

End of Class Assessment

- Tests what you've learned
- Exercise many parts of Django
- Use all or part of Friday afternoon

This end of class project is to see what you've learned in the Django class. It is similar to the chapter exercises, but is more comprehensive. Unlike those, it will be evaluated by the instructor.

It is an open-book project. I.e., you can use the Django documentation, the course notes, class examples, and the Internet as you complete the project.

Do not collaborate with other students.

Deadline for turning in the project on Friday is

5 PM CET (Germany)

4 PM GMT (England)

6 AM HST (Hawaii)

When you are ready, zip up your project and put it in the **submit** folder on the remote desktop. If you are not using the remote desktop, or you want to be extra sure it gets turned in, you can email it to the instructor: jstrickler@gmail.com

On the remote desktop you can create a zip file as follows:

```
zip -r lastname_django_project wantads
```

This will create a file named *lastname_django_project.zip*

NOTE

Be sure to put your last name in the zip file name, something like *lastname_django_project.zip*. If the instructor can't determine your name, you won't get any credit and will fail the course, and we don't want that. It's not a bad idea to put a readme in the top level of your project with your name and email, just in case.

Project Overview

- Create a want ads site
- Manage items for sale
- Provide search feature

Create a web site for want ads. Think of it as a mini-Craig's List (very mini).

Name the project **wantads** and make an app named **ads**. (Or, make up your own names!)

Register your models with the admin tool.

Create some ads using your app (not via the Django shell or the Admin interface).

OPTIONAL: Setup and manage user logins

Project Architecture

The app will have four pages (views):

- Main page with search blank
- Result page with list of ads
- Detail page for one ad (will link to main page)
- Entry page for adding new ads (will link back to main page)

Main Page

- Search blank and a search button
- Search titles of ads only
 - Search can be partial (i.e., title contains any part of search term)
 - Empty search lists all ads
- Link to page for adding new ad

Results page

- List of just the titles of all matching ads, linked to their detail pages
- Link to return to main page

Detail page

- All details for one ad
- Link to return to main page

Create page

- Form with blanks for all ad fields
- SAVE button on form for saving ad — returns to main page
- Link to return to main page

HINT: Use a `ModelForm` to create the new add. It will handle the categories automagically.

HINT: When saving an ad, check whether the category exists. If it does not, add it before adding the ad itself.

Models

Ad

- Title
- Price
- Date added
- Email
- Phone #
- Category (foreign key to **Category** model)

Category

- Name
 - Unique constraint

NOTE

For this assessment, let users make up their own categories. In real life, we would probably pre-populate the categories table.

Final Note

The app does not have to be beautiful or enterprise-ready. What we're looking for is that you learned how to create a simple Django project and app, that you can create page views, that you know how to define models, and that you can set up forms and extract the filled-in data from them. Good luck!