Homework 5: More Social Interaction

Due date: ~~February 22, 2016~~ => February 23, 2016, last late day still February 26, 2016

This homework is the third in a series of homeworks in which you will build an increasingly sophisticated nano-blogging site. This site will eventually be a featureful, interactive web application including user registration and authentication, email integration for user verification, photo upload, and quasi-real-time updates.

For this assignment, you will extend your Homework 4 solution to use Ajax to provide comments on posts in your social network site. You will also use Ajax refresh the global stream to show new posts.

The learning goals for this assignment are to:

- Demonstrate mastery of the learning goals from the previous homeworks, including both technical features of web applications and the development process.

- Gain familiarity with client-side web programming, including the use of JavaScript (and optionally—but hopefully—JavaScript libraries) for DOM manipulation and Ajax.

Specification

This section describes the enhancements you will make to your social network application. To begin your assignment, recall that you should start by copying over your project from the last homework over to the hw5 folder (see Turning in your work for the directory structure).

1. The global stream updates with new posts, without refreshing the HTML page, every 5 seconds.

2. Logged-in users are able to add comments to posts anywhere posts are shown.
    - Comments show the author, the profile image, and the time the comment was made.
    - Comments displayed in chronological order (newest on bottom) per post.
    - It should be clear in the display of the social network design which comments belong to which posts.
    - Comments are added to the posts using Ajax (i.e. the insertion of the new comments take place without refreshing the entire web page).

Requirements

Your application must also follow these requirements:

- You must meet all requirements specified in the previous assignment, including in particular:
    - The empty URL (i.e. http://localhost:8000/) must route to the first page of your application.
    - Your application should not use any hard-coded absolute paths.
    - Your application should run with **Django 1.8** or **Django 1.9**.
    - If you are using **Python 3** please note this in a `README.md` file in the hw5 directory.
    - Your application should use the default Django database configuration based on a SQLite database file (named `db.sqlite3`) in your project directory.
    - Your application should not crash as a result of any input sent to the server-side or because of any actions the user performs.
    - Your application should use template inheritance, reverse URL resolution, as well as complete validation of client-submitted data with Django Forms.
    - All tasks and features in the specification must be easily accomplishable using the user interface for your social network.

- You must fix all errors that the mentioned in your hw3 feedback (grades for hw3 will be posted well before this assignment is due).

- When implementing the stream refresh, your application must only send and load new posts since the last update. Do not generate and reload the entire stream as was demonstrated in the to-do list example.
    - You do not need to check for new comments on existing posts; you only need to check for new posts. (This is an excellent optional feature, though!)

- Cite all external resources used and any additional notes you would like to convey to your grader in the README.md file.

Implementation hints

1. It is a *lot* easier to implement Ajax calls using jQuery rather than plain JavaScript. Consider this a strong suggestion from the course staff.
    - http://learn.jquery.com/ajax/jquery-ajax-methods/

2. Recall that you must be able to comment on all posts—including the posts newly added by the stream refresh. This means that you must make sure to have event handlers attached to the newly-created DOM elements, which is annoying if you are dynamically attaching event handlers. One simple technique that is commonly used throughout front-end development to combat this problem is called **event delegation**. You can read about this technique here:
    - http://learn.jquery.com/events/event-delegation/

Grading criteria

For substantial credit your solution must clearly demonstrate the learning goals for this assignment, which are described above in the introduction.

Committing your work

As with the previous homeworks, we will be evaluating your version control usage. Keep in mind that good version control usage typically means (1) incremental, modular commits with (2) descriptive and useful commit messages.

Specification fulfillment

Your submission must follow all specifications and requirements introduced in the previous sections.

Validation

As with previous assignments, any client request (achievable or not by your user interface) must not be able to crash the application. Additionally, we are *still* looking to see if you correctly use Django Forms for input validation.

Make sure that all Ajax requests are properly validated.

Coverage of technologies

You must demonstrate pro use of the technologies of this assignment, particularly:

- basic JavaScript (DOM manipulation, event handling)
- Ajax requests

Design

As with previous assignments, we will provide comments and suggestions on your user interface, but will not be grading specifically the aesthetics of your project.

Turning in your work

Your submission should be turned in via Git and should consist of a Django application in the hw5 directory. Name your project **webapps** and the application **socialnetwork**. The directory structure will look somewhat like the following (some files/directories omitted):

```
[YOUR-ANDREW-ID]/hw5/
   |-- webapps/
       |-- settings.py
       |-- urls.py
       |-- [etc.]
   |-- socialnetwork/
       |-- static/
       |-- templates/
       |-- models.py
       |-- views.py
       |-- [etc.]
   |-- manage.py
   |-- db.sqlite3
   |-- README.md
```