COMP-8347-91 I/S 2020 LAB #4
Prepared by: Dr. Saja Al Mamoori
Due Date: Sun Jun 21 (Sec. 57-59), Tues Jun 23 (Sec. 51-56)

It is time to start our Django E-learning webapp. In this lab, you will do the following tasks:

- **Create the app models, Topic, Course, Student, and Order.**
- **Update the database with new models to create corresponding tables with their relationships.**
- **Populate the tables with data from the file lab4dataset.txt attached with the lab.**
- **Query the database to retrieve information and verify the tables' data is correct.**

On BB, you should submit the file *models.py* in your project for Part 1 and a separate .txt or .pdf file for Part 3.

PART 1: Edit *models.py* to create the necessary models.

from django.db import models

import datetime

from django.contrib.auth.models import User

from django.utils **import** timezone

1. Add the following models.

a. ***Topic*** with the following fields:

```
class Topic(models.Model):
    name = models.CharField(max_length=200)
```

b. ***Course*** with the following fields:

```
class Course(models.Model):
    topic = models.ForeignKey(Topic, related_name='courses',
on_delete=models.CASCADE)
    name = models.CharField(max_length=200)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    for_everyone = models.BooleanField(default=True)
    description = models.TextField(max_length=300, null=True, blank=True)
```

c. ***Student*** with fields below:

```
class Student(User):
    CITY_CHOICES = [('WS', 'Windsor'),
        ('CG', 'Calgery'),
        ('MR', 'Montreal'),
        ('VC', 'Vancouver')]
    school = models.CharField(max_length=50, null=True, blank=True)
    city = models.CharField(max_length=2, choices=CITY_CHOICES, default='WS')
    interested_in = models.ManyToManyField(Topic)
```

2. Create database tables (make sure *myapp* is included under INSTALLED_APPS in *settings.py*). See what happens after each step.

a. **Tools → Run manage.py Task…** (opens a window where you can type ***manage.py*** commands)
b. In ***manage.py*** window: Type **makemigrations myapp** in dialog box.
c. In ***manage.py*** window: Type **sqlmigrate myapp 0001** #Check latest file in *migrations* dir
d. In ***manage.py*** window: Type **migrate**

3. Update database tables.

a. Add a new model ***Order*** with fields:

- *course* (*ForeignKey(Course)*)
  - indicates the *course* that was ordered
- *Student* (*ForeignKey(Student)*)
  - indicates the *studnt* that ordered the *course*
- *levels* (PositiveIntegerField)
  - indicates how many levels of the *course* were ordered
- *order_status(IntegerField)*
  - choices of valid values = {0,1}. The default value is **1**. The values are interpreted as: `[(0,'Cancelled'), (1, 'Order Confirmed')]`.
  - **HINT**: Use similar format as *city* field in ***Student*** model.
- *order_date*: (*DateField*)
  - indicates the date the *order_status* was last updated

b. Add a new required field *category* to **Topic** model. This indicates the category/class of this topic, e.g., in ***lab4dataset*** file, Development is the category of Web Development Topic.

c. Add a new 'optional' field *description* to **Course** model. This provides a description of the Course. The field should be of type *TextField*.

d. Make the field *school* in **Student** model 'optional'. This field indicates the school (if any) the student is studying at.

e. Set the default value of *city* field in **Student** model to 'Windsor'.

f. Write **__str__** methods for each model.

g. For the **Order** model, write a method `def total_cost(self):` that returns the total cost for all courses in the order.

Run **makemigrations**, **sqlmigrate** and **migrate** again until there are no errors. What is the latest file in *migrations* dir? Open it and check its contents.

PART 2: Enter data into database through Admin interface.

1. Update *admin.py* as follows:

```
from django.contrib import admin
from django.db import models
from .models import Topic, Course, Student, Order

# Register your models here.
admin.site.register(Topic)
admin.site.register(Course)
admin.site.register(Student)
admin.site.register(Order)
```

2. Start your server (**Run → Run 'mysiteS20'**) and navigate to admin site (127.0.0.1:8000/admin).

3. Login using *superuser* name and password (done in Lab #3).

4. Enter the information for each Topic, Course, Student and Order as given in ***lab4dataset*** through the admin interface. How is the data being displayed? Would it be more useful to display additional information? Add other data as you see appropriate.

COMP-8347-91 I/S 2020 LAB #4
Prepared by: Dr. Saja Al Mamoori
Due Date: **Sun Jun 21 (Sec. 57-59), Tues Jun 23 (Sec. 51-56)**

Part 3: Querying the database.

1. **Tools → Python or Debug Console.** In **Python console** import Django then models from *models.py*, then write queries to obtain the following information. Verify if your query generates the correct answer using ***lab4dataset.***

```
import django

from myapp.models import Topic, Course, Student, Order
```

a. List all the *courses* in the db.

b. List all the *students* in the db.

c. List all the *orders* in the db.

2. Write queries to do the following.

a. List all *students* whose last name is 'Jones'

b. List all *courses* that for Topic 'Management'

c. List all *students* that live on 'Sunset Avenue'.

d. List all *students* that live on an 'Avenue' and live in 'Windsor' *city*.

e. List all the *students* that are interested in *Topic* 'Health & Fitness'

f. List the *courses* that cost more than $150.00

g. List the *students* that do NOT live in Windsor.

h. List the *Orders* placed by a *student* whose *first_name* is 'Chris'.

i. List the *courses* that are currently NOT *for_everyone.*

j. Get the first name of the *student* of the *Order* with *pk*=1.

k. List all *topics* that the *studentt* with username 'john' is *interested_in*.

l. List all the *courses* with a *price* < $150 and is *for_everyone*.

m. List the *categories* that the *students* who *ordered* a *Web Dev Bootcamp* is *interested_in*.

n. List the *category* of the *topic* that 'alan' is interested in. (You may assume that 'alan' is interested in exactly one *topic*.)