 Due before the 25^{th} March, by 2:30pm

CMPSC 312
Database Systems
Spring 2019

Lab 5 Assignment:
The Django *Film* App for your Film Collection
Submit deliverables through your assignment GitHib repository.
Place working project, source-code and markdown material in the `djangoProject/`, `src/` and `writing/` directories, respectively

Objectives

To learn how to build a Django webserver project for film data. Your web server will host information about your favourite picks titles of Hollywood-type movies. This data will be inserted by Python code that uses the `python manage.py shell` utility.

GitHub Starter Link for Groups

STOP! STOP!
Not everyone will be clicking this link at this time!
Only the team leader will be clicking the link to create the repository!!
`https://classroom.github.com/g/f0lnwf24`

Creating your repository

We will use a group assignment functionality of GitHub Classroom for this assignment. For group assignments **only one person will be creating the team while the other team member will join that team.** Please form a team of **no more than two people** and select one person to create the repository.

The selected person of the team should go into the link to the lab in the assignment sheet. Copy this link and paste it into your web browser. Now, you should accept the laboratory assignment and create a new team with a unique and descriptive team name (under "Or Create a new team").

Now the other members of the team can click on the assignment link and select their team from the list under "Join an Existing Team". When other team members join their group in GitHub Classroom, a team is created in our GitHub organization. Every team member will be able to push and pull to their teams repository.

To use this link, please follow the steps below.

- Click on the link and accept the assignment
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab,
- Clone this repository (bearing your name) and work locally

- As you are working on your lab, you are to commit and push regularly. The commands are the following.
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

Introduction

For the last two weeks, we have been using Python to interact with SQL code and databases. In this lab, you will be mixing Python and SQL code and databases again. This time, Django, the Python-based web server platform, *Django* (available from `https://www.djangoproject.com/`), will handle the database management for you. You will be controlling this web server software, using Python code that is very similar to that which was also discussed during class.

During class, we learned how to set up the apps, configure settings, and connect to its automatically-produced databases. In addition, we studied how to view database output using a browser using Django's web-server functionality. From this practice, you should have a working copy of a functional server project: your own copy and that from the instructor that will likely be helpful to you as you work on this lab.

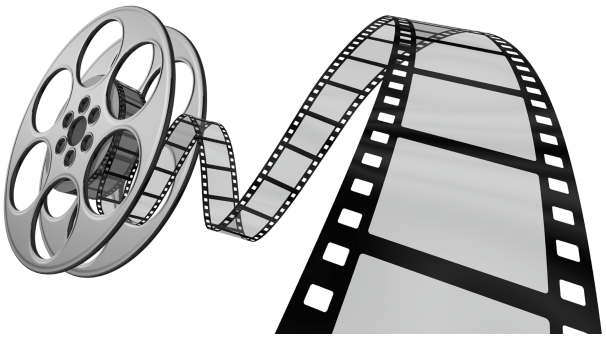

Figure 1: Build an app in Django to contain information about Hollywood productions.

Main Ideas

In this lab, you will build an app in Django to contain a catalog of information about Hollywood-type films in the same way we were able to create an app to contain a catalog of music. You are to use your abstract Python programming skills to populate the film App's database. Note: You are to use abstract programming to insert at least five (5) Hollywood-type films into your database.

Inserting Data with the `shell`

To populate the database, you will use the `python manage.py shell` and paste Python code that will populate the film table(s). The code to populate your database will look similar to the following.

```
# above code is all on one line
a = Album(artist = "The Bill-Browns",
album_title = "Ready or not",
genre="Rock",
album_logo = "Find-Link-Of-Image-Online")
# Now save your object and Django will populate the Database!
a.save()
# show the ID (primary key)
a.id
```

You have done all this before!

Please return to your notes and code from class to help you with this project (i.e., building the Django project and then adding the data using the `python manage.py shell` command. The end deliverable of this project will be a working Django web server and page system (containing links) to display Hollywood-type film information and a film logo in a browser.

Your final (`video`) project will resemble the `music` app that we studied in class and you are encouraged to follow the classes example as you build your own server for at least five films of your choice.

Group Work

If you prefer, this lab may be completed in groups of no more than two (2) people. If you decide to work with a partner, please *clearly* indicate this person's name along with your own name in your completed work. Each person is to share resources and work out of the same GitHub repository to work equally on the deliverables which are described at the end of this assignment sheet.

Creating Your Film App and Database

First Steps: Creating the App

In class, as outlined in your slides, we setup an app in Django called *music*. When this app was created, a database called `db.sqlite3` was automatically created by Django and was found in the root directory of the project. Create your Django project similar to how we created our projects in class. Then create a *film* app using the same steps as those from class.

At the completion of our coding, we were able to view this SQL database using `sqlite3` and the admin browser. Furthermore, we noted that our music data was stored in a table called, *music_album*. To construct your *film* app, please follow the same steps as those followed in class to create the *music* app. Remember your app will be called "`film`" (i.e., **not** "`music`.") As before, the *film* database will be created automatically when you create the new app.

Please be careful when copying and pasting code from the slides into your project: the variables and Python formatting may have changed. Make sure check for errors from copying and pasting from slides.

Attributes and Schema

Be sure to use the four attributes (listed below) to contain your data for Hollywood-type films in the app's database. Note: these attributes may be set and edited in the configuration file, `mysite/film/models.py`.

- The main class will be called, *album*.
- `mainActor`, to contain the name of a leading actor of the film
- `filmTitle`, to contain the film's title
- `director`, to contain the name of the director
- `genre`, to contain the type of film (i.e., drama, comedy, horror, etc).

Inserting Data: Abstract Programming

Once you are sure that your Django web server is correctly working with its (connected) film database, you are to use Python commands (by source code) from the `python manage shell` command. For this, you may copy and paste the code to populate your database. Please save this Python code in a safe place as a file because it will included as one of your deliverables.

1 Summary of the Required Deliverables

This assignment invites you to submit an electronic version of the following deliverables through your group GitHub assignment repository. Do not forget to add your names to your work.

There are three deliverables for this project, listed below.

1. **Report**: Update the given Markdown file `writing/report.md` to include you include convincing evidence that your film database for Django works, including screen shots of the records in the browser-view of your film database. In this report, please discuss any difficulties (coding, database support, etc) that you encountered and the solutions that you used to resolve them. The instructor will be using this document to determine the working-nature of your Django web server.
    - **Names of your group-partner(s)**: If you opted for working in a group, please let the instructor know who your partner was so that points can be awarded correctly. Add your names to the report document.
2. **Python Code**: Update the given Python code in the file `src/forShell.py` to create code that that can be pasted into the `python manage.py. shell` utility. This is how you will be automating the population of your web site. An example of the code for the `music` app is given above.
3. **Django Project**: Your entire (working) Django project, `mysite`. Copy your project into a directory called `djangoProject/` that you will need to create in your assignment repository.

Please see the instructor if you have any questions about this assignment.