Classified Ad Web Site - Milestone 4

In this assignment, you will expand your classified ads web site to add search and tags functionality equivalent to:

https://chucklist.dj4e.com/m4

Important: The number of lines you need to add to your code is *relatively* small. Take your time read the sample code carefully - only make changes that you understand. Wholesale cutting and pasting sample code will make it almost impossible to complete this assignment.

This is somewhat like the kind of real work you do when you have a working application and want to add a feature the the application. First - don't break what you have working.

Adding Search

The `well` sample application contains code you can adapt to implement the code to search the title and text:

https://samples.dj4e.com/well/

To avoid getting too much broken at one time - it is probably a good idea to make search work and then further evolve your code to support tags.

Adding Support for Tags

The `tagme` application adds a `tags` field to the model and and adds support for tags to the user interface and search code.

https://samples.dj4e.com/tagme/

You should also review the documentation for the `django-taggit` library at:

https://django-taggit.readthedocs.io/en/latest/

You might find the easiest path is to use the `taggit` documentation to make your changes, looking at the `tagme` code to verify what you are doing.

There is a bit of an extra wrinkle when adapting the approach in `tagme` because we are using a ModelForm in order to process uploaded pictures. The key is that you have to save the tags after the form has been copied to the model and the model has been saved because the tags are stored using a many-to-many data model.

https://django-taggit.readthedocs.io/en/latest/forms.html#commit-false

In your `forms.py` code you will need to (a) add 'tags' to the field list and (b) update the code in the commit to look like:

```
if commit:
    instance.save()
    self.save_m2m()    # Add this
```

In your `views.py` , we have our own code to pull data from the form to the model and then save the model. This code is in both the insert and edit views:

```
# Adjust the model owner before saving
inst = form.save(commit=False)
inst.owner = self.request.user
inst.save()

# https://django-taggit.readthedocs.io/en/latest/forms.html#commit-false
form.save_m2m()    # Add this
```

You need to add the `save_m2m()` call *after* the instance was saved.

Finally, the detail template for the `tagme` application contains code that can be adapted to display the tags in your application.

Manual Testing

It is always a good idea to manually test your application before submitting it for grading. Here are a set of manual test steps:

- Log in to your application and create several ads add a tag to one of the ads. Make sure that you put unique words that are only in the title, text, and tag so you can verify that you can search for a word in any of the three places.
- Go into the detail page for an ad and verify that it shows you the title, text, and tag(s) that you entered
- Make sure to edit an ad and verify that you are able to change the title, text, and tags.
- Search for some text that is only in a title and verify that the correct ads come back.
- Search for some text that is only in a body and verify that the correct ads come back.
- Search for some text that is only in a tag and verify that the correct ads come back.
- Note the "?search=" in the location bar in your browser while you are doing searched
- Clear the search and see all of the results and verify there is no "?search=" get parameter