COMP 8347: Internet Applications and Distributed Systems
SUMMER 2021 LAB #4

PART 1: Edit *models.py* to update existing models and create new models

```
from django.db import models
import datetime
from django.contrib.auth.models import User
from django.utils import timezone
```

1. Add the following model.

a. *Student* with fields below:

```
class Student(User):
    LVL_CHOICES = [
        ('HS', 'High School'),
        ('UG', 'Undergraduate'),
        ('PG', 'Postgraduate'),
        ('ND', 'No Degree'),
    ]

    level = models.CharField(choices=LVL_CHOICES, max_length=2, default='HS')
    address = models.CharField(max_length=300)
    province=models.CharField(max_length=2)
    registered_courses = models.ManyToManyField(Course, blank=True)
    interested_in = models.ManyToManyField(Topic)
```

2. Create new db table. 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. Optional step: Check latest file in *migrations* dir (use **sqlmigrate myapp ****** in *manage.py***)**
d. In ***manage.py*** window: Type **migrate**

3. Update db tables.

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

- *courses* (*ManyToManyField (Course)*)
  - indicates the *course* that was ordered
- *Student* (*ForeignKey(Student)*)
  - indicates the *student* that ordered the *course*
- *order_status(IntegerField)*
  - choices of valid values = {0,1,2}. The default value is **1**. The values are interpreted as: `[(0,'Cancelled'), (1, 'Confirmed'), (2, 'On Hold')]`.
  - **HINT**: Use similar format as *level* field in ***Student*** model. But field type will be different.
- *order_date*: (*DateField*)
  - indicates the date the *order_status* was last updated. The default value is current date (i.e. timezone.now).

b. Add a new required field *length* to **Topic** model, with a default value of 12. This indicates the *length* of the courses for this *topic* in number of weeks.

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 *address* in **Student** model 'optional'. This field indicates the physical *address* of the *student*.

e. Set the default value of *province* field in **Student** model to 'ON'.

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 through Admin interface

1. Update *admin.py* as follows:

```
from django.contrib import admin
  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 'mysiteF20'**) and navigate to admin site (127.0.0.1:8000/admin).

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

4. Enter/update the information for each *Topic*, *Course*, *Student*, and *Order* as given in ***lab4dataF20.txt*** through the admin interface. **Note**: Some data was already entered in Lab 3.

How is the data being displayed? Would it be more useful to display additional information?

5. a. Write **__str__** methods for each model.

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

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 *lab4dataF20.txt*.

```
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 *topics* whose course length is 8 weeks

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

d. List all *students* that live on an 'Avenue' and live in *province* 'ON'.

e. List all the *students* that are interested in *Topic* 'Sports'

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

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

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 *student* with username 'john' is *interested_in*.

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

m. List the *Topics* that the *student* who *ordered* a *Web Dev Bootcamp* is *interested_in*. Assume there is exactly one order for *Web Dev Bootcamp* course.

n. Find the *length* of the courses for the *topic* that 'chris' is interested in. (You may assume that 'alan' is interested in exactly one *topic*.)

o. Find the number of courses that 'chris' is registered in.