# Django Second Assignment

##### This Assignment will cover: Django Models, and GET/POST request

Consider you have to Develop a Software for Managing a Zoo. The zoo is divided into 3 categories Mammals, Birds, and Fish life. Zoo wants to manage the following features for each category:<br />

Mammal<br />

1. Name (name) -> String<br />

2. Species (species) -> String<br />

3. The food they eat (food) -> string<br />

4. Last Feed Time (last\_feed\_time)-> Date Time<br />

5. Gender (gender) -> 2 choices (M, F)<br />

Bird<br />

1. Name (name) -> String<br />

2. Species (species) -> String<br />

3. The food they eat (food)-> String<br />

4. Last Feed Time (last\_feed\_time)-> Date Time<br />

Fish<br />

1. Color (color) -> String<br />

2. Species (species) -> String<br />

3. The Food they eat (food) -> String<br />

4. Count (count) -> Integer<br />

5. Last Feed Time (last\_feed\_time) > Date Time<br />

\*\*Reason there is no name in the Fish is because Fish are majory in groups in a Zoo. Hence it has a count\*\*

### Assignment:

1. Create a Django app named animals<br />

2. Create 3 models in animals app -> Mammal, Bird and Fish with features mentioned above<br />

3. Do not change Database changes in settings.py<br />

4. Run makeMigrations command<br />

5. Finally, run migrate command to create tables in the database<br />

6. Create views to Get and POST Data for Mammals, Bird, Fish Models Hint: Use ORM<br />

7. For Views and Urls More Details are given below <br />

##### Some Guide for urls.py:<br />

1. Name in Django URL -> name is used for accessing that URL from your Django / Python code.<br />

2. For example you have this in animals/urls.py<br />

url(r'^main/', views.main, name='main')

3. In point 2, "main" is the name of this URL

4. URL namespaces -> URL namespaces allow you to uniquely reverse named URL patterns even if different applications use the same URL names. At Line 21 in zoo/urls.py, path('', include((animals.urls', animals),namespace=animals) ), namespace=animals is set Remember, Both Name and Namespace are important for the assignment

5. For this assignment's test cases we have used some predefined names and namespace

6. Those names are defined below in the problem statement describes as "Django Url name"

7. Please make sure you use those names in Django URLs or else your assignment's test will fail

Summary: \*\*Namespace\*\* is "animals" set in zoo/urls.py file and \*\*name\*\* is to be set for each URL in animals/urls which is stated in the problem statement

#### Views: Expected I/O

\*\*Part 1: GET and POST request for fetching Mammals\*\*

\*\*Django URL name: mammals\*\*<br />

No parameter<br />

\*\*Actions\*\*:<br />

1. GET - get list of all mammals<br />

2. POST - Add a new mammal to the Zoo<br />

<br />

1. GET:

Response Data format {"data": [name,species,gender,food]}

If no data is present

/animals/mammals/

Response -> {‘data’: []}

/animals/mammals

Response ->{"data": [["lion", "Panthera", "M", "meat"]]}

2. POST

Response Data format {"created": {"name": name}}

/animals/mammals/ -d "name=lion&species=Panthera&gender=M&food=meat"

Response -> {"created": {"name": "lion"}}

\*\*Part 2: GET and POST request for fetching Birds<br />\*\*

\*\*Django URL name: birds\*\*<br />

No parameter<br />

\*\*Actions\*\*

1. GET - get list of all Birds<br />

2. POST - Add a new bird to the Zoo<br />

<br />

1. GET:

Response Data format {"data": [name,species,food]}

If no data is present

/animals/birds/

Response -> {"data": []}

/animals/birds/

Response ->{"data": [["Parrot", "parrata", "insects"]]}

2. POST

Response Data format {"created": {"name": name}}

/animals/birds/ -d "name=Parrot&species=parrata&food=insects"

Response -> {"created": {"name": "Parrot"}}

\*\*Part 3: GET and POST request for fetching Fishes<br />\*\*

<br />

\*\*Django URL name: fishes\*\*<br />

No parameter<br />

\*\*Actions\*\*

1. GET - get list of all Fish species<br />

2. POST - Add a new fish species to the Zoo<br />

<br />

1. GET:

Response Data format {"data": [name,species,food]}

If no data is present

/animals/fishes/

Response -> {"data": []}

/animals/fishes/

Response ->{"data": ["yellow", "GoldFisha", "grain", 100]}

2. POST

Response Data format {"created": {"species": species}}

/animals/fishes/ -d "color=yellow&species=GoldFisha&food=grain&count=2000"

Response -> {"created": {"species": "GoldFisha"}}

## Installation Steps

1. Open up your Terminal / Command Line

2. git clone the repository

3. cd into the directory of the step (the one you just git cloned)

4. make sure you have python3 installed

5. Install virtualenv by following the steps

```

Mac

python3 -m pip install --user virtualenv

python3 -m venv env

source env/bin/activate

Windows

py -m pip install --user virtualenv

py -m venv env

.\env\Scripts\activate

```

6. Run

```

python -m pip install -r requirements/requirements.txt

```

6. No Database setup needs to be done. Do not edit Database settings in settings.py

7. If everything runs fine till here, create the application using the command

```

python manage.py startapp animals

```

8. Now uncomment the following lines

```

Line 21 in zoo/urls.py -> path('animals/', include(('animals.urls', 'animals'), namespace='animals')),

Line 40 in zoo/settings.py -> 'animals',

```

Now Create File animals/urls.py and add lines

```

from django.urls import path

from . import views

urlpatterns = []

```

9. Run

```

python manage.py makemigrations

```

10. Now Run the Server using command

```

python manage.py runserver

```

11. Great! Start Coding the assignment for above usecase

12. After Finishing the assignment, you can test them locally using command

```

coverage run --source='.' manage.py test --no-input

```

1. Push the code into the repo using git, Check Pull Requests Tab.
2. 13. Check if all the tests are Passed