Objectives

Building on what you’ve learned with class based views, user registration, multi-user sites, and ajax, you’ll be building a Pinterest clone. In fact, the Pinterest site is developed in Django. If you have been successful with the Time Clock assignment, this assignment should be easier to manage.

Requirements

- User Stories:
  - A user can join the website and create his or her own pin board
  - A pin is made up of a photo, link to a site (optional), private / public designation, date posted, and a description
  - Pins can be put into multiple groups called pinboards. Pinboards have a name and an owner.
  - All public pins are available to outsiders
  - A user can re-pin another user’s pin
  - When viewing pins, you can see the user’s Gravatar (profile picture)
  - The home page contains a list of the latest public pins
  - Logged in users can leave a comment on a pin
  - A user should be able to edit and delete his or her own pins
  - Scrolling down on a long page should automatically load more pins (infinite scrolling)

- Tests
  - You can’t edit, delete, etc. someone else’s pins

Extra Credit Ideas

- Have image uploads go to and serve from an Amazon S3 account by using the S3 storage backend
- List of users allowed to view a pin board

Hints

- Use [http://github.com/nvie/django-gravatar](http://github.com/nvie/django-gravatar) to generate the avatar picture for the user