Dharmsinh Desai University, Nadiad
Faculty of technology, Department of Computer Engineering
Subject : Software Project
Lab Manual

Lab 6 : Web Application development with Python and Django

Aim : Connecting to database and creating tables using Python and Django.

Requirements : Python, Django, Oracle, MySQL (Optional)

Procedure:

0. Initialize the virtual environment : (We create virtual environment so that we can work with our configuration of python and django, without effecting system's global configuration. Either create a new virtual environment or use the environment which was created in the last lab.)

1. Create Project: Inside the virtual environment create a project by applying following command,

```
$django-admin startproject mytest
```

(instead of 'myproj' you may use name appropriate for your project)

2. Configure database: Django project's database is defined in the settings.py file. In order to use oracle as the database we need to perform following steps.

2.1 Install database driver (If not already isntalled) by applying the following command.

```
$python3.6 -m pip install cx_oracle –upgrade
```

Following dependencies you may need to be installed for MySQL server,

```
$sudo apt-get install python-dev python3-dev
$sudo apt-get install libmysqlclient-dev
$pip install MySQL-python
$pip install mysqlclient --user
```

2.2 Update the settings.py file (located in project's root (mytest) directory)

To use MySQL as the default database server, make following changes,

```
DATABASES = {
      'default' :{
            'ENGINE' : 'django.db.backends.mysql',
            'NAME' : 'ce<sem4rollno>' , # Name of the database*
            'USER': 'ce<sem4rollno>',  #user name for the database
            'PASSWORD' : 'ce<sem4rollno>', # password
            'HOST' : '192.168.29.150',  # IP address of the database
            'PORT' : '3306', # port number of database server
      }
}
```

*** ce<sem4rollno> --> ce1 if your roll number is 1.**

```
78 DATABASES = {
79
80 'default' :{
81         'ENGINE' : 'django.db.backends.mysql',
82         'NAME' : 'ce1' , # Name of the database*
83         'USER': 'ce1',  #user name for the database
84         'PASSWORD' : 'ce1', # password
85         'HOST' : '192.168.29.150',  # IP address of the database
86         'PORT' : '3306', # port number of database server
87         }
88
89 }
90
```

❖ **DATABASE configuration**

- to use mysql database ypu mast have installed **mysqlclient**.you can install using following command:

$pip3 install mysqlclient --user

```
[user1@hadoop-clone mytest]$ pip3 install mysqlclient --user
Collecting mysqlclient
  Downloading https://files.pythonhosted.org/packages/de/d7/919042fb7ed78a89e21c4e43d5
daed90d5df18ec183bae3de493a7d9e3ed/mysqlclient-1.4.1.tar.gz (85kB)
    100% |████████████████████████████████| 92kB 317kB/s
Installing collected packages: mysqlclient
  Running setup.py install for mysqlclient ... done
Successfully installed mysqlclient-1.4.1
You are using pip version 18.1, however version 19.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

❖ **Installation of mysqlclient**

❖ To use Oracle as the default database server, make following changes,

```
DATABASES = {
     'default' :{
          'ENGINE' : 'django.db.backends.oracle',
          'NAME' : 'XE' , # Name of the database
          'USER': 'ce4_1',  #user name for the database
          'PASSWORD' : 'ce4_1', # password
          'HOST' : '192.168.29.152',  # IP address of the database
          'PORT' : '1521', # port number of database server
          }
     }
```

- This will set oracle database as the default database for the application.

3. Create new application : Create a new application to connect and access the database by applying the following command.

```
$python3.6 manage.py startapp dbtestapp
```

(You may replace 'dbtestapp' to a name suitable for your project)

4. Create database tables: One way to create database tables and access it in the application is through models.py file. Every application of a Django project has a models.py file, which describe the schema for the project. It describes database table as a class (Refer, ORM – Object Relation Mapper for more details). Following example shows a description for student table in models.py file. To create such table, open models.py file defined in app 'mydbtest' and enter following code.

```
from django.db import models

class Student(models.Model):
     student_name = models.CharField(max_length=100)
     student_dob = models.DateTimeField('date published')
```

You need to create one class for every table in the database.

models.py

```
1 from django.db import models
2
3 # Create your models here.
4
5 class book_master(models.Model):
6       book_id = models.IntegerField(primary_key=True)
7       book_name = models.CharField(max_length=20)
8       book_price = models.IntegerField()
```

❖ **models.py**

5. Add reference of the 'dbtestapp' in 'mytest' project : Update INSTALLED_APPS details in project's settings.py file as follows,

```
INSTALLED_APPS = [
     'dbtestapp.apps.DbtestappConfig',
     ......
     ....
]
```

```
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'dbtestapp.apps.DbtestappConfig',
41
42 ]
```

❖ **Adding reference of the 'dbtestapp'**

*** *DbtestappConfig* class is in the dbtestapp/apps.py file, so it's path is 'dbtestapp.apps.DbtestappConfig'.**

apps.py

```
1 from django.apps import AppConfig
2
3
4 class DbtestappConfig(AppConfig):
5     name = 'dbtestapp'
```

❖ **apps.py**

```
[user1@hadoop-clone mytest]$ python3.6 manage.py startapp dbtestapp
[user1@hadoop-clone mytest]$ cd mytest/
[user1@hadoop-clone mytest]$ gedit settings.py
[user1@hadoop-clone mytest]$ cd ..
[user1@hadoop-clone mytest]$ ls
dbtestapp  hello_world  manage.py  mytest
[user1@hadoop-clone mytest]$ cd dbtestapp/
[user1@hadoop-clone dbtestapp]$ gedit  models.py
[user1@hadoop-clone dbtestapp]$ cd ..
[user1@hadoop-clone mytest]$ cd mytest/
[user1@hadoop-clone mytest]$ gedit settings.py
[user1@hadoop-clone mytest]$ cd ..
[user1@hadoop-clone mytest]$ ls
dbtestapp  hello_world  manage.py  mytest
```

❖ execution flow for dbtestapp

6.Configure and build project: Supply the following commands to build the project.

```
$python3.6 manage.py makemigrations

$python3.6 manage.py migrate
```

```
[user1@hadoop-clone mytest]$ python3.6 manage.py makemigrations
Migrations for 'dbtestapp':
  dbtestapp/migrations/0001_initial.py
    - Create model book_master
[user1@hadoop-clone mytest]$ python3.6 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, dbtestapp, sessions
Running migrations:
  Applying dbtestapp.0001_initial... OK
[user1@hadoop-clone mytest]$
```

7. Check database: Step 7 creates a table named dbtestapp_student. Go to mysql database and check if the table.

```
[user1@hadoop-clone mytest]$ mysql -u ce1 -h 192.168.29.150 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6202
Server version: 5.6.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| ce1                |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> use ce1
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

❖ **mysql database changed**

```
mysql> show tables;
+----------------------------+
| Tables_in_ce1              |
+----------------------------+
| COFFEES                    |
| COFFEE_DESCRIPTIONS        |
| COFFEE_HOUSES              |
| COF_INVENTORY              |
| DATA_REPOSITORY            |
| MERCH_INVENTORY            |
| RSS_FEEDS                  |
| SUPPLIERS                  |
| Student                    |
| auth_group                 |
| auth_group_permissions     |
| auth_permission            |
| auth_user                  |
| auth_user_groups           |
| auth_user_user_permissions |
| dbtestapp_book_master      |
| django_admin_log           |
| django_content_type        |
| django_migrations          |
| django_session             |
| hello_world1_book_master   |
| staff                      |
| student_master             |
+----------------------------+
23 rows in set (0.00 sec)

mysql>
```

- **dbtestapp_book_master table created**

Exercise :

Create database tables for your project.

Reference and further reading:

https://docs.djangoproject.com/en/2.0/topics/db/models/