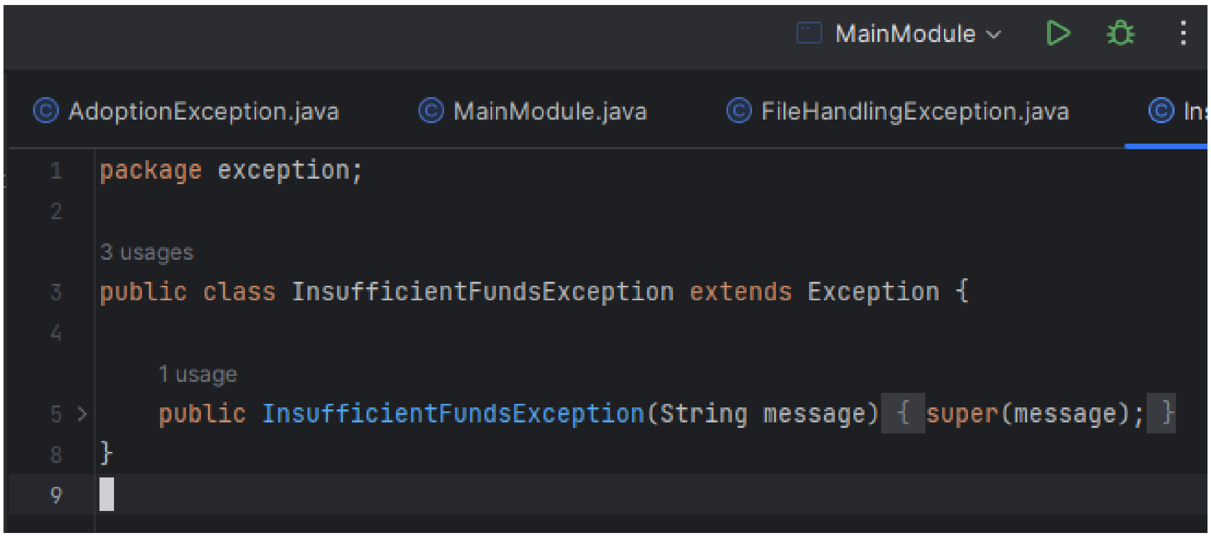
```java
package exception;

3 usages
public class AdoptionException extends Exception {

    1 usage
    public AdoptionException(String message) {
        super(message);
    }
}
```

Implemented the exceptions in Adoption method to check the age of pet if it is less than 0 then exception will be thrown



```java
private static void addPet(Scanner scanner) {
    try {
        System.out.print("Enter Pet Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();

        System.out.print("Enter Pet Age: ");
        int age = scanner.nextInt();

        Pet pet = new Pet(name, age, breed: "Unknown Breed");

        petDAO.addPet(pet);

        System.out.println("Pet added successfully!");
    } catch (InvalidPetAgeException e) {
        System.out.println("Error: " + e.getMessage());
        System.out.println("Pet added successfully!");
    } catch (SQLException e) {
        System.out.println("Database Error: " + e.getMessage());
        System.out.println("Pet added successfully!");
    }
}
```

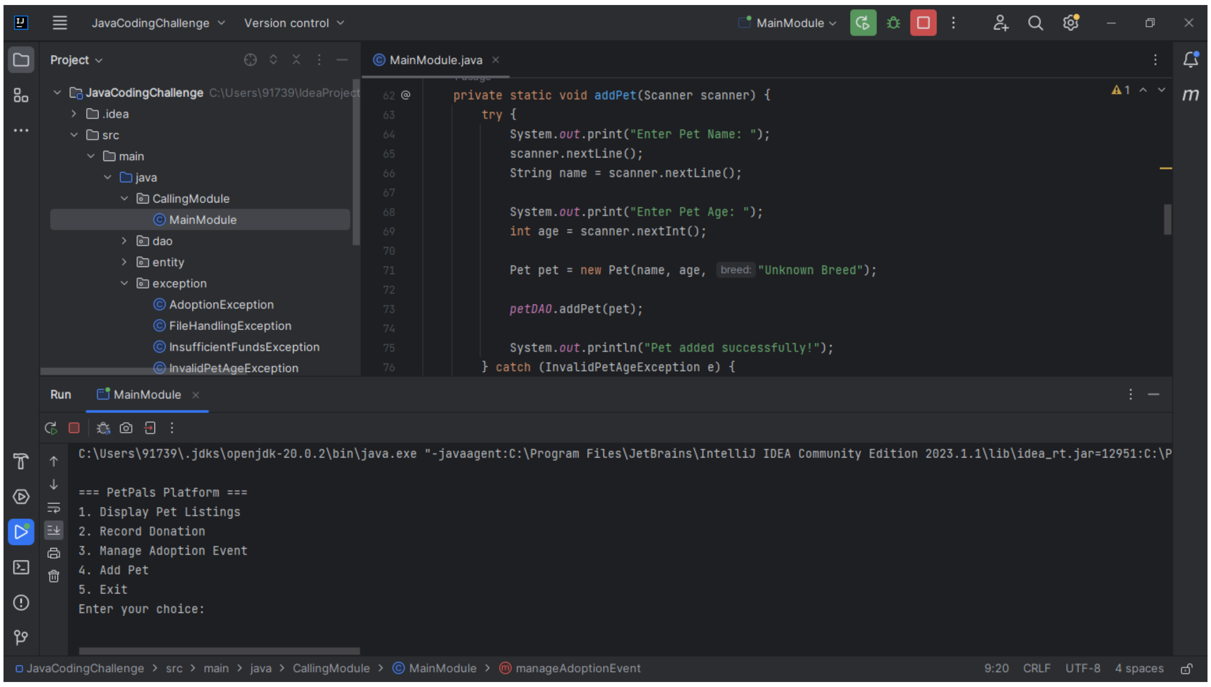
Likewise created exceptions for all the required methods one by one



```java
package exception;

public class InsufficientFundsException extends Exception {

    public InsufficientFundsException(String message) { super(message); }
}

```

RESULTS

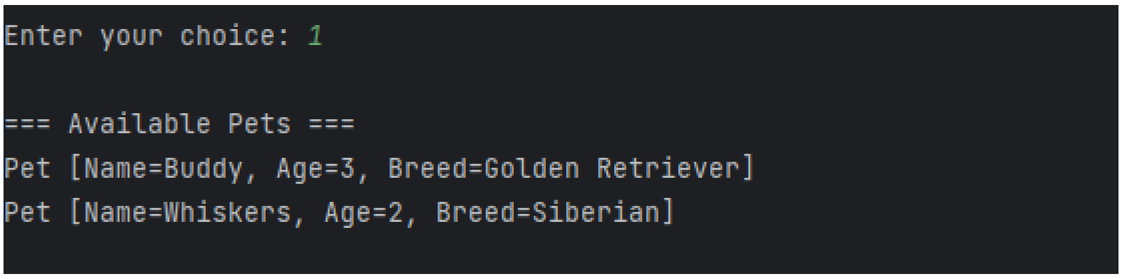
Finally executed the program and main method is triggered and the output is displayed



```java
    private static void addPet(Scanner scanner) {
        try {
            System.out.print("Enter Pet Name: ");
            scanner.nextLine();
            String name = scanner.nextLine();

            System.out.print("Enter Pet Age: ");
            int age = scanner.nextInt();

            Pet pet = new Pet(name, age, breed: "Unknown Breed");

            petDAO.addPet(pet);

            System.out.println("Pet added successfully!");
        } catch (InvalidPetAgeException e) {
```

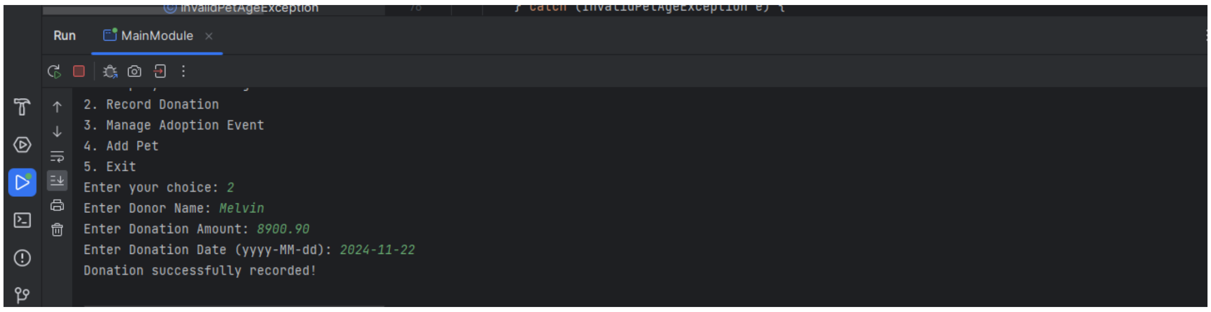
```
C:\Users\91739\.jdks\openjdk-20.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.1\lib\idea_rt.jar=12951:C:\P

=== PetPals Platform ===
1. Display Pet Listings
2. Record Donation
3. Manage Adoption Event
4. Add Pet
5. Exit
Enter your choice:
```

Selecting case 1



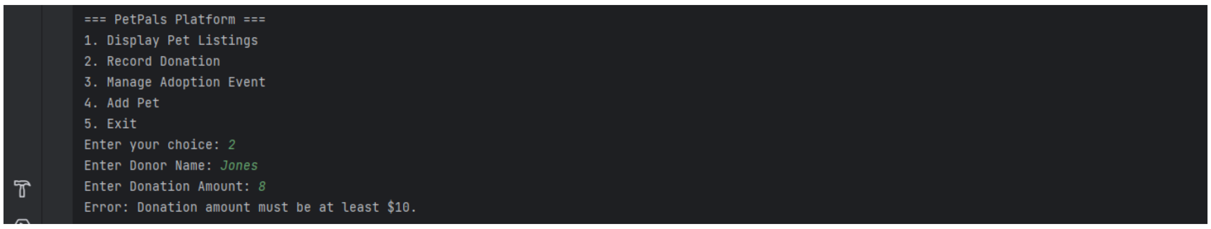
```
Enter your choice: 1

=== Available Pets ===
Pet [Name=Buddy, Age=3, Breed=Golden Retriever]
Pet [Name=Whiskers, Age=2, Breed=Siberian]
```

Selecting case 2



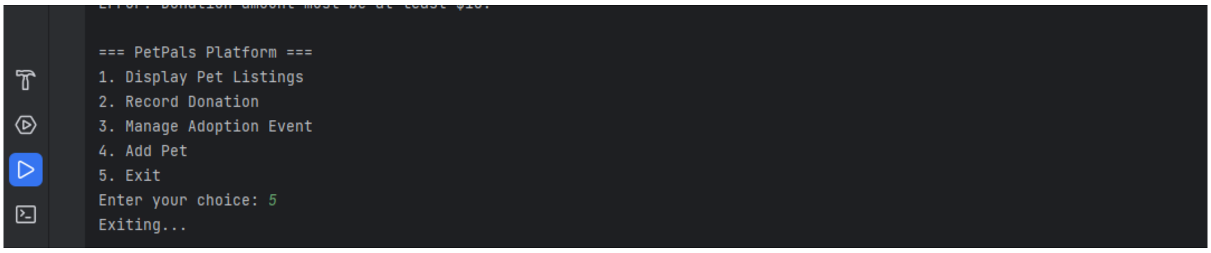
```
2. Record Donation
3. Manage Adoption Event
4. Add Pet
5. Exit
Enter your choice: 2
Enter Donor Name: Melvin
Enter Donation Amount: 8900.90
Enter Donation Date (yyyy-MM-dd): 2024-11-22
Donation successfully recorded!
```

Getting an exception if donation amount is less than 10



```
=== PetPals Platform ===
1. Display Pet Listings
2. Record Donation
3. Manage Adoption Event
4. Add Pet
5. Exit
Enter your choice: 2
Enter Donor Name: Jones
Enter Donation Amount: 8
Error: Donation amount must be at least $10.
```

Case 5 to exit the program



```
=== PetPals Platform ===
1. Display Pet Listings
2. Record Donation
3. Manage Adoption Event
4. Add Pet
5. Exit
Enter your choice: 5
Exiting...
```

Database getting updated after the output.

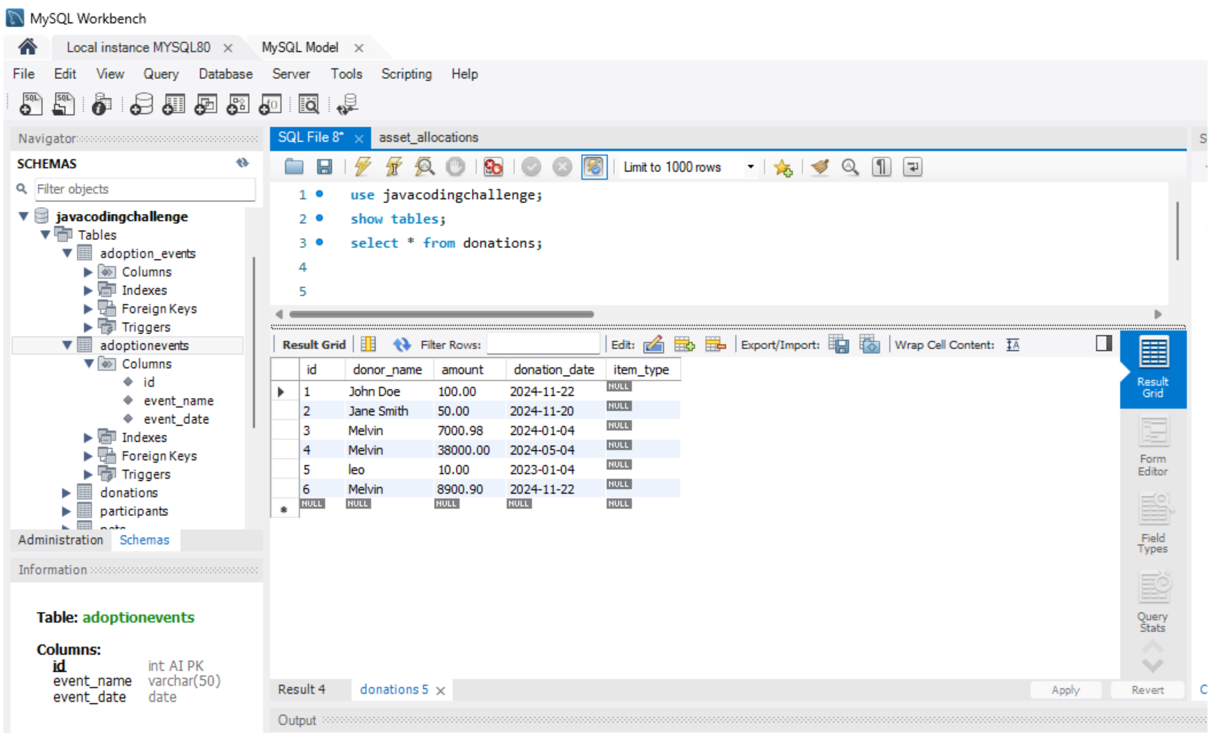


Table: adoptionevents

Columns:

id	int AI PK
event_name	varchar(50)
event_date	date

```
use javacodingchallenge;
show tables;
select * from donations;
```

id	donor_name	amount	donation_date	item_type
1	John Doe	100.00	2024-11-22	NULL
2	Jane Smith	50.00	2024-11-20	NULL
3	Melvin	7000.98	2024-01-04	NULL
4	Melvin	38000.00	2024-05-04	NULL
5	leo	10.00	2023-01-04	NULL
6	Melvin	8900.90	2024-11-22	NULL
NULL	NULL	NULL	NULL	NULL

Finally, I have successfully implemented the methods and the Output is displayed as expected.