MIT AITI
Python Software Development
Lab DJ1:

This lab will help you get Django installed and write your first application.[1]

Each person in your group must complete this lab and have it checked off. Make sure that you are running Ubuntu. All file editing will be done in gedit.

1. Download Django. Follow option 1 here: https://www.djangoproject.com/download/
    a. Note that you have to `cd` (change directories in a terminal) into your directory of choice (probably ~/Downlaods) before running the three listed commands:

```
tar xzvf Django-1.4.tar.gz
cd Django-1.4
sudo python setup.py install
```

2. Navigate to your Desktop

```
$ cd ~/Desktop
```

3. Create a new project called "webnotes":

```
$ django-admin.py startproject webnotes
```

4. Change to the `webnotes` directory and look at the list of files. You should see a folder called webnotes and a file called manage.py in the directory.

```
$ cd webnotes
$ ls
```

5. Make a new directory called `templates`

```
$ mkdir templates
```

6. We are now going to make a base template for our site. Create an html file called "`base.html`" inside the `templates` folder, and then make a `notes` directory.

```
$ cd templates
$ mkdir notes
$ gedit base.html
```

7. Place the following text into base.html

```
<html>
    <head>
        <meta http-equiv="Content-type" content="text/html;
charset=utf-8" />
        <title>Webnotes</title>
```

[1] This lab was adapted from: http://komunitasweb.com/2010/02/django-tutorial-simple-notes-application/.

```
    </head>
    <body>
        <div id="content">
       {% block content %}
       {% endblock %}
       </div>
    </body>
</html>
```

8. Change into the `notes` directory and add two files: `detail.html` and `list.html`. These files will extend the `base.html` file that you have created, by overriding the `content` section.

```
$ cd notes
$ gedit detail.html
   {% extends 'base.html' %}
   {% block content %}
       <h2>{{ note.title }}</h2>
       <p>{{ note.content }}</p>
   {% endblock %}

$ gedit list.html
   {% extends 'base.html' %}
   {% block content %}
       {% if note_list %}
       <ul>
       {% for note in note_list %}
           <li><a href="{% url notes_detail note.id
   %}">{{ note.title }}</a></li>
       {% endfor %}
       </ul>
       {% else %}
       You don't have webnotes
       {% endif %}
   {% endblock %}
```

9. Edit the `settings.py` file so that django knows where to look for your templates:

```
$ cd ~/Desktop/webnotes/webnotes/
$ gedit settings.py
```

   a. At the top of the file, add these two lines:

```
import os # put this at the top of the file
PROJECT_ROOT = os.path.realpath(os.path.dirname(__file__))
```

   b. find the line that starts with “TEMPLATE_DIRS” and replace it with:

```
TEMPLATE_DIRS = (
    os.path.join(PROJECT_ROOT, '../templates')
)
```

10. Keep `settings.py` open and edit the database configuration, by specifying sqlite3. Find the following two variables and edit them to have these values:

    ```
    ‘ENGINE’: 'django.db.backends.sqlite3',
    ‘NAME’: 'webnotes.db',
    ```

11. Now start (create) a new application inside this webnotes project:

    ```
    $ cd ~/Desktop/webnotes/
    $ django-admin.py startapp notes
    ```

12. The application will be placed in a new folder called “`notes`”. Navigate into it and look at the list of files.

    ```
    $ ls
    $ cd notes
    $ ls
    ```

13. Open `models.py` and add a `Note` model

    ```
    $ gedit models.py
    ```

    ```
    from django.db import models
    class Note(models.Model):
        title   = models.CharField(max_length=255)
        content = models.TextField()
        def __unicode__(self):
              return self.title
    ```

14. Open `views.py` and add a list and detail view. These views define what data will be displayed and how it will be displayed.

    ```
    $ gedit views.py
    ```

    ```
    from django.template import Context, loader
    from django.http import HttpResponse
    from models import Note

    def notes_list(request):
        note_list = Note.objects.all()
        t = loader.get_template('notes/list.html')
        c = Context({
            'note_list': note_list,
        })
        return HttpResponse(t.render(c))

    def notes_detail(request, id):
        note = Note.objects.get(pk=id)
        t = loader.get_template('notes/detail.html')
        c = Context({
            'note': note,
        })
        return HttpResponse(t.render(c))
    ```

15. Create a new file called “`urls.py`” and define some URLs that will access your views:

```
$ gedit urls.py

   from django.conf.urls.defaults import *
   import views

   urlpatterns = patterns('',
       url(r'^list/$', views.notes_list, name='notes_list'),
       url(r'^detail/(?P<id>\d+)/$', views.notes_detail,
   name='notes_detail'),
   )
```

16. Create a new file called `admin.py` and register the `notes` application with the built-in admin application

```
$ gedit admin.py

   from notes.models import Note
   from django.contrib import admin

   admin.site.register(Note)
```

17. Navigate up one directory to get back to your project home and enter the webnotes directory. Open the `settings.py` file and go to the INSTALLED_APPS section. Add ‘`notes`’ and uncomment the `django.contrib.admin` app.

```
$ cd ../webnotes/
$ gedit settings.py

   INSTALLED_APPS = (
       ...
       'notes',
       'django.contrib.admin',
   )
```

18. Open the `urls.py` for your project (not the one for the notes app). Tell your project `urls.py` that it should include the URLs from your notes app. Uncomment the reference to the admin urls.

```
$ gedit urls.py
   ...
   from django.contrib import admin
   admin.autodiscover()
   ...
   urlpatterns = patterns('',
        url(r'^notes/', include('notes.urls')),
        ...
        url(r'^admin/', include(admin.site.urls)),
   ...
```

19. Tell django to build a database based on your models.

    ```
    $ cd ..
    $ python manage.py syncdb
    ```

    When asked to make a superuser, say “yes” and enter a username, email, and password. DO NOT FORGET THIS.

20. Run the development server

    ```
    $ python manage.py runserver
    ```

21. visit http://localhost:8000/admin, and add a note

22. visit http://localhost:8000/notes/list and see if your note is there

23. Once all of this is done, it's time to make a change. Add an author to the notes application and output it in the detail page. Here's how to do it:
    a. Open models.py

       ```
       $ gedit ~/Desktop/webnotes/notes/models.py
       ```

    b. Add an `author` field.

       Hint: it's very similar to the `title` field.

    c. Edit the details template

       ```
       $ gedit ~/Desktop/webnotes/templates/notes/detail.html
       ```

    d. Output the author after the title in an `<h3>` tag
    e. Go to the terminal where you ran `python manage.py runserver`, press `ctrl+c` to stop it
    f. Delete all previous data so that you have a clean database.

       ```
       $ python manage.py reset notes
       ```

    g. Synchronize the database

       ```
       $ python manage.py syncdb
       ```

    h. Start the server again

       ```
       $ python manage.py runserver
       ```

    i. If your change worked, you should see an author field when adding notes in the admin interface