Django

Build an organized web application

Presented by Youssef BADDI

Department of Computer Science, EST SB, Chouaib Doukkali University

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. Tutorial
5. Conclusion

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. Tutorial
5. Conclusion

Introduction

What is Django?

A high-level Python Web Framework

Introduction

What is Framework?

Front-end & Back-end co-development

Introduction

What is Framework like?

A stage company for performers

Introduction

Why should I use Django?

Clean & Rapid development

Introduction

Who use Django?

Instagram & Pinterest & Bitbucket

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. Tutorial
5. Conclusion

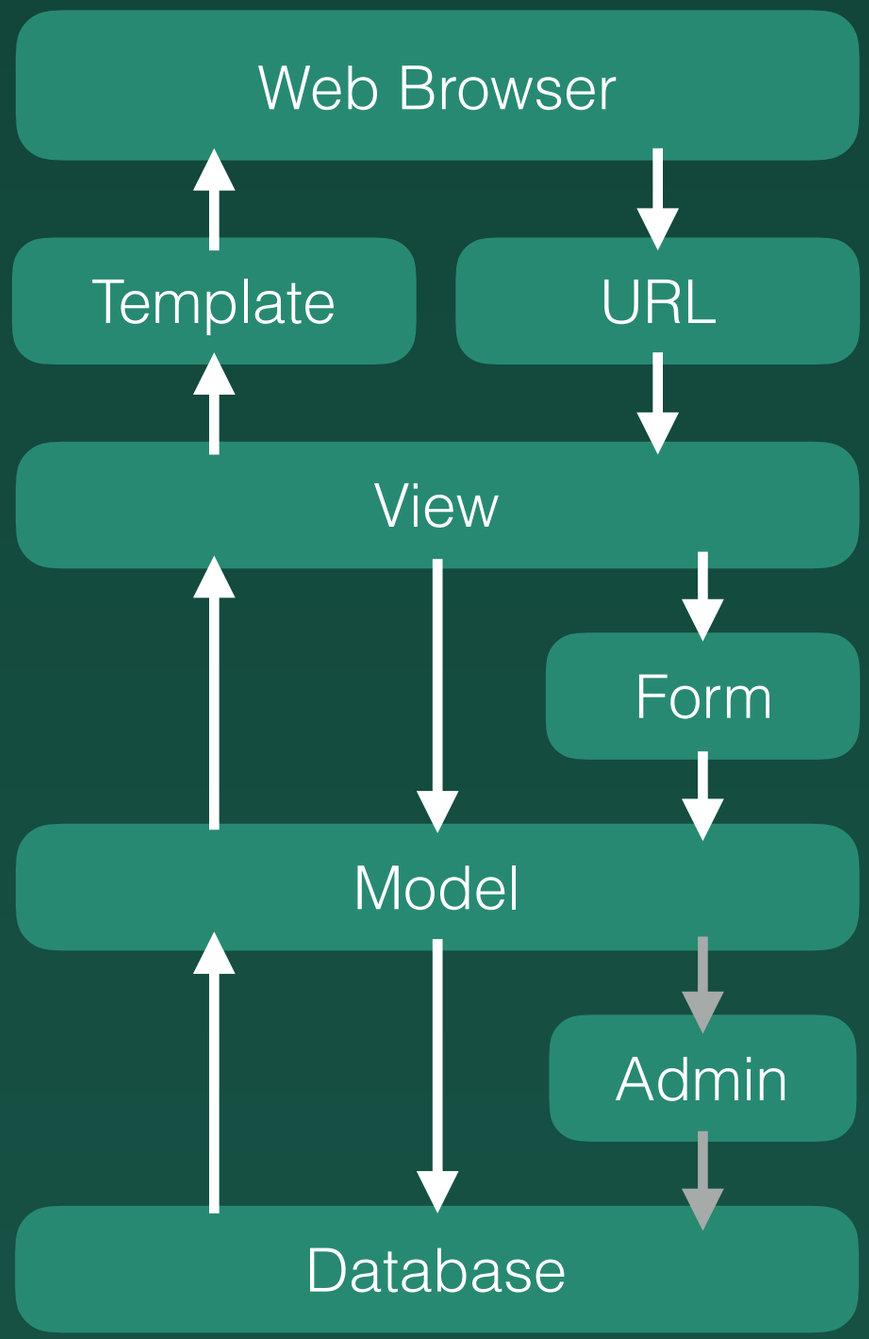
Outline

1. Introduction
2. **Basic Knowledge**
3. Technical Details
4. Tutorial
5. Conclusion

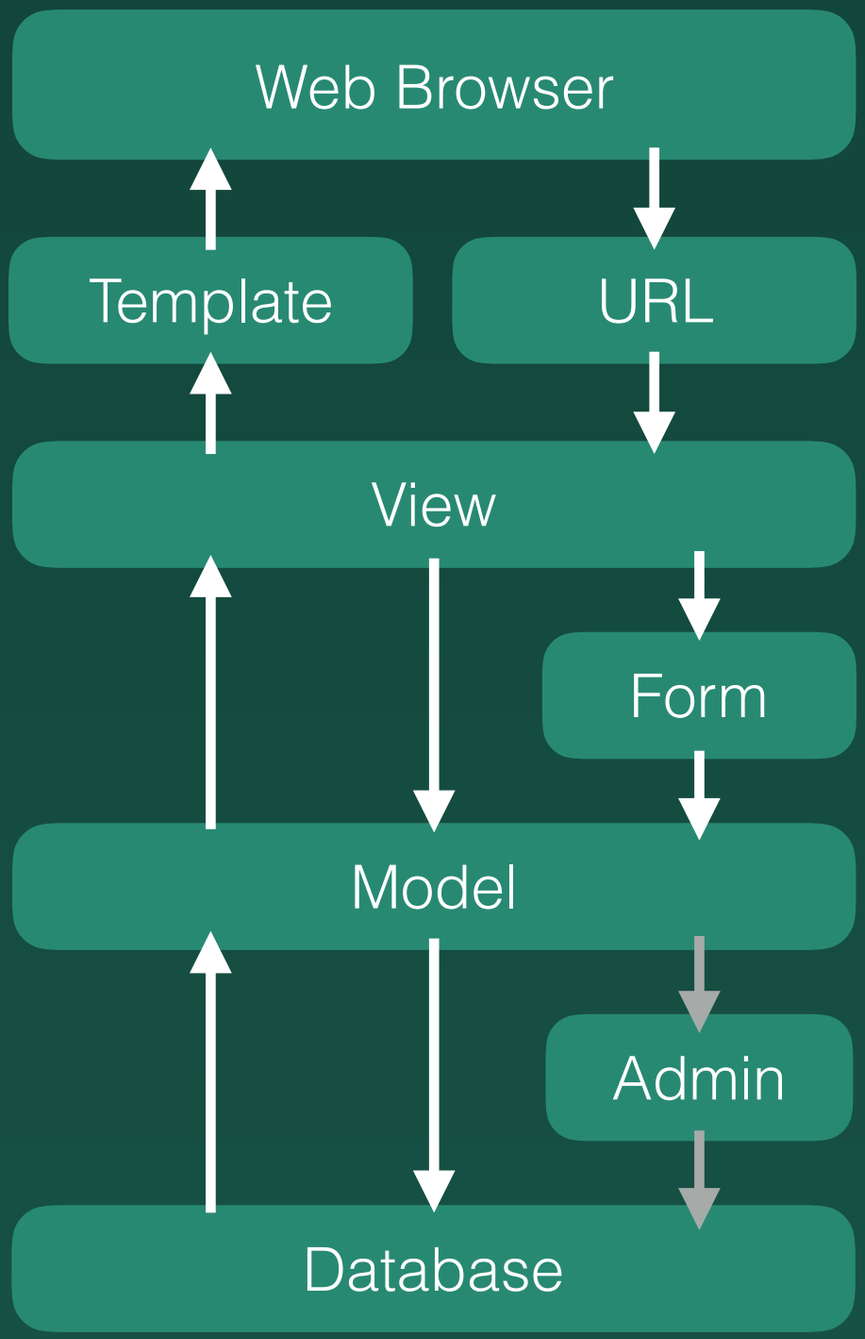
Basic Knowledge

Basic Knowledge

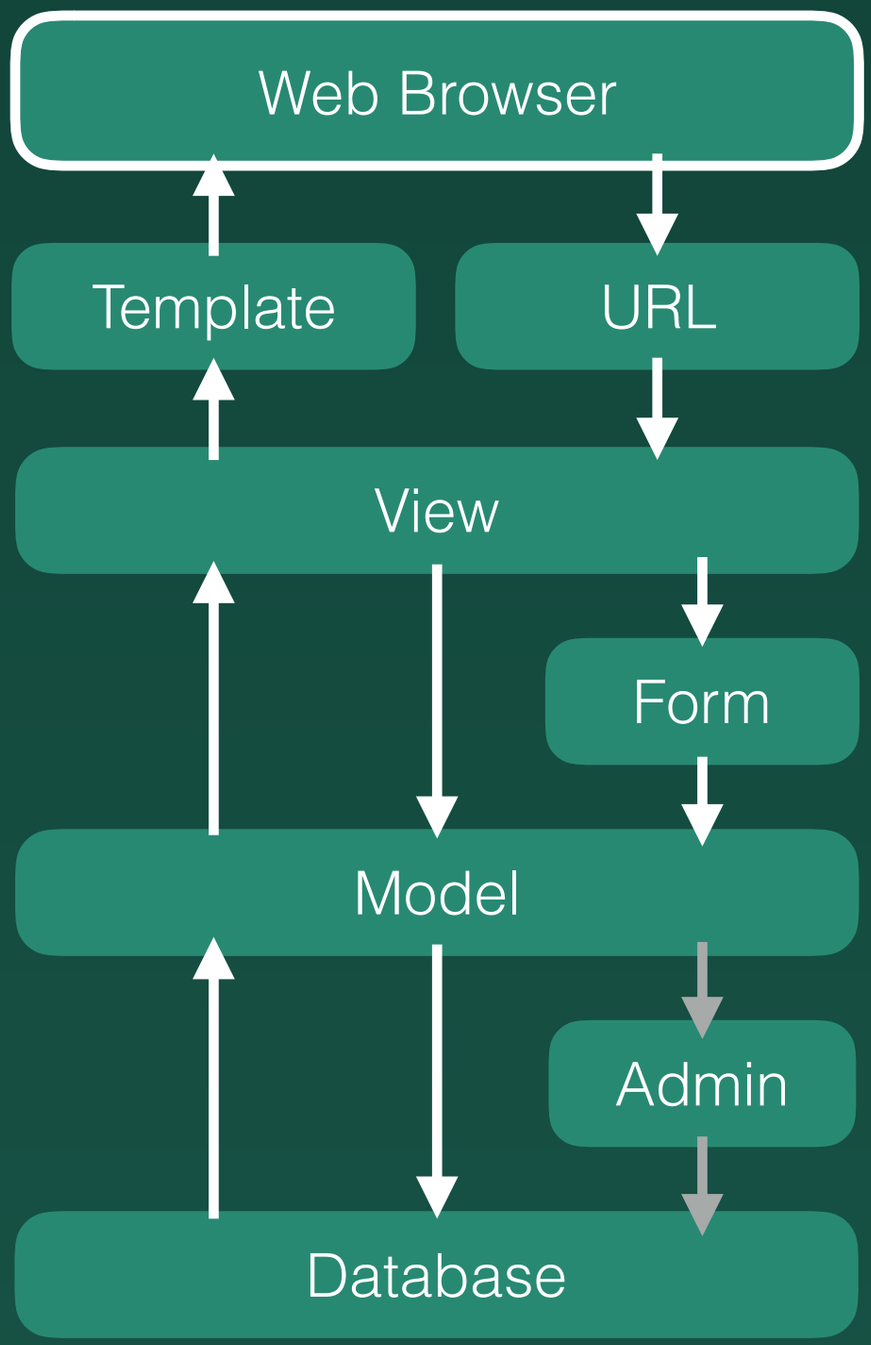
Workflow



Basic Knowledge



Basic Knowledge



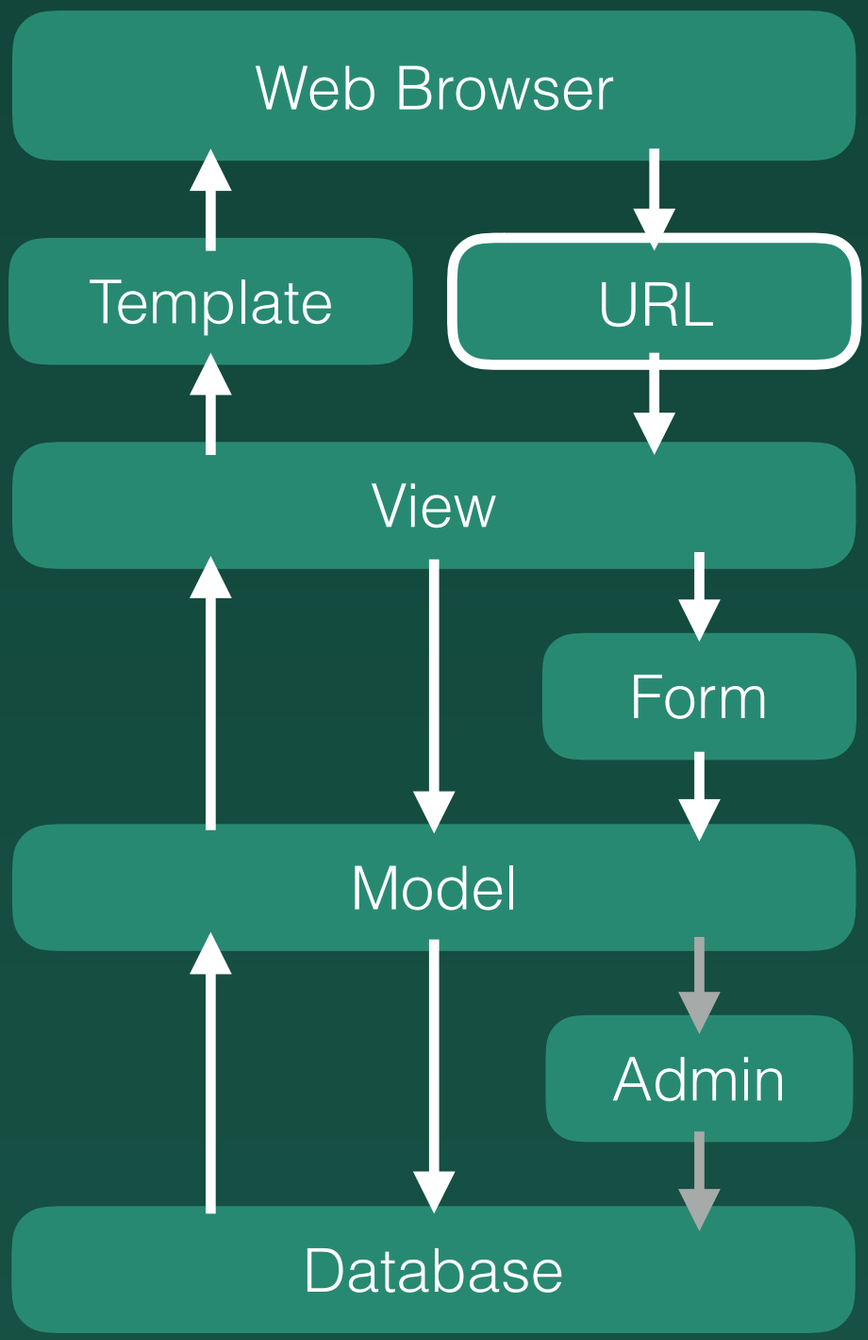
Web Browser

- What users actually see
- Responds to what users do

when they

1. Click
2. Type
3. Press Enter

Basic Knowledge



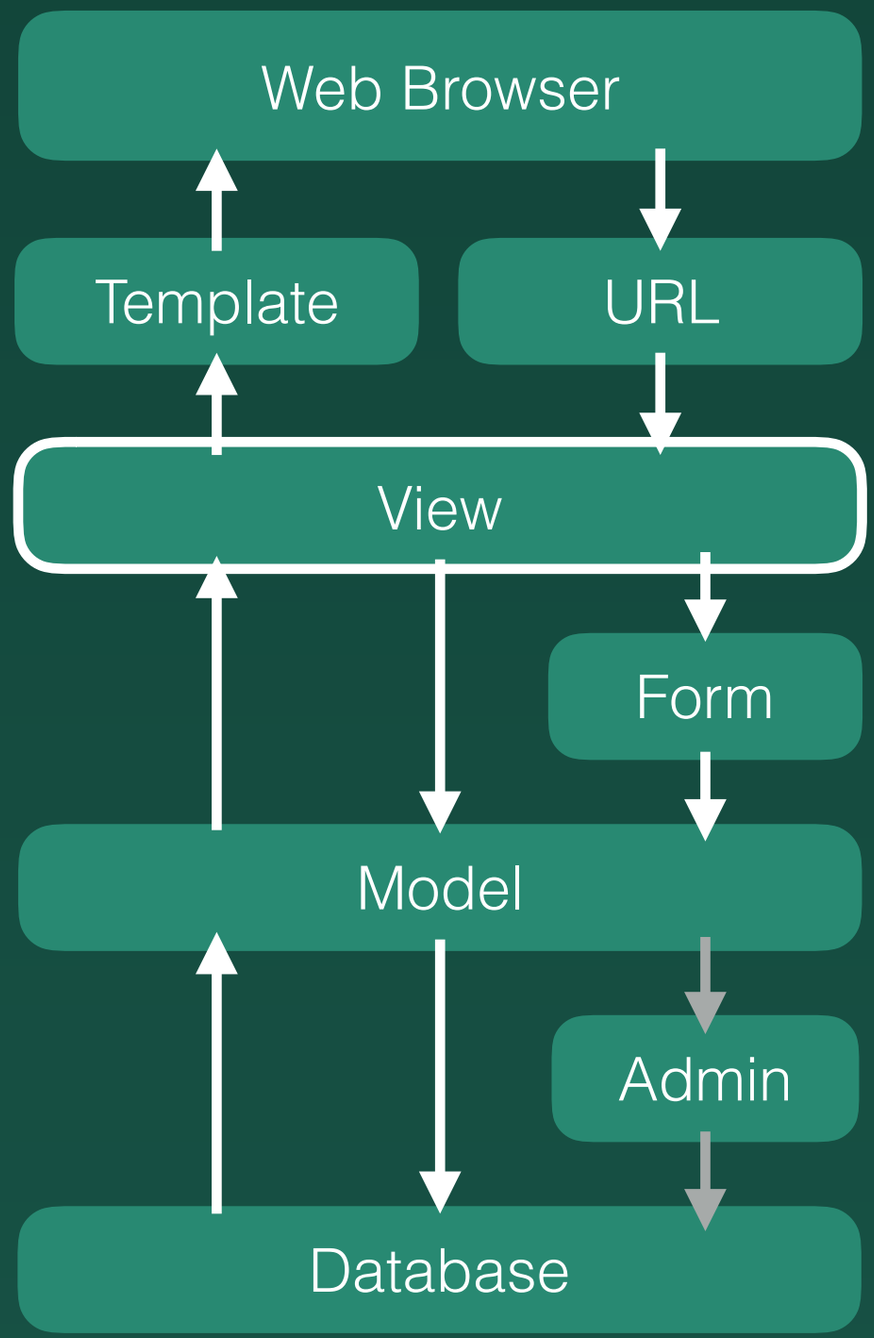
URL

- Often referred as the ‘web address’
- Provides **mapping** to View

What is mapping?

The act of assigning **functions** to **URLs**

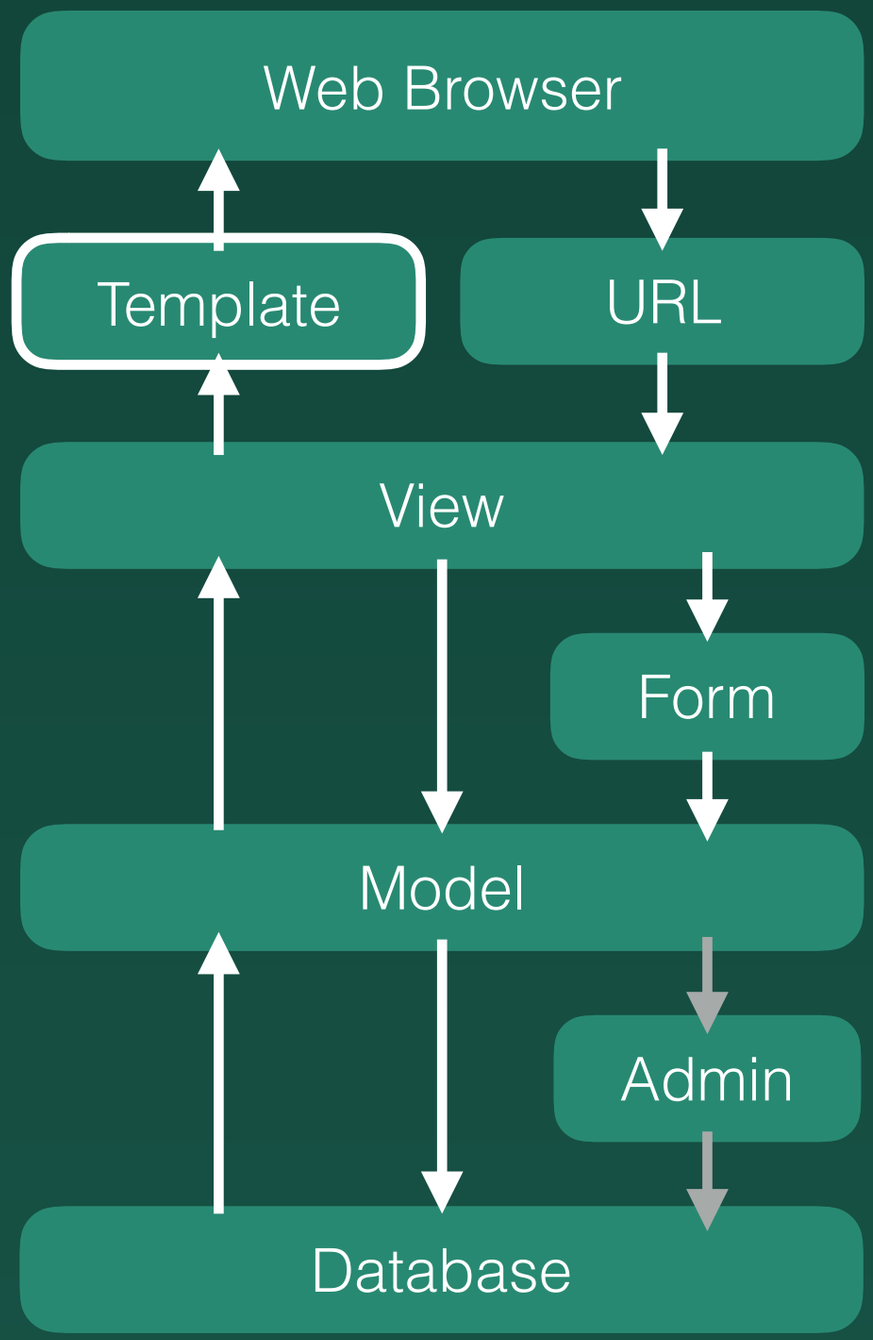
Basic Knowledge



View

- Where all the functions are written
- Render content to **Template**
- Get information from **Model** before rendering content
- Put information into **Model** and into **Database** through **Form**

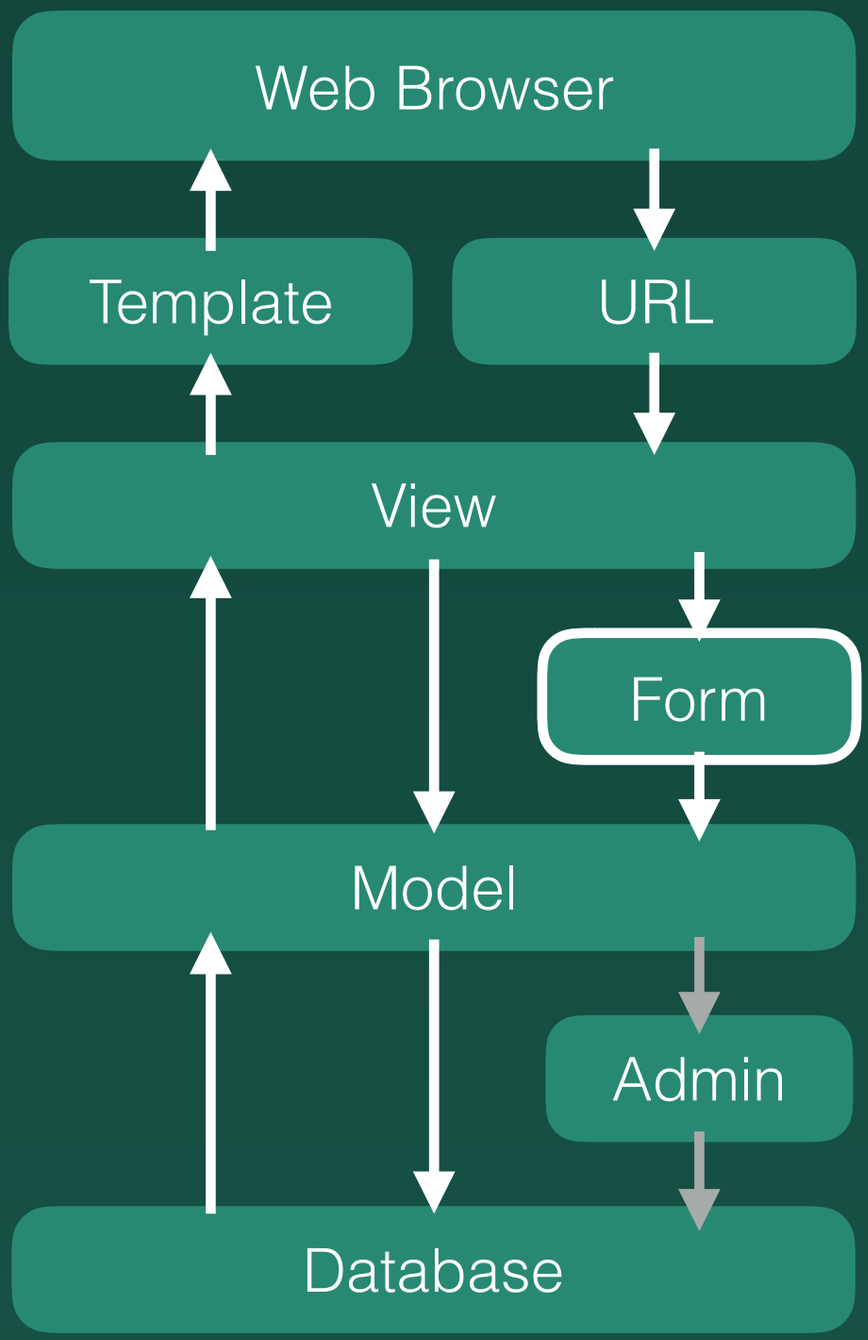
Basic Knowledge



Template

- The place you systematically store all of your **Html** files
- You will have a ‘static’ folder to store other **CSS** files, **Javascript** files, or Images

Basic Knowledge



Form

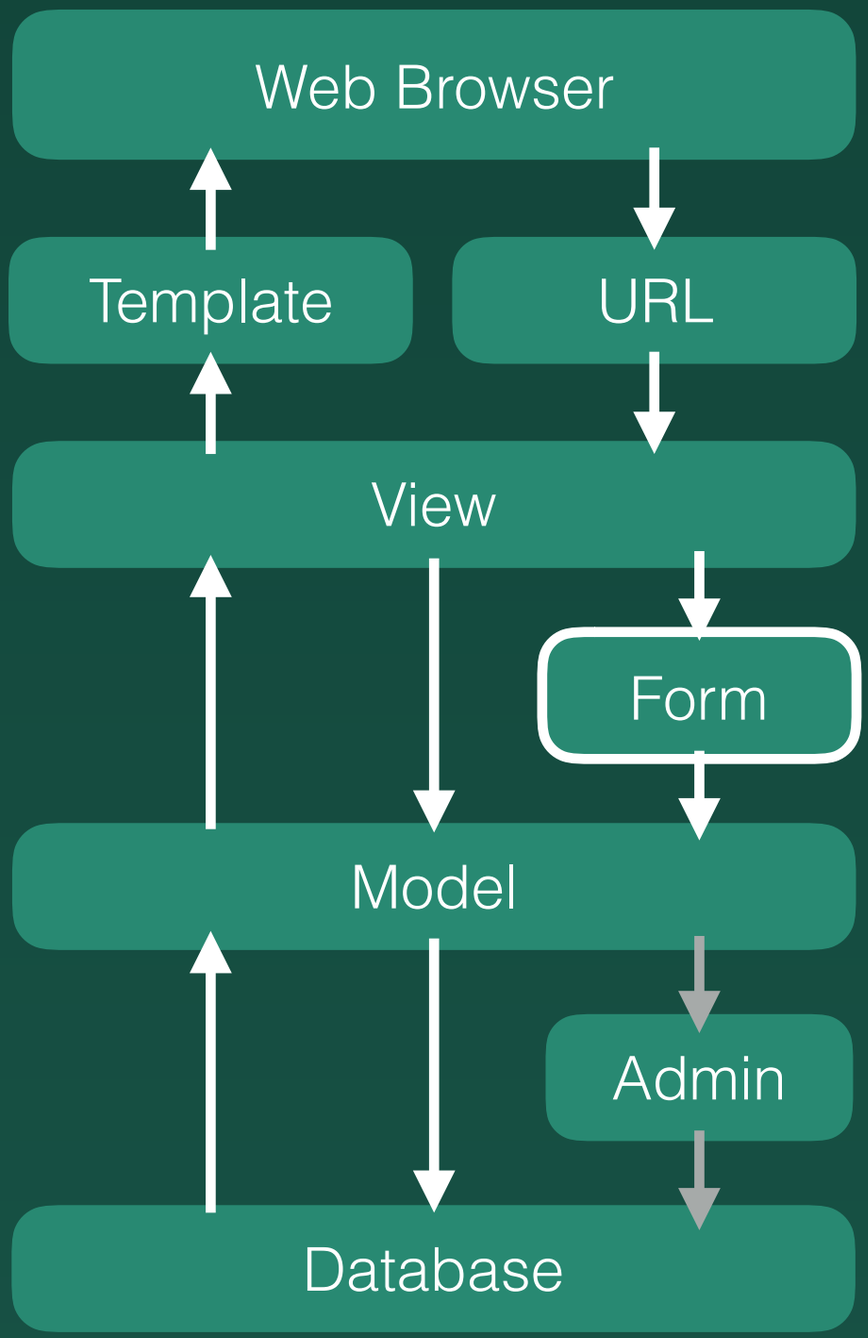
HTML Form ≠ Django Form

HTML Form

Consists of

- input element
- checkbox
- submit button
- radio button
- and much more

Basic Knowledge



Form

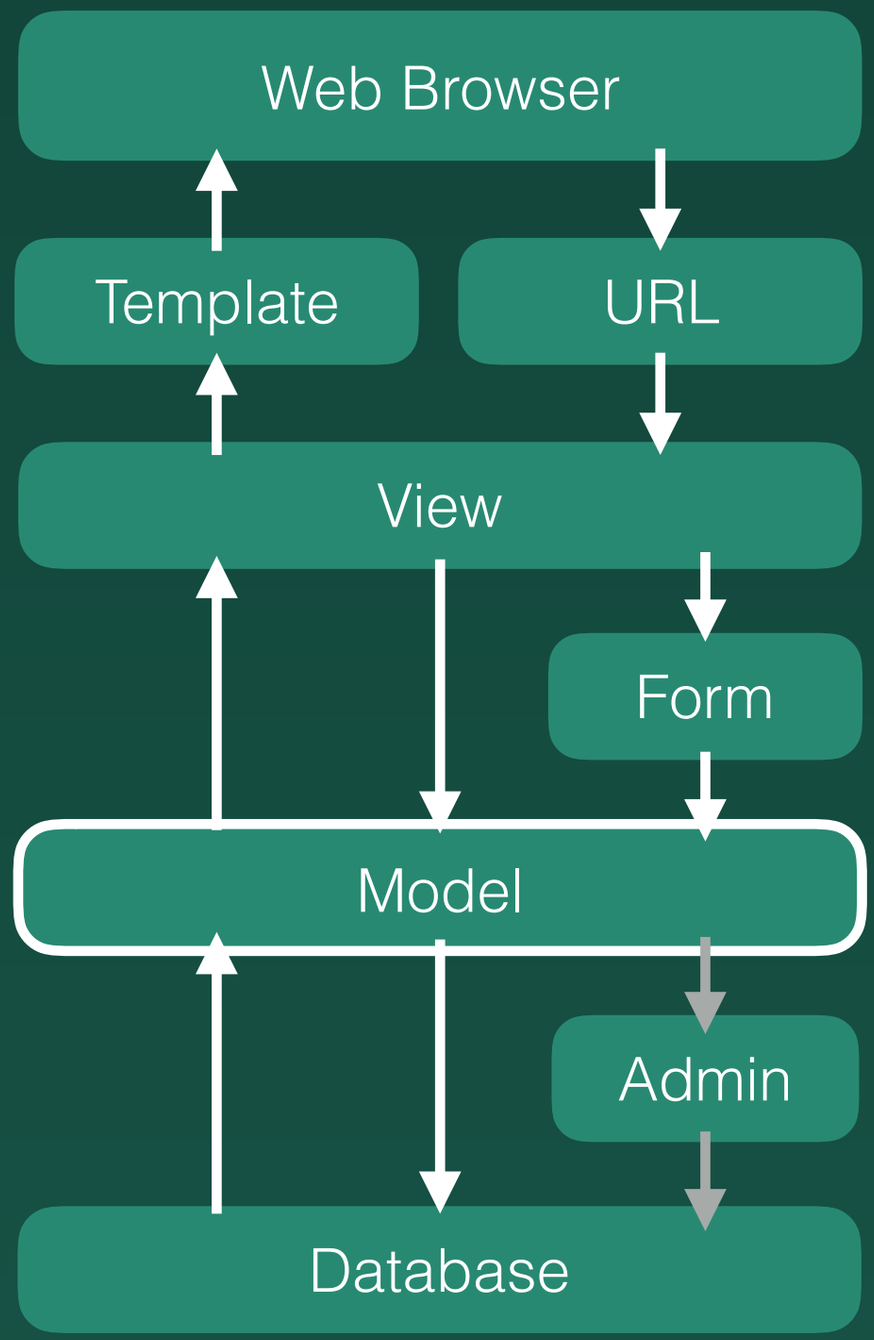
HTML Form ≠ Django Form

Django Form

Aims to

- fetch data from html form
- helps to connect to **Model**

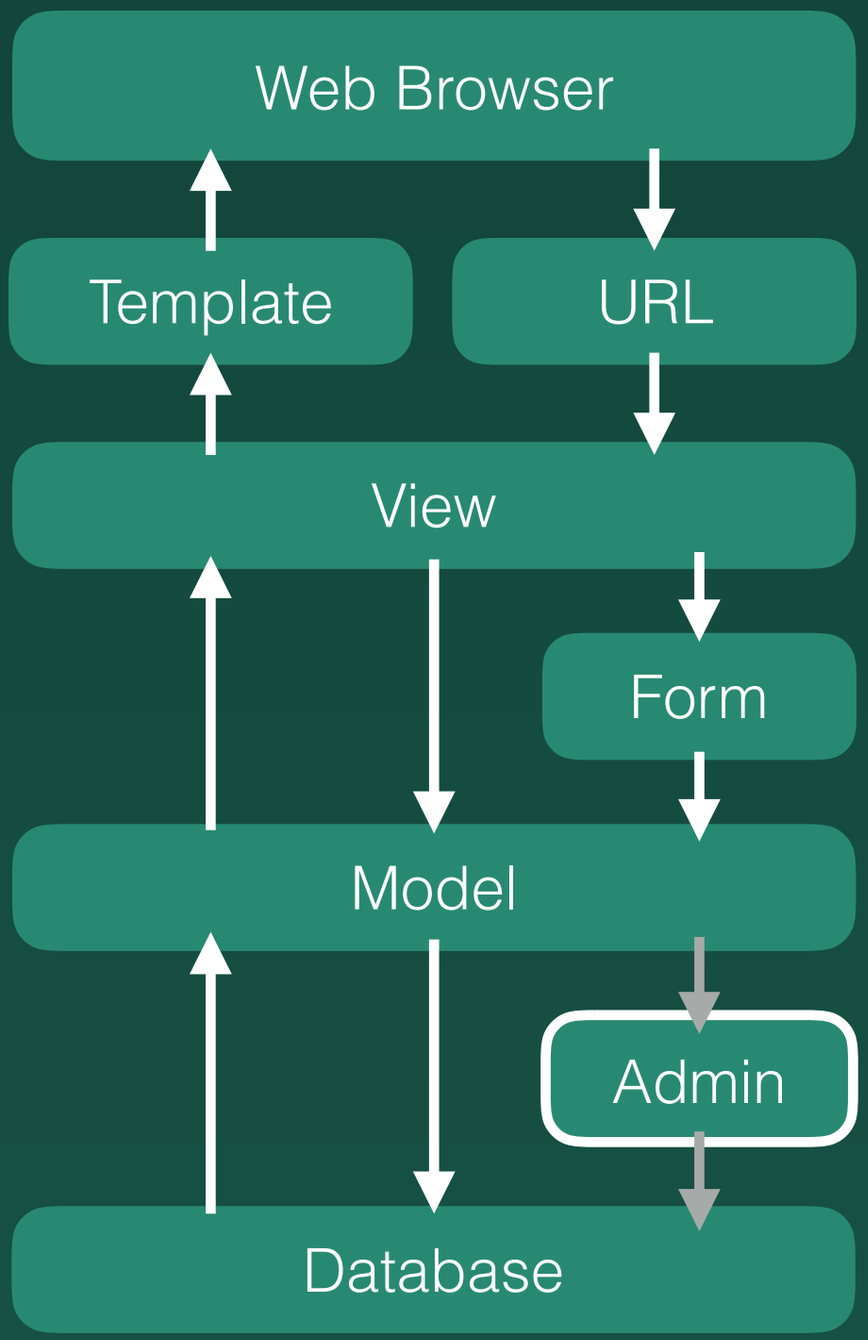
Basic Knowledge



Model

- Describes the structure of an object you want to store in **Database**
- Goes straight to **Database** and create & edit & request information as you wish

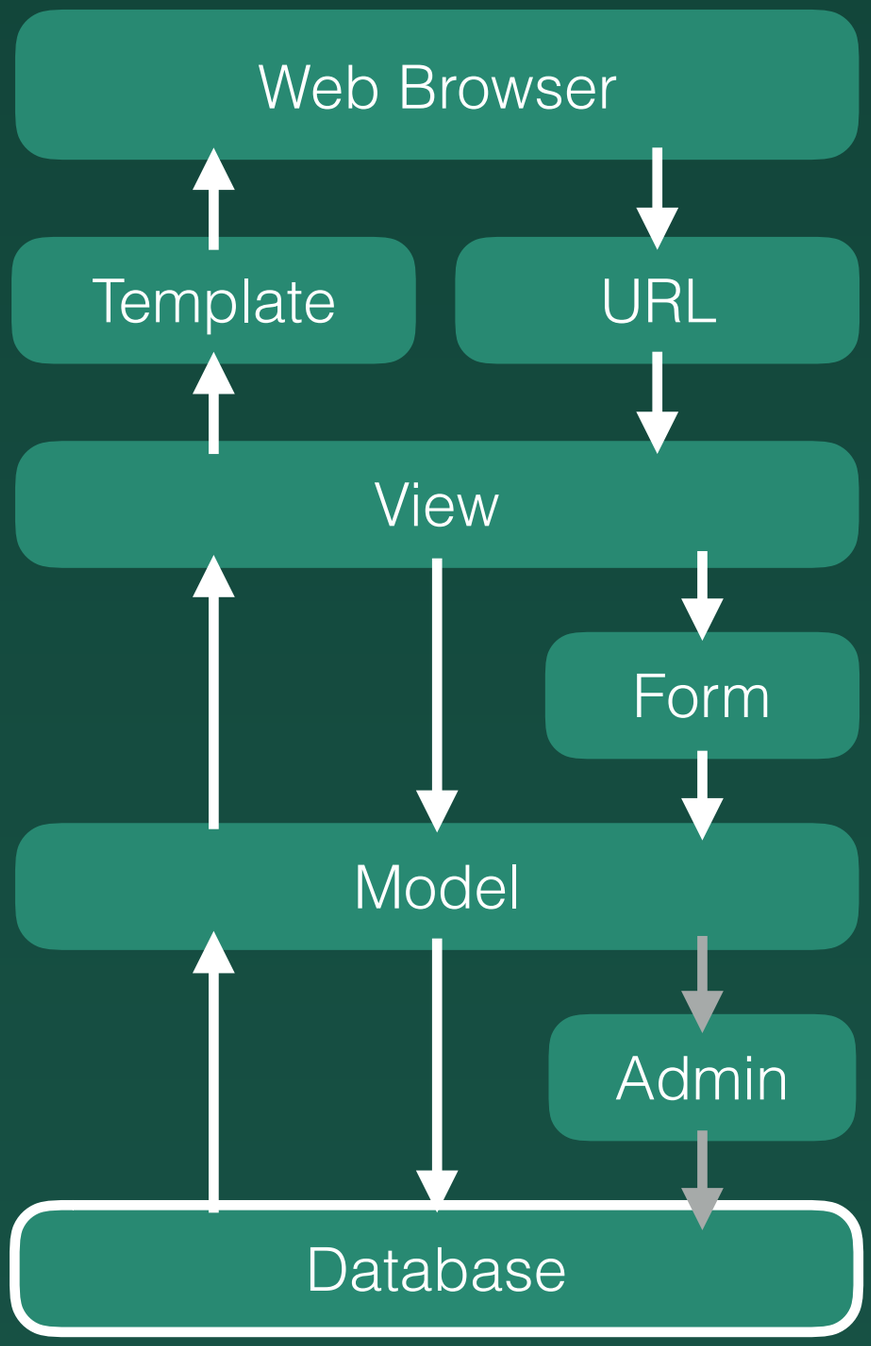
Basic Knowledge



Admin

- Helps to register your object in your **Model** so you can manage data in **Database**
- The registration has to be done in the first place

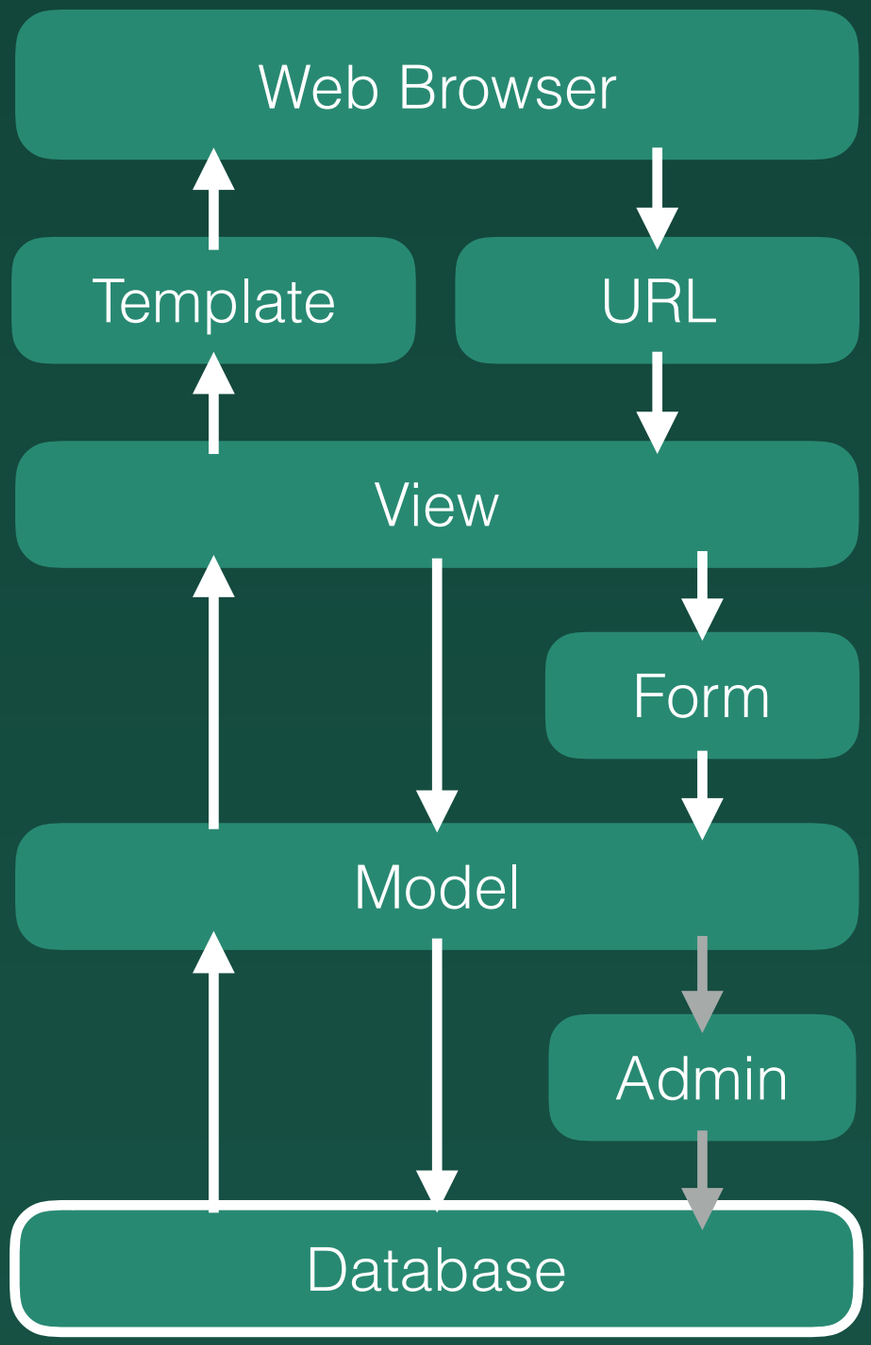
Basic Knowledge



Database

?

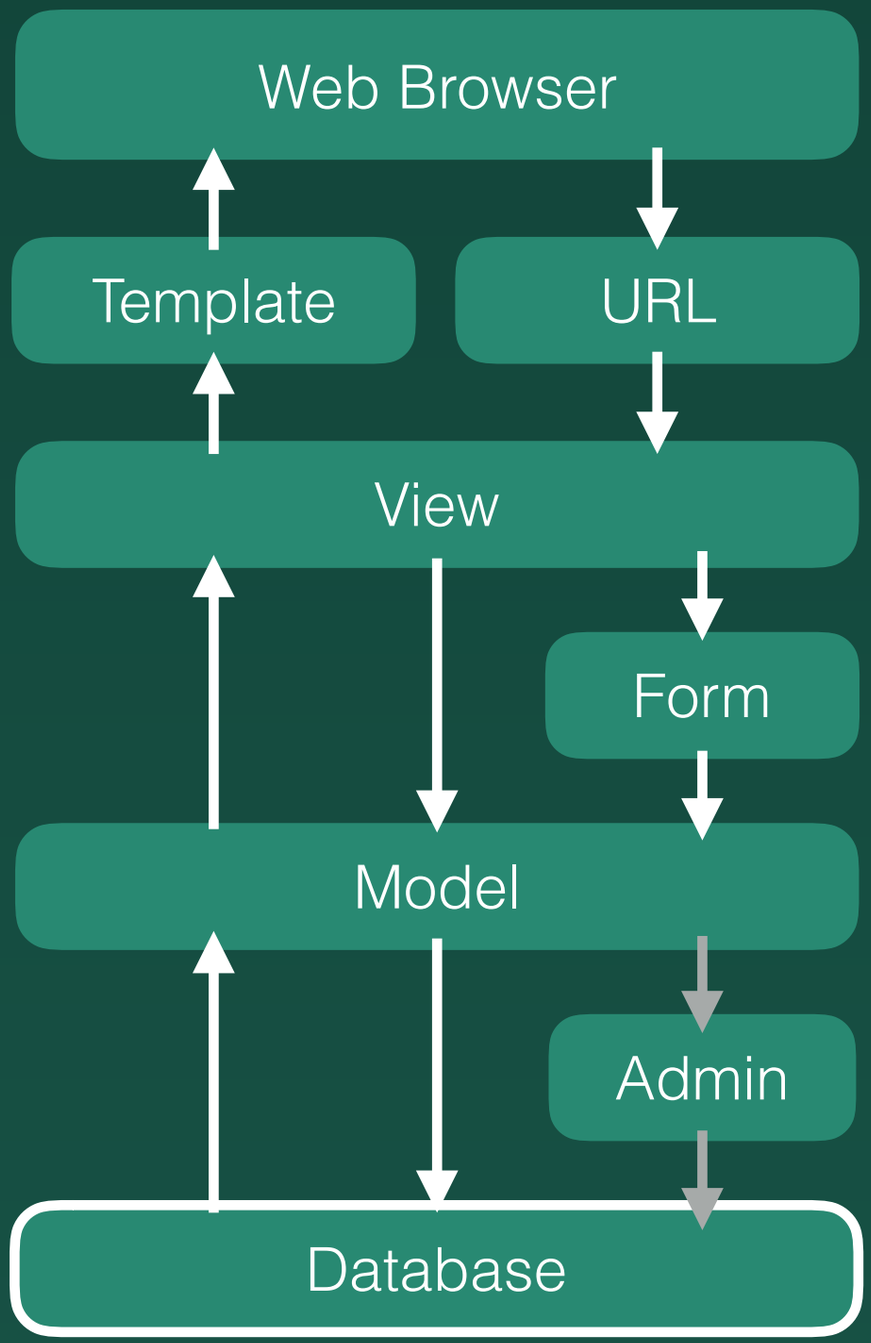
Basic Knowledge



Database

Collection of data

Basic Knowledge

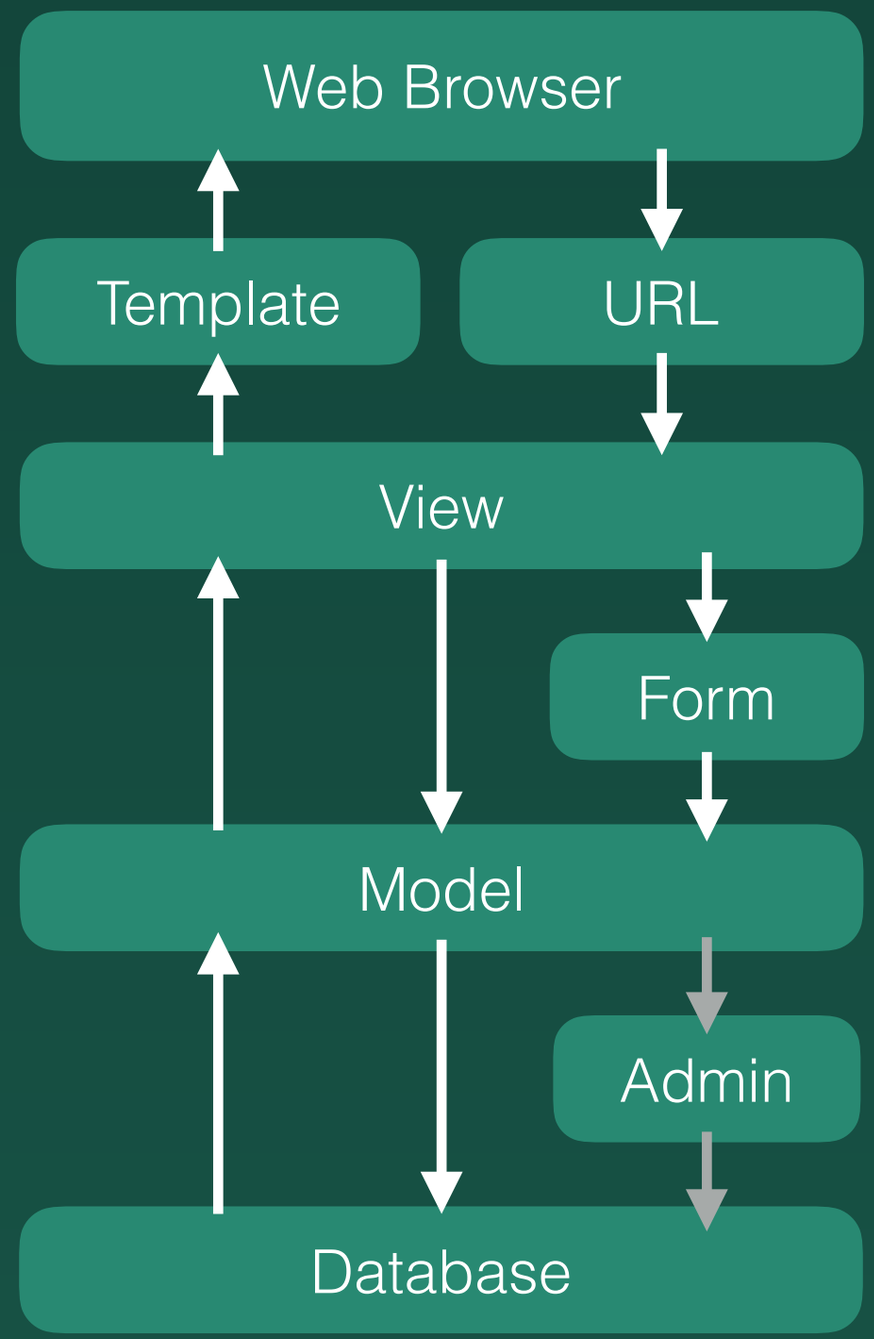


Database

- Collection of data
- Provided with a wonderful back-end platform for easy management

Basic Knowledge

Workflow



Outline

1. Introduction
2. **Basic Knowledge**
3. Technical Details
4. Tutorial
5. Conclusion

Outline

1. Introduction
2. Basic Knowledge
3. **Technical Details**
4. Tutorial
5. Conclusion

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Technical Details

Directory Architecture

project_name/ — Container of your entire project, which often referred as '**workspace**'

- **manage.py** — The command-line utility to **interact** with your Django project
  E.g1. python manage.py help E.g2. python manage.py runserver -h
- **your_project/** — The **name** of your Django project
  - **__inti__.py** — The file required for Python to treat this directory as a **package**
  - **settings.py** — **Configuration** for this Django project
  - **url.py** — Management of URLs to provide **mapping** to **view.py**
- **your_app/** — One of the web applications of this Django project
  - **__inti__.py**
  - **migration/** — The file which stores all the **variations** in your database
  - **static/** — The file which stores all of your **CSS, JS, images**
  - **templates/** — The file which stores all of your **Html**
  - **admin.py** — It reads your model and provides **interface** to your database
  - **form.py** — It is used to **fetch** data and performs **validation**
  - **model.py** — Description of the format or structure of an **object** stored in Database
  - **views.py** — All the **functions** needed to process or respond user's request
  - **db.sqlite3** — Your database

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Technical Details

Module

```
mypackage/
        __init__.py
        mymodule.py
```

- A module is a python source file
- A package is a directory of Python module(s)

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Technical Details

Commands

$ python manage.py startproject project_name

To start an project

Technical Details

Commands

$ python manage.py startapp app_name

To start an application

Technical Details

Commands

$ python manage.py createsuperuser

To create a superuser

Technical Details

Commands

$ python manage.py run server 0.0.0.0:8080

To start up your server

Note: 0.0.0.0:8080 —> address:port

Technical Details

Commands

$ python manage.py makemigrations

To create new migration based on the changes you made in your models

$ python manage.py migrate

To apply migration into your database

Note: Migration is Django’s way of propagating the changes you made into your database schema

Technical Details

Commands

$ python manage.py -h

To ask for what kind of command can be used

$ python manage.py runserver -h

To ask for what kind of command can be used after runserver

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Technical Details

Template Tags

```
{% for object in objects %}
    {{ object.attribute }}
{% endfor %}
```

Django saves you the trouble of repeating your codes

Technical Details

Template Tags

{{ form.as_p }}

This is Django’s way of rendering Html Form

Note: Remember to wrap it within your HTML form tag

Technical Details

Template Tags

{{ csrf_token }}

This is used prevent ill-intentioned hacker from hacking into your database

Note: Remember to wrap it within your HTML form tag

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Technical Details

High-level URL Configuration

url(r'^$', views.function)

?P< v > : to take v and sent to view as a variable
^ : beginning of the url
$: end of the url
() : to capture part of the pattern
+ : to indicate the previous item should repeat at least once
{} : to indicate a specific number of repetition

Technical Details

High-level URL Configuration

url(r’^anything/(?P<variable>[0-9]+)/$’, views.function)

?P< v > : to take v and sent to view as a variable
^ : beginning of the url
$: end of the url
() : to capture part of the pattern
+ : to indicate the previous item should repeat at least once
{} : to indicate a specific number of repetition

Technical Details

High-level URL Configuration

url(r'^(?P<variable>/[0-9]{n})/$', views.function)

?P< v > : to take v and sent to view as a variable
^ : beginning of the url
$: end of the url
() : to capture part of the pattern
+ : to indicate the previous item should repeat at least once
{} : to indicate a specific number of repetition

Technical Details

1. Directory Architecture
2. Module
3. Commands
4. Template Tags
5. High-level URL Configuration

Outline

1. Introduction
2. Basic Knowledge
3. **Technical Details**
4. Tutorial
5. Conclusion

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. **Tutorial**
5. Conclusion

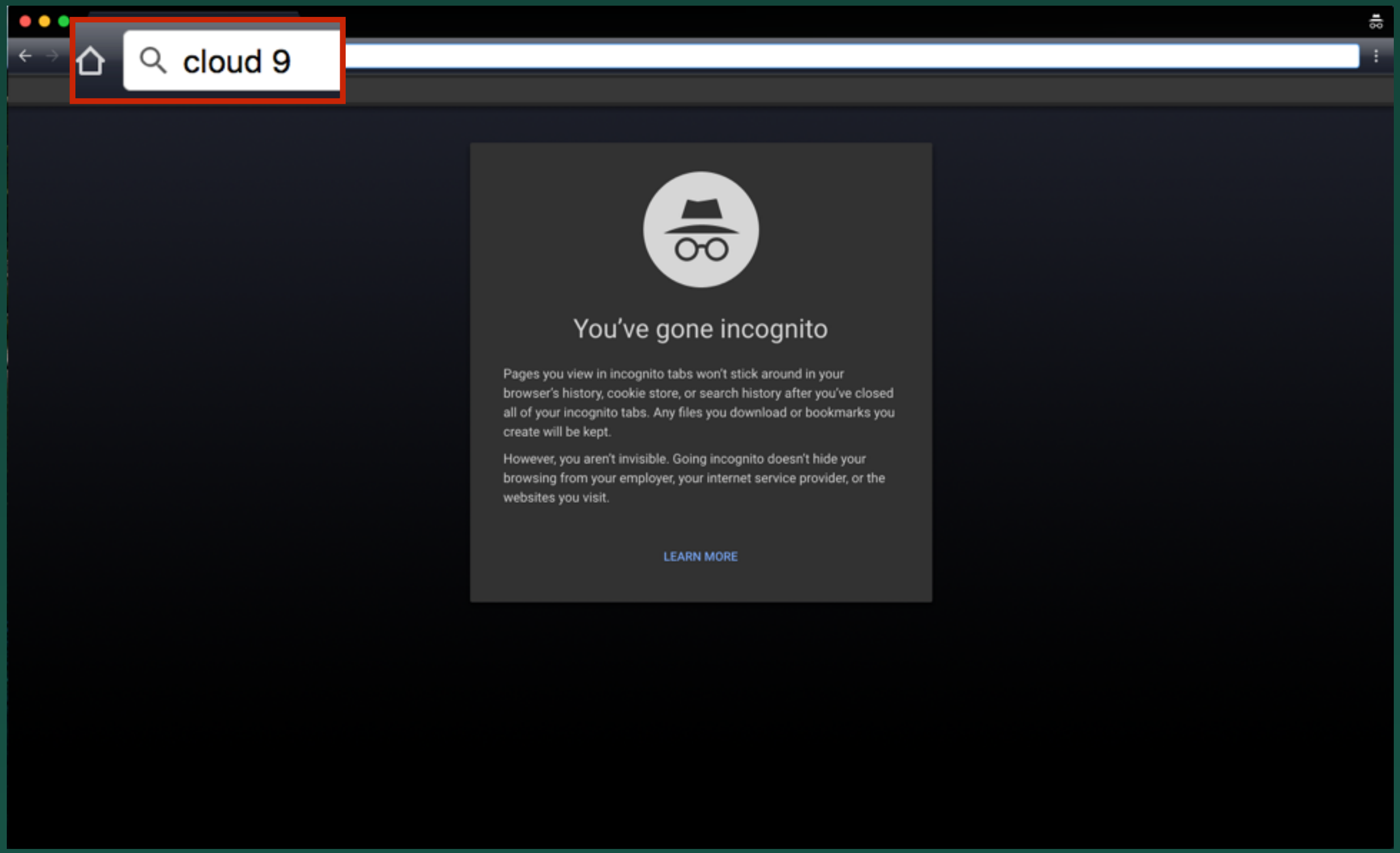
Tutorial

Build an online contact list

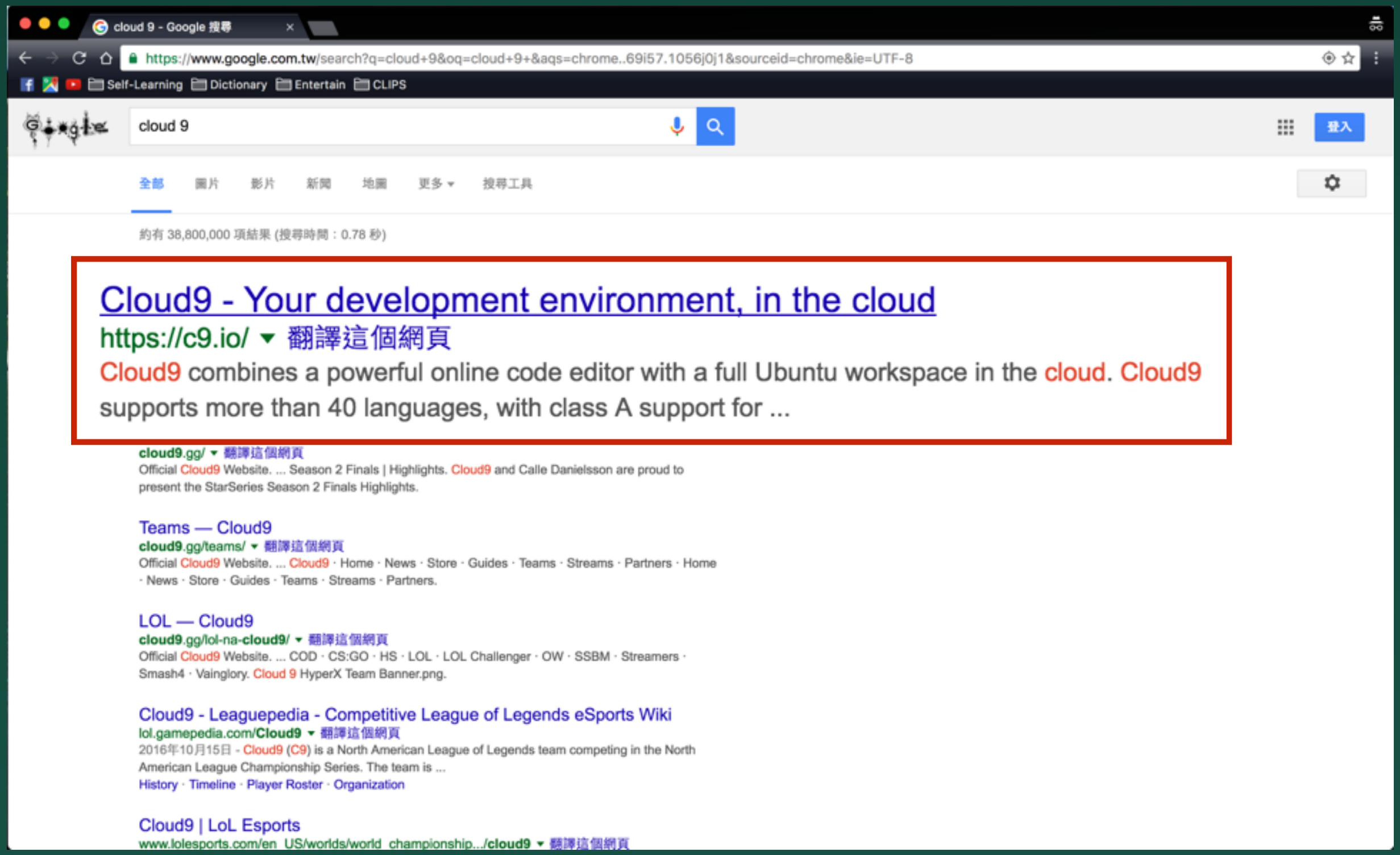
https://github.com/Denffer/django_tutorial

Find your Google Chrome

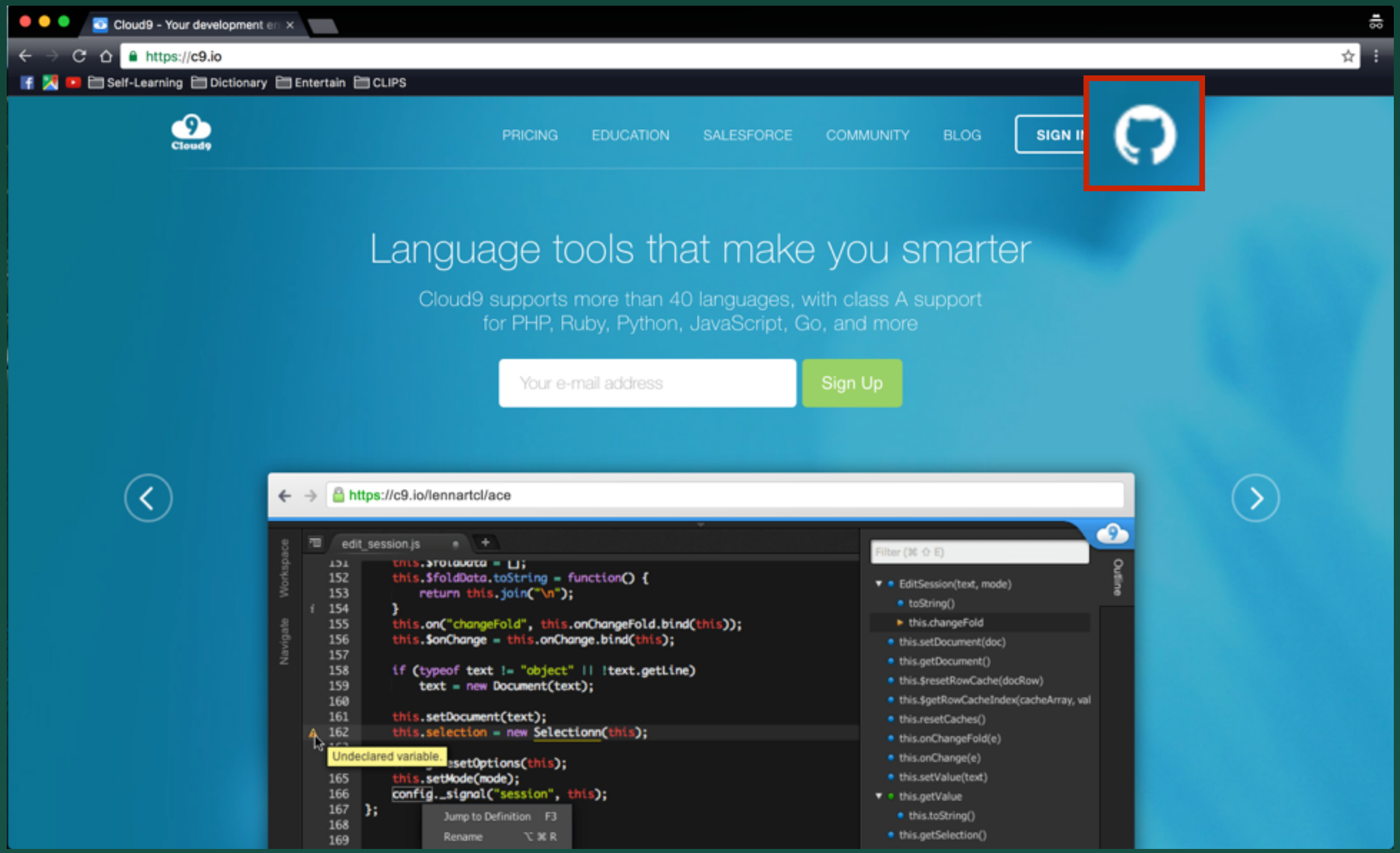
Google “Cloud 9”



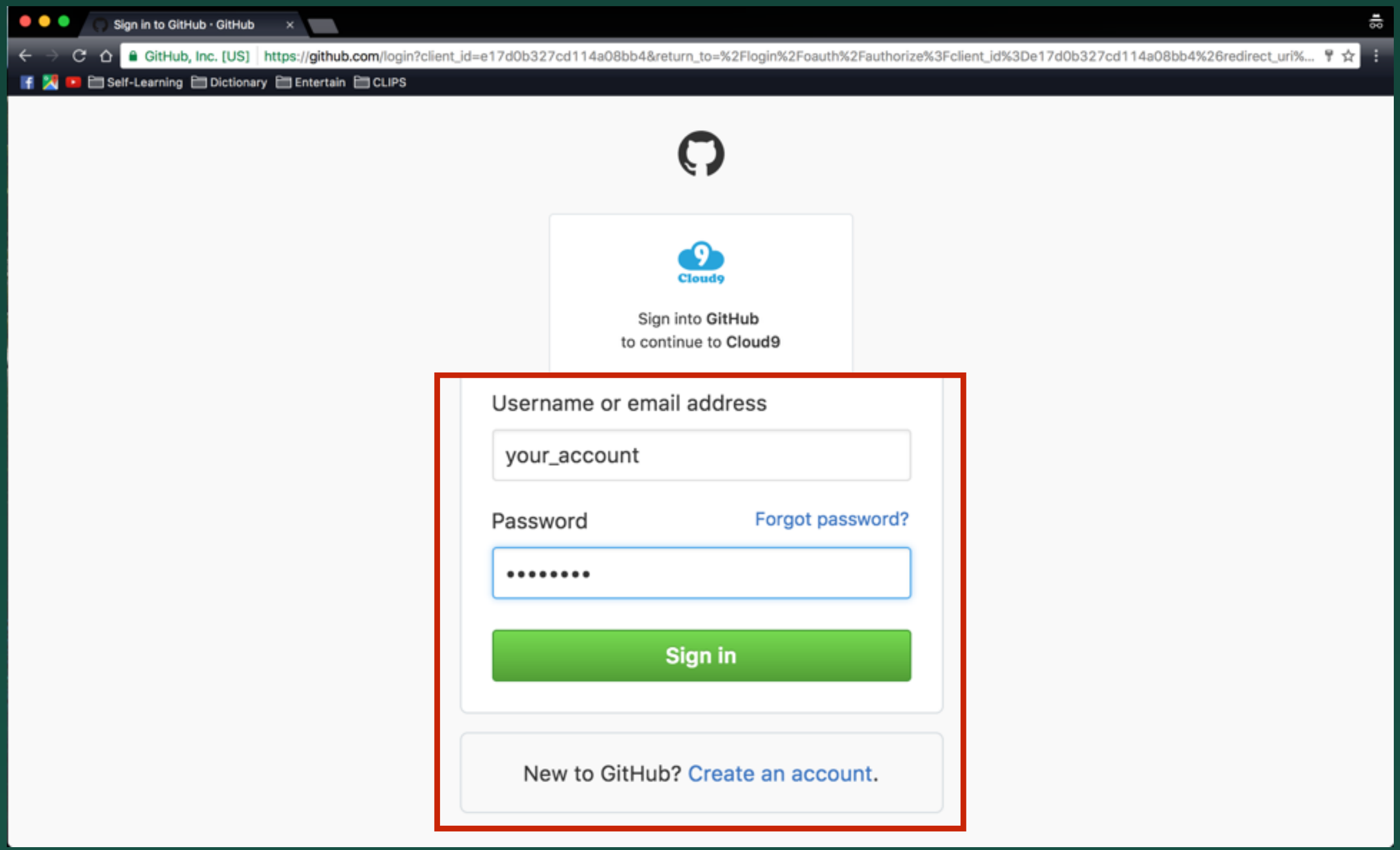
Find “Cloud 9”


cloud 9 - Google 搜尋
https://www.google.com.tw/search?q=cloud+9&oq=cloud+9+&aqs=chrome..69i57.1056j0j1&sourceid=chrome&ie=UTF-8
Self-Learning
Dictionary
Entertain
CLIPS
cloud 9
登入
全部
圖片
影片
新聞
地圖
更多
搜尋工具
約有 38,800,000 項結果 (搜尋時間：0.78 秒)
Cloud9 - Your development environment, in the cloud
https://c9.io/ ▾ 翻譯這個網頁
Cloud9 combines a powerful online code editor with a full Ubuntu workspace in the cloud. Cloud9 supports more than 40 languages, with class A support for ...
cloud9.gg/ ▾ 翻譯這個網頁
Official Cloud9 Website. ... Season 2 Finals | Highlights. Cloud9 and Calle Danielsson are proud to present the StarSeries Season 2 Finals Highlights.
Teams — Cloud9
cloud9.gg/teams/ ▾ 翻譯這個網頁
Official Cloud9 Website. ... Cloud9 · Home · News · Store · Guides · Teams · Streams · Partners · Home · News · Store · Guides · Teams · Streams · Partners.
LOL — Cloud9
cloud9.gg/lol-na-cloud9/ ▾ 翻譯這個網頁
Official Cloud9 Website. ... COD · CS:GO · HS · LOL · LOL Challenger · OW · SSBM · Streamers · Smash4 · Vainglory. Cloud 9 HyperX Team Banner.png.
Cloud9 - Leaguepedia - Competitive League of Legends eSports Wiki
lol.gamepedia.com/Cloud9 ▾ 翻譯這個網頁
2016年10月15日 - Cloud9 (C9) is a North American League of Legends team competing in the North American League Championship Series. The team is ...
History · Timeline · Player Roster · Organization
Cloud9 | LoL Esports
www.lolesports.com/en_US/worlds/world_championship.../cloud9 ▾ 翻譯這個網頁

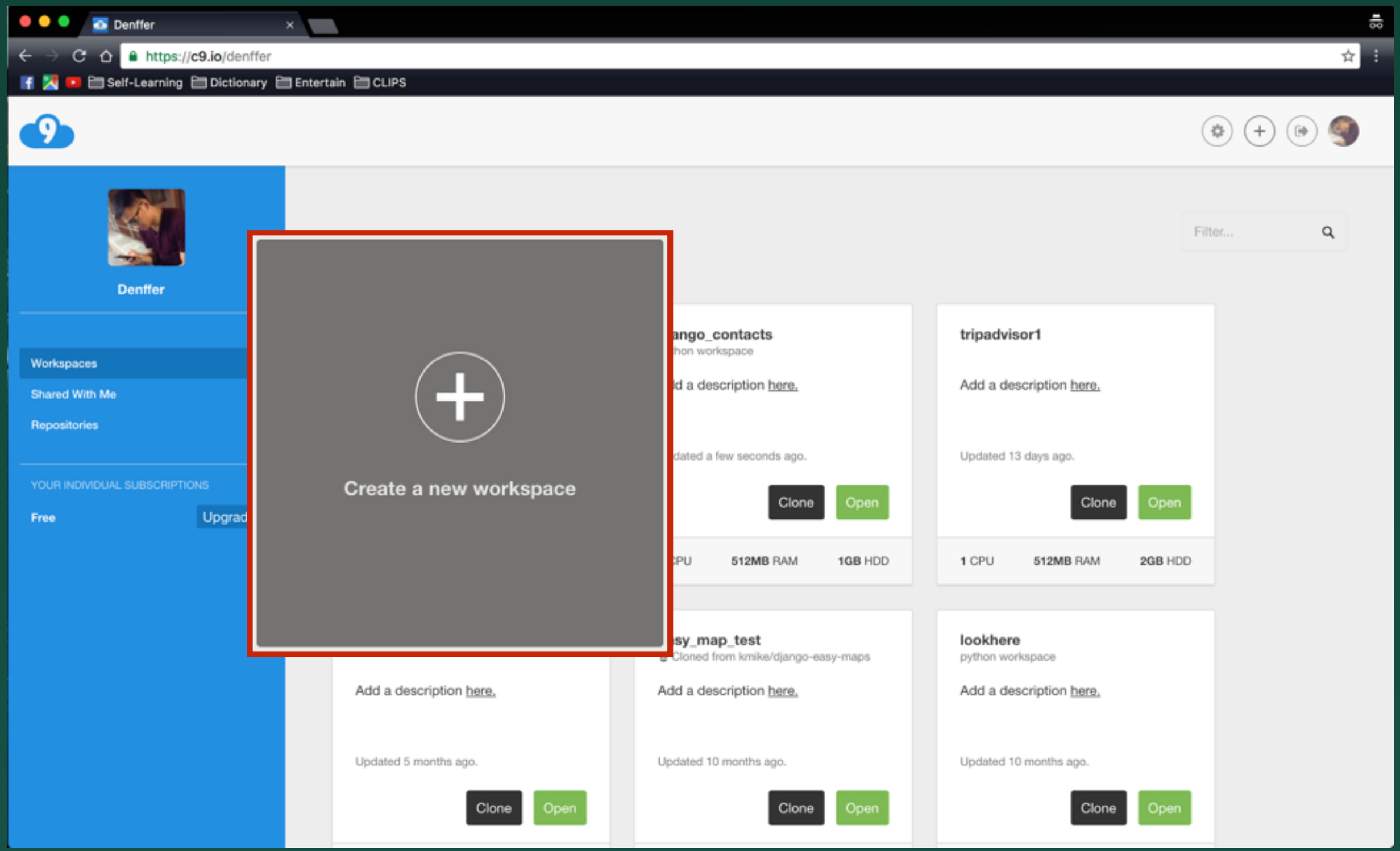
Click on Github logo



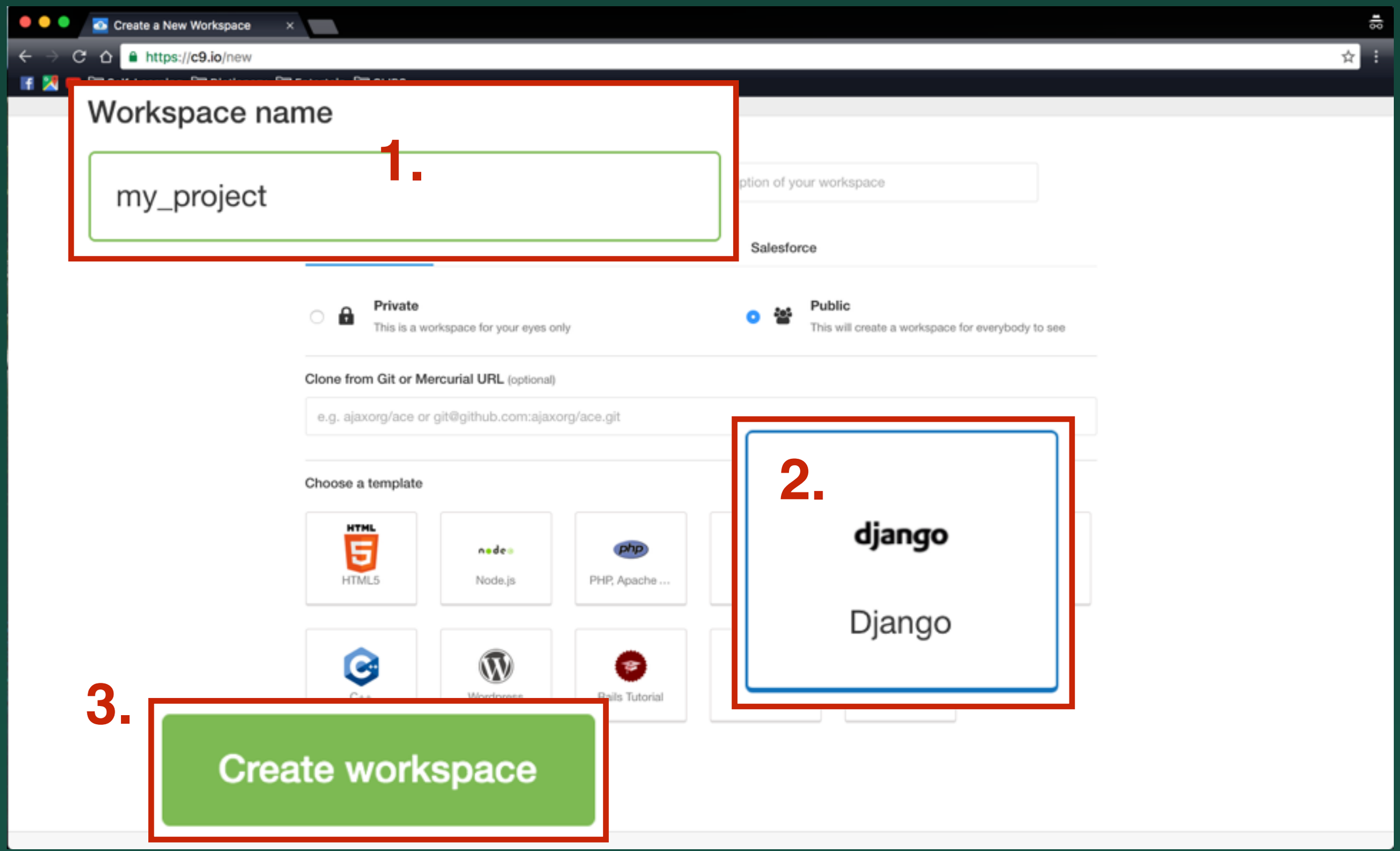
Login your Github



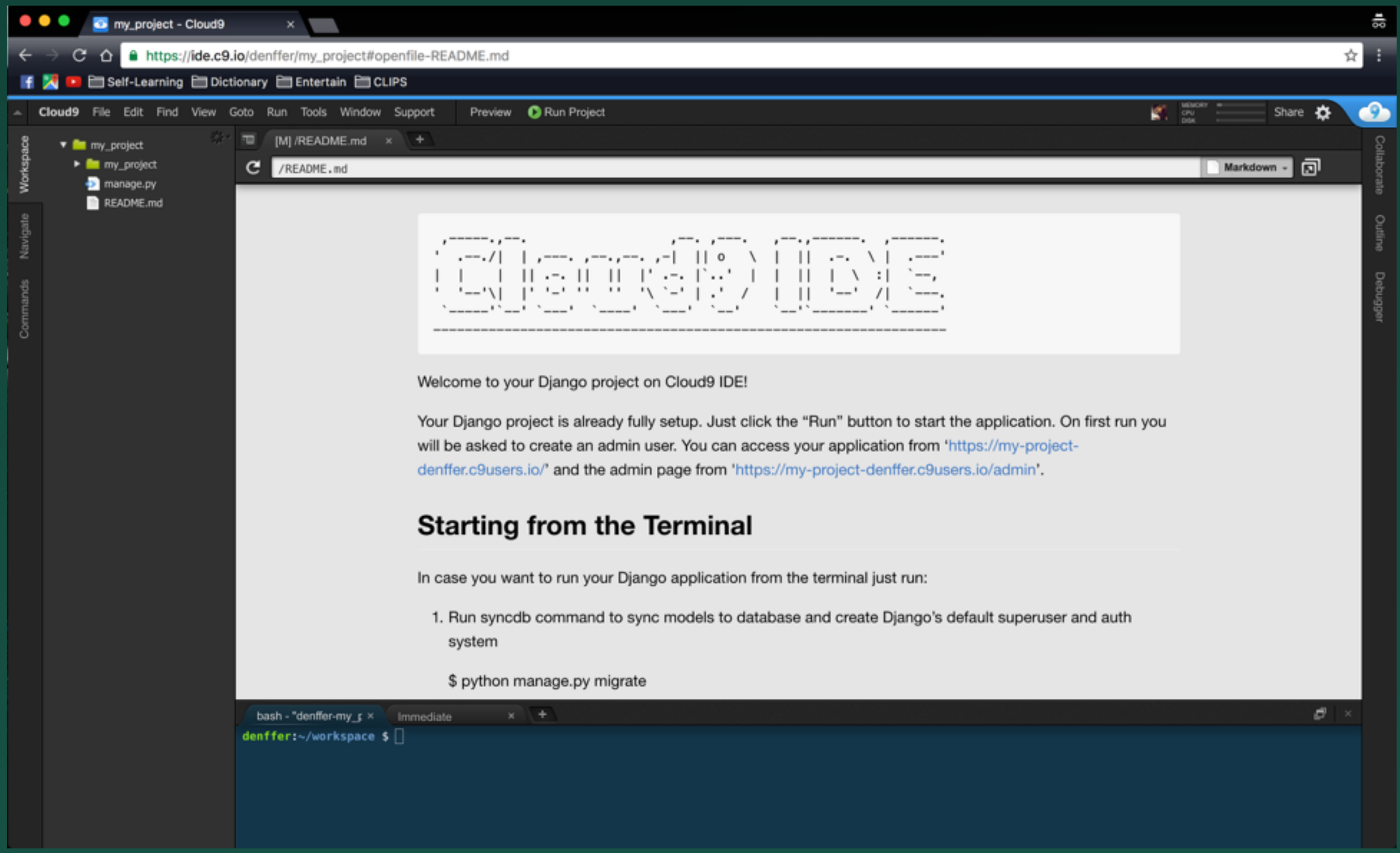
Create a new workspace



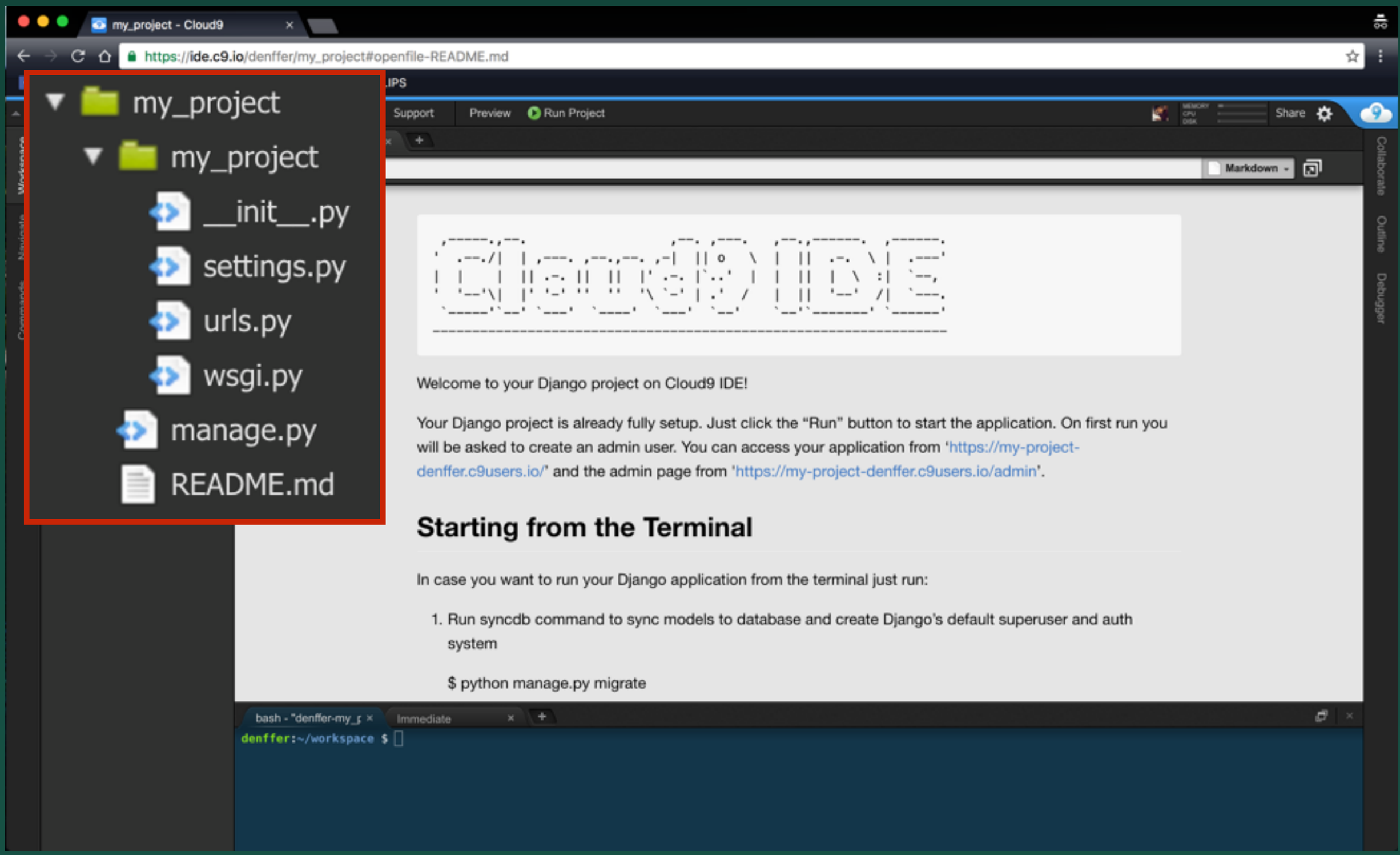
Create a new workspace



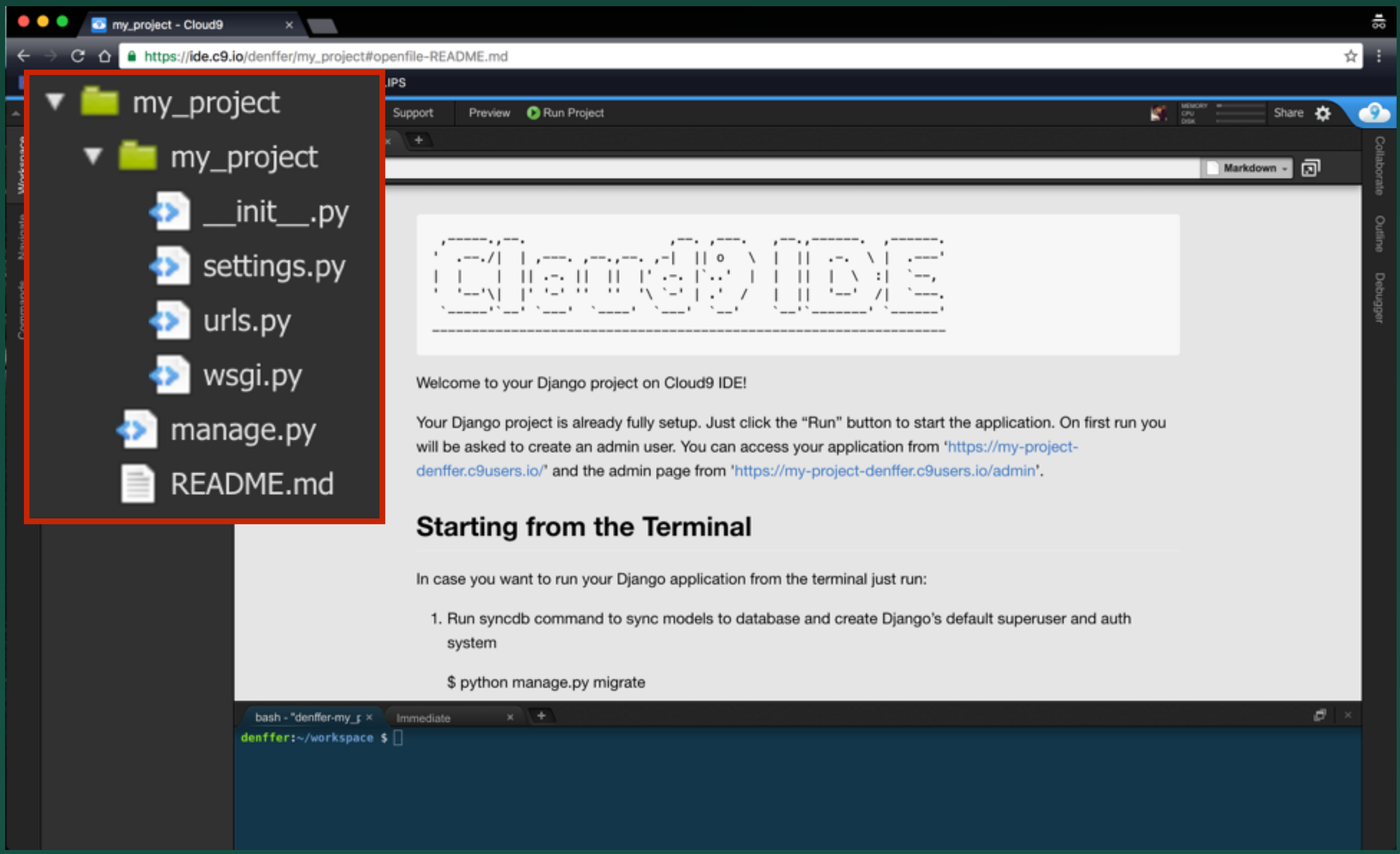
Welcome to Django on Cloud 9



Click on ‘my_contacts’ and see what are the things initially given by Django



Anything beyond these should be added or created by yourself

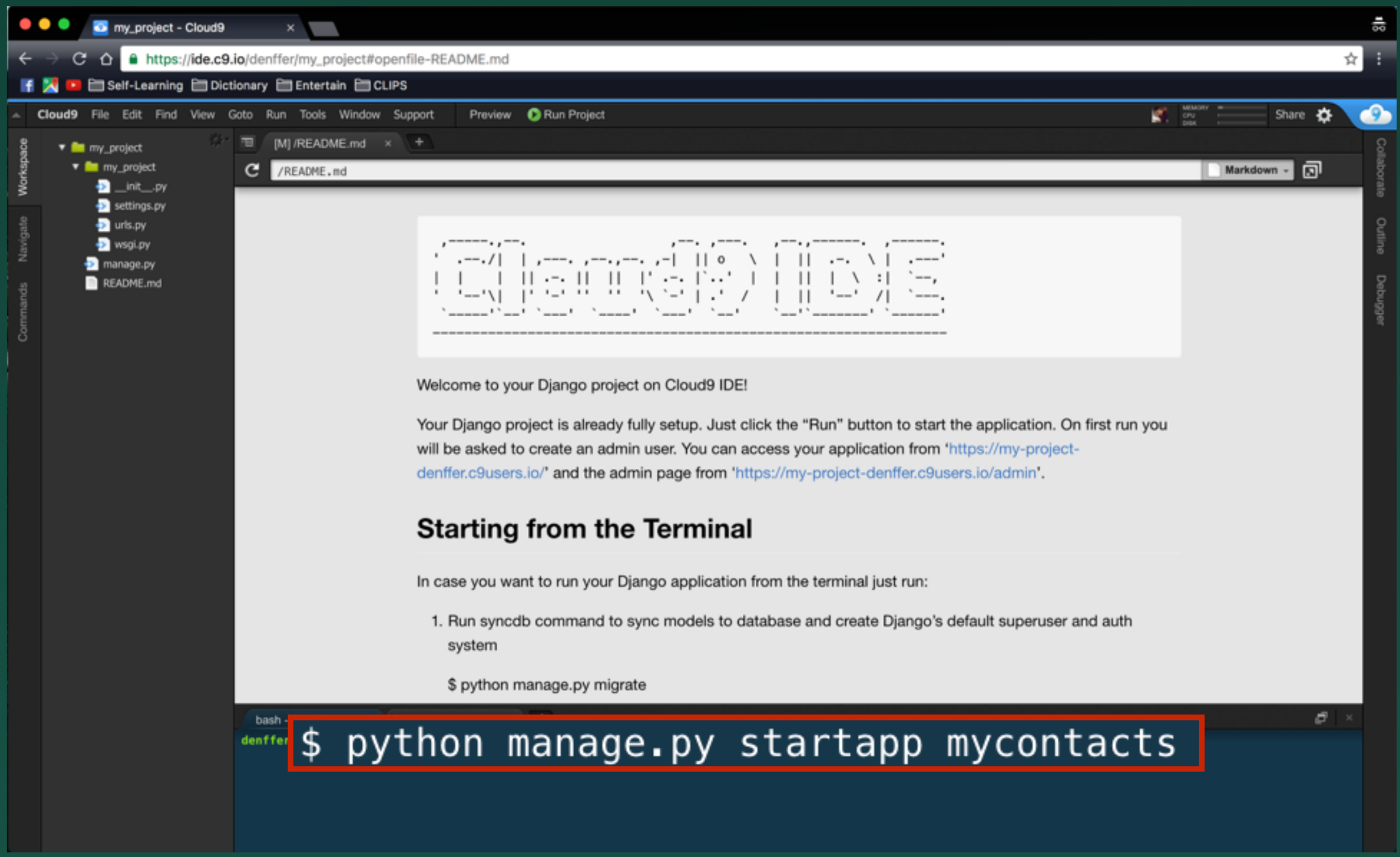


Once your workspace is created on Cloud 9,
your project is created as well.

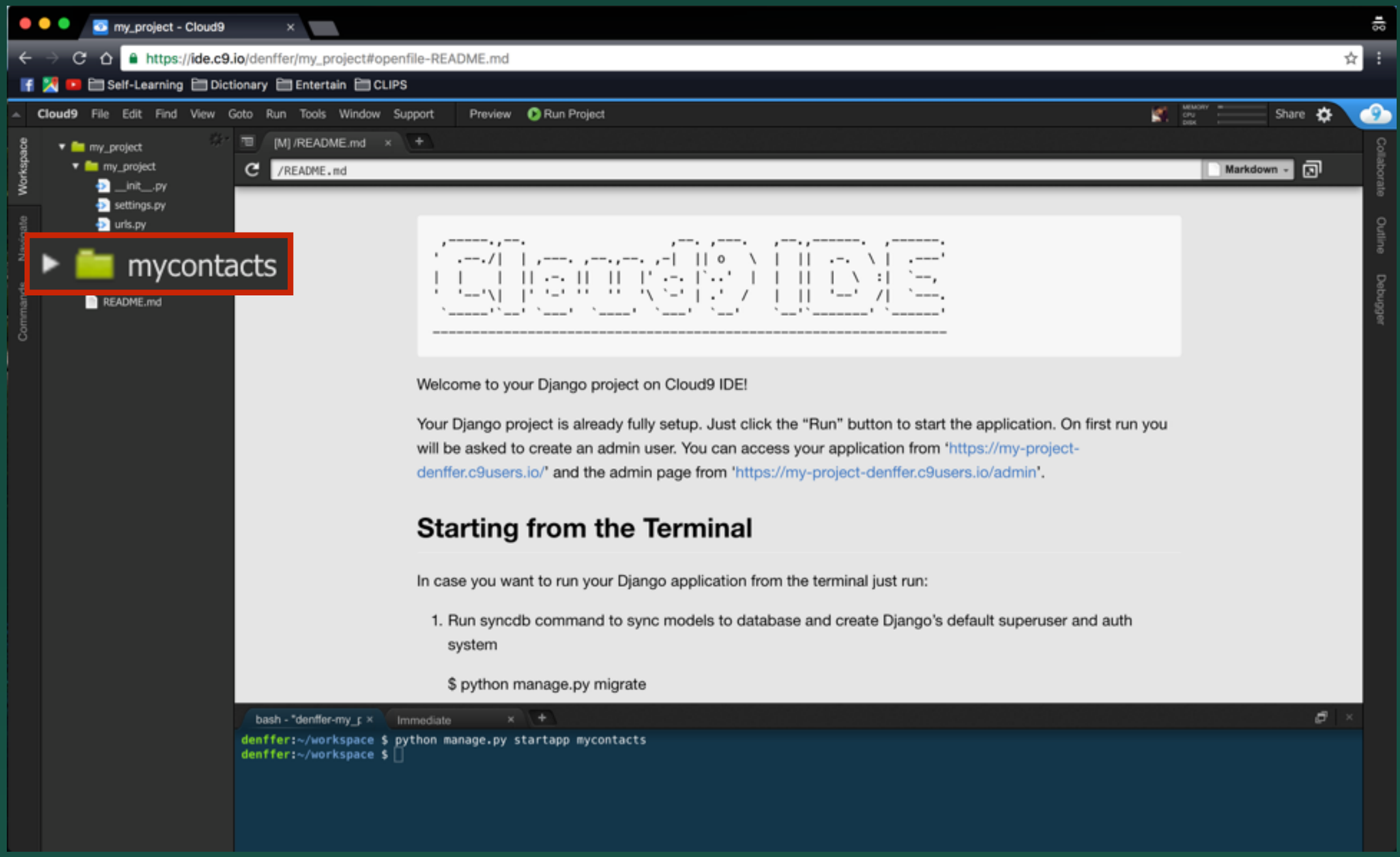
So there is **NO** need for you to

```
$ python manage.py startproject my_project
```

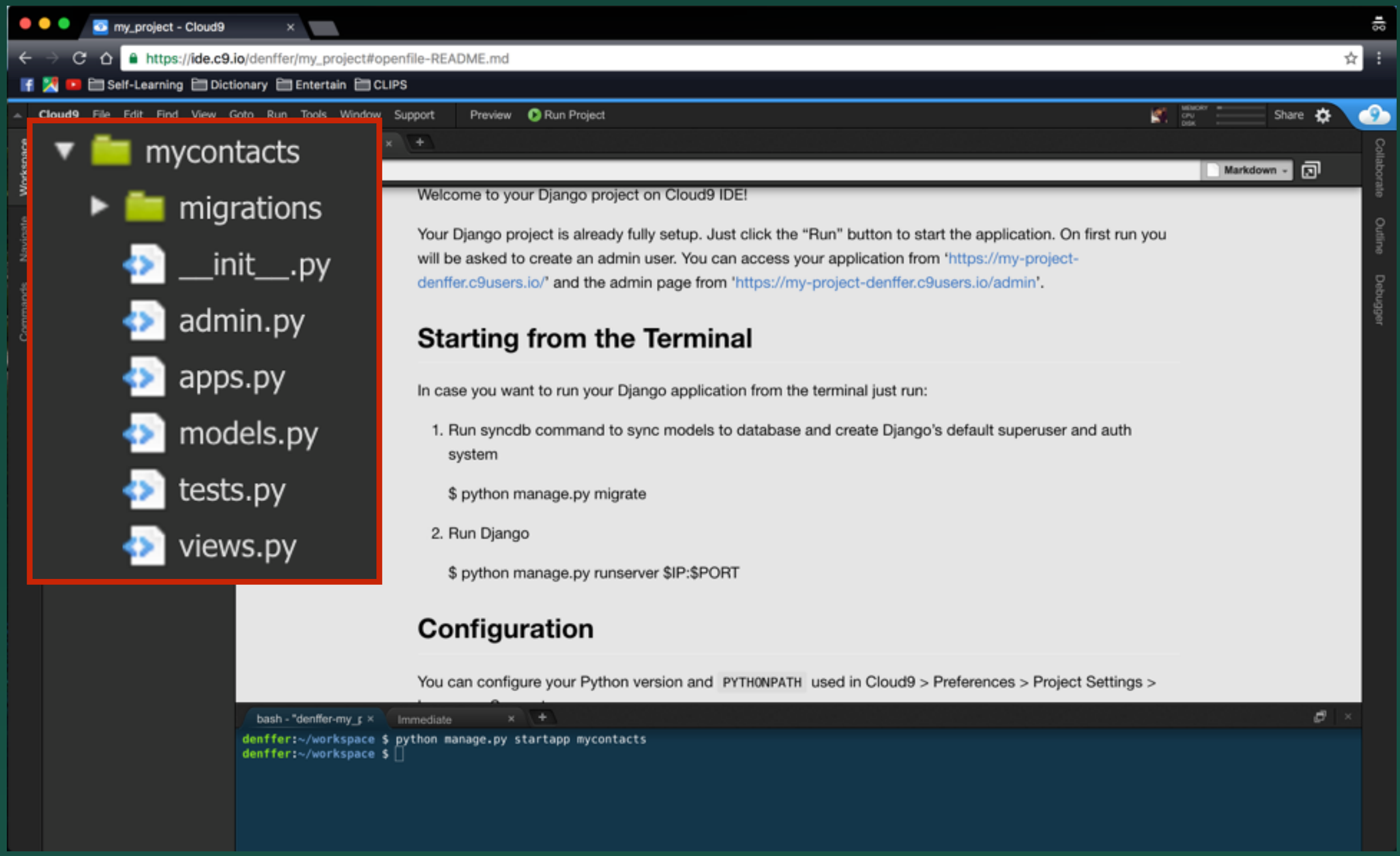
Now, create your first Django application



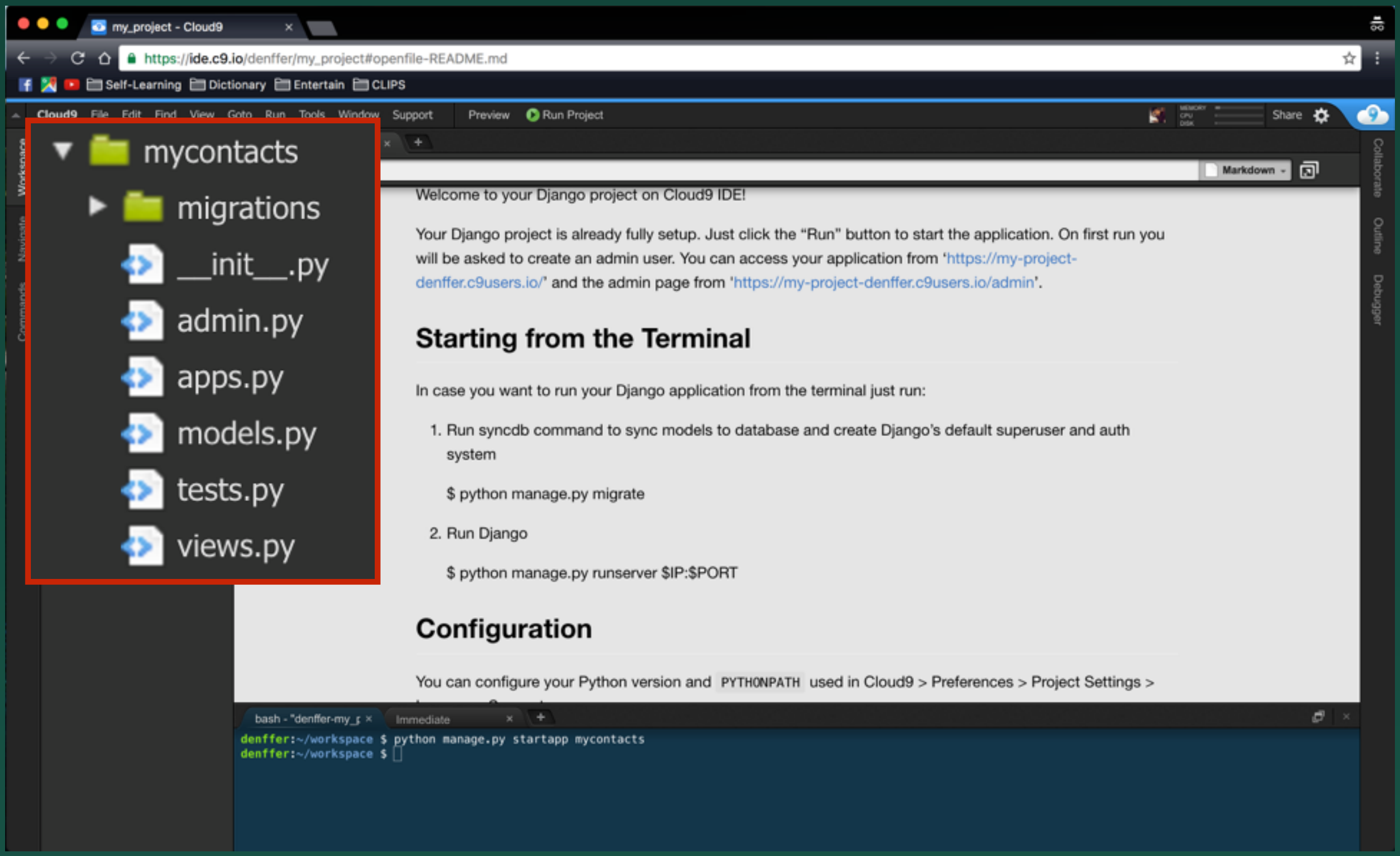
There you go. Your first Django application



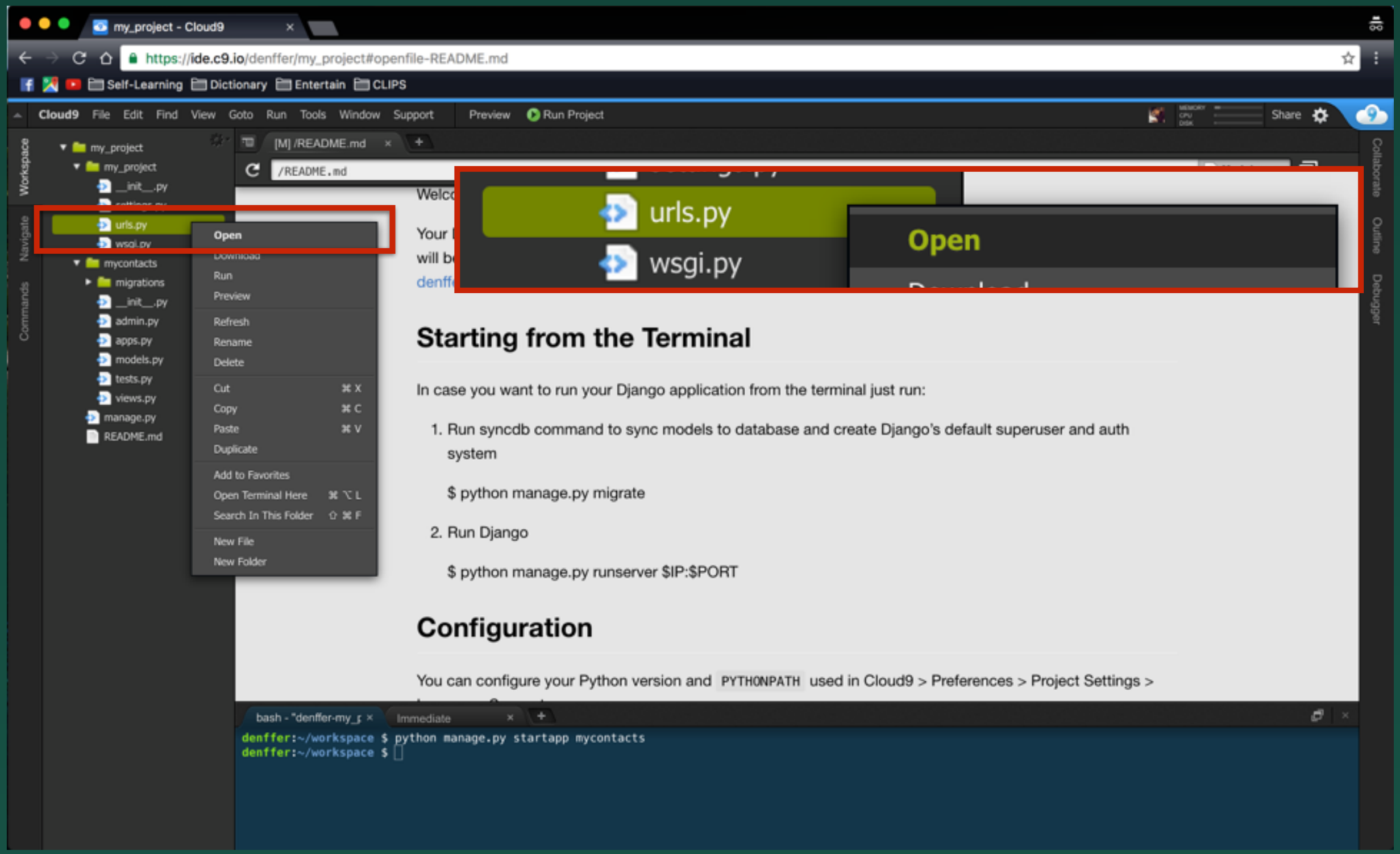
Check on what is inside



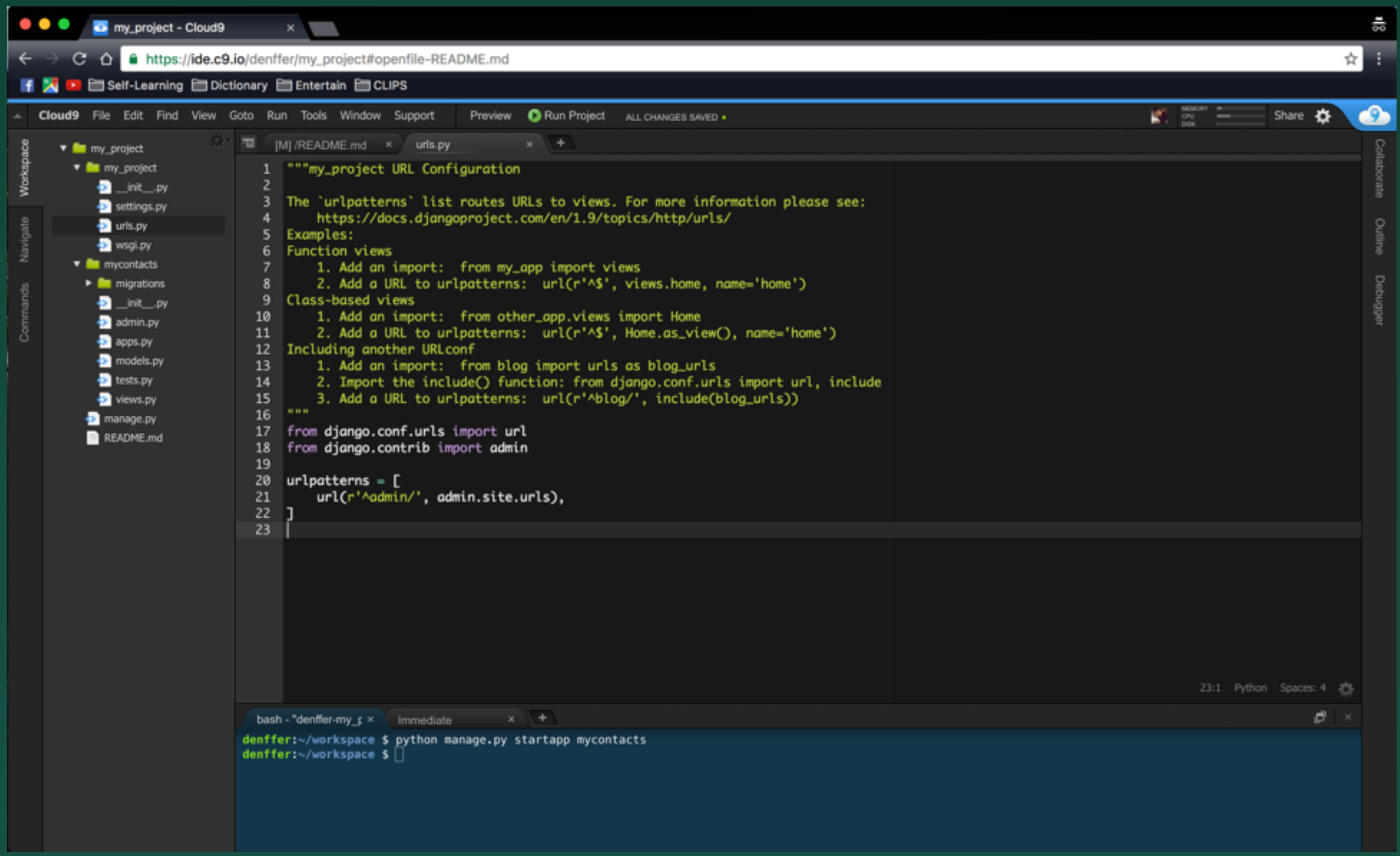
See that? Your application is actually a package, and it contains the modules we've seen earlier.



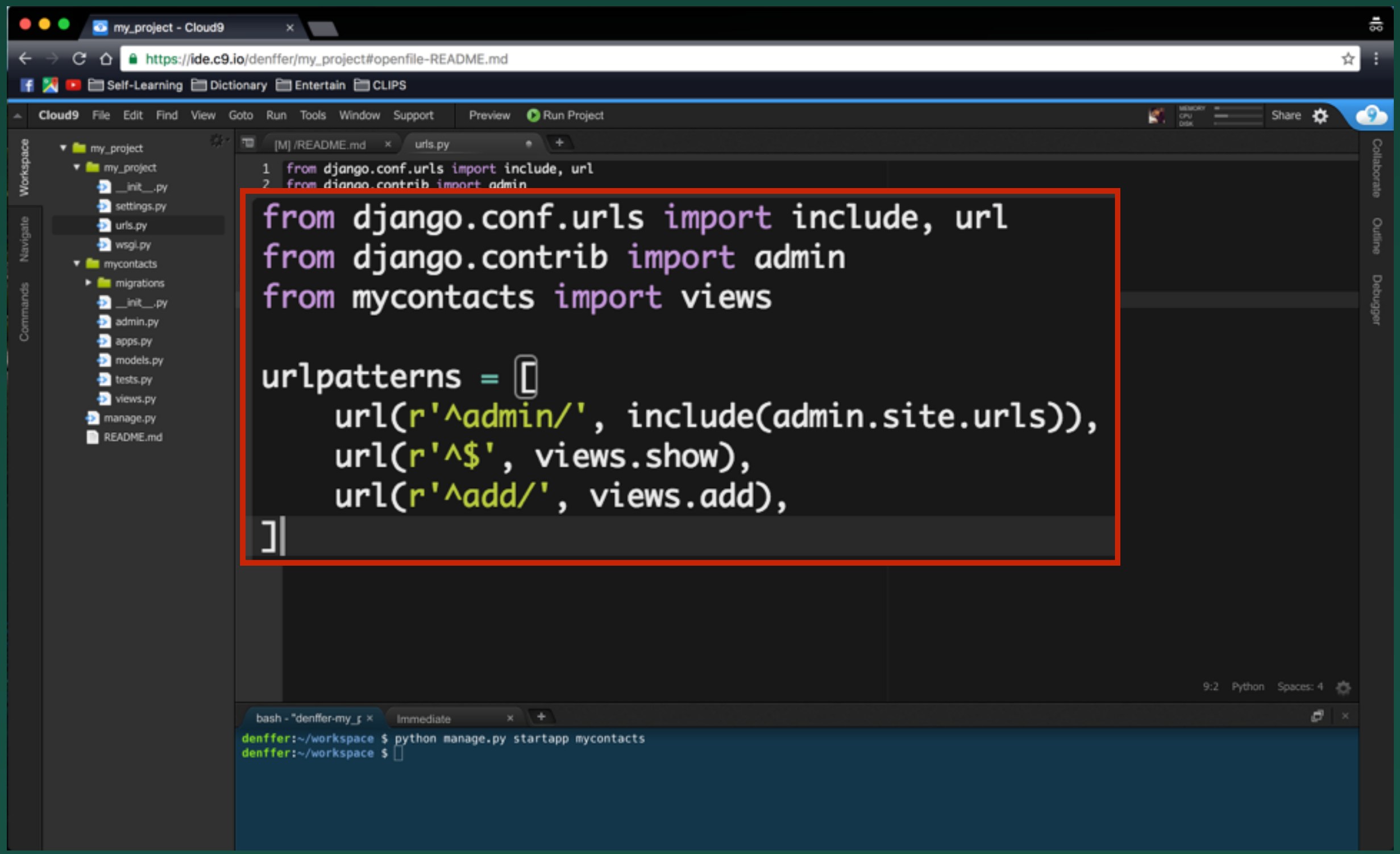
Open your urls.py



The default url given by Django is the web address connecting to your back-end management platform

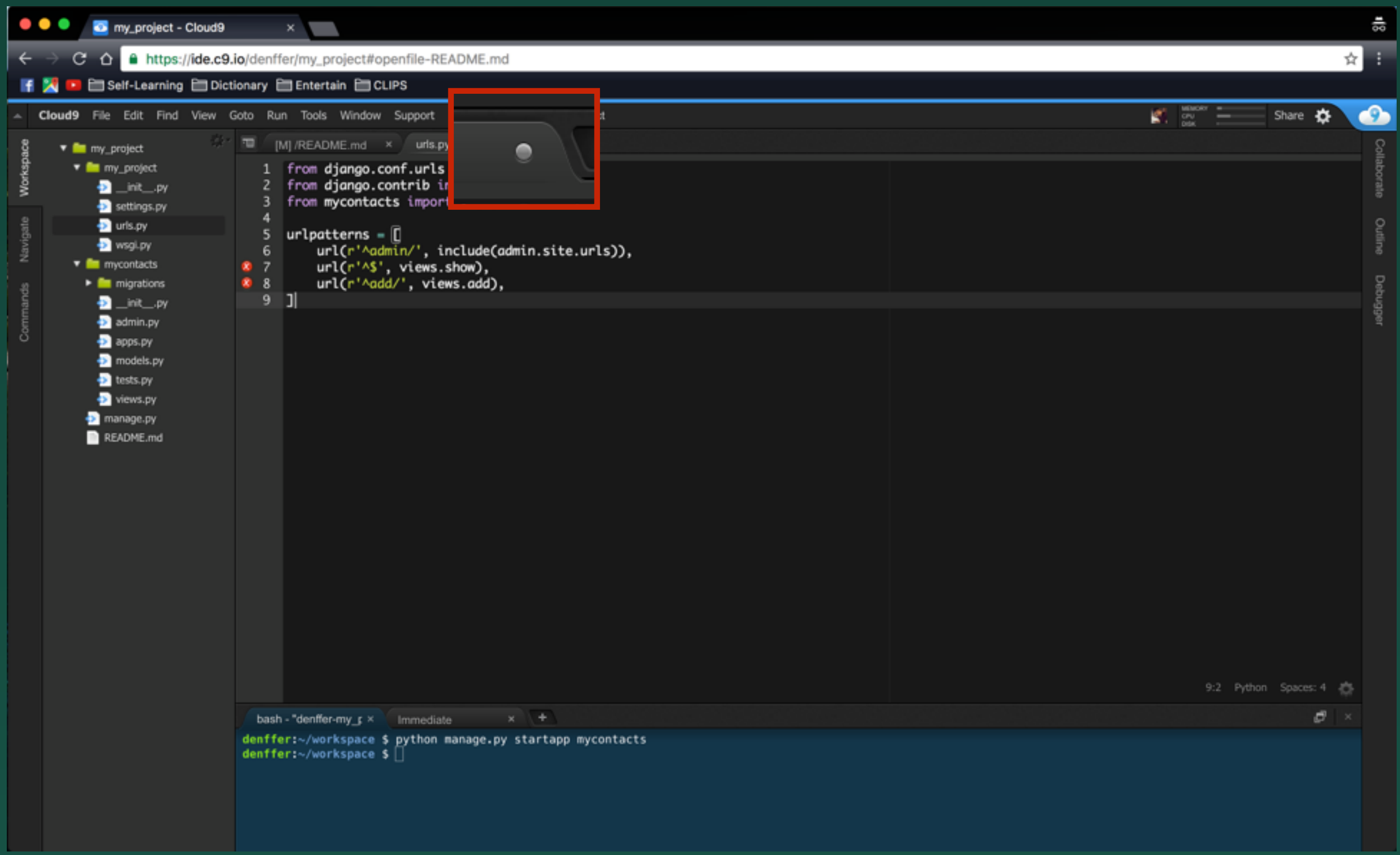


Now, copy and paste the codes in **urls.py** from the repository you cloned from Github

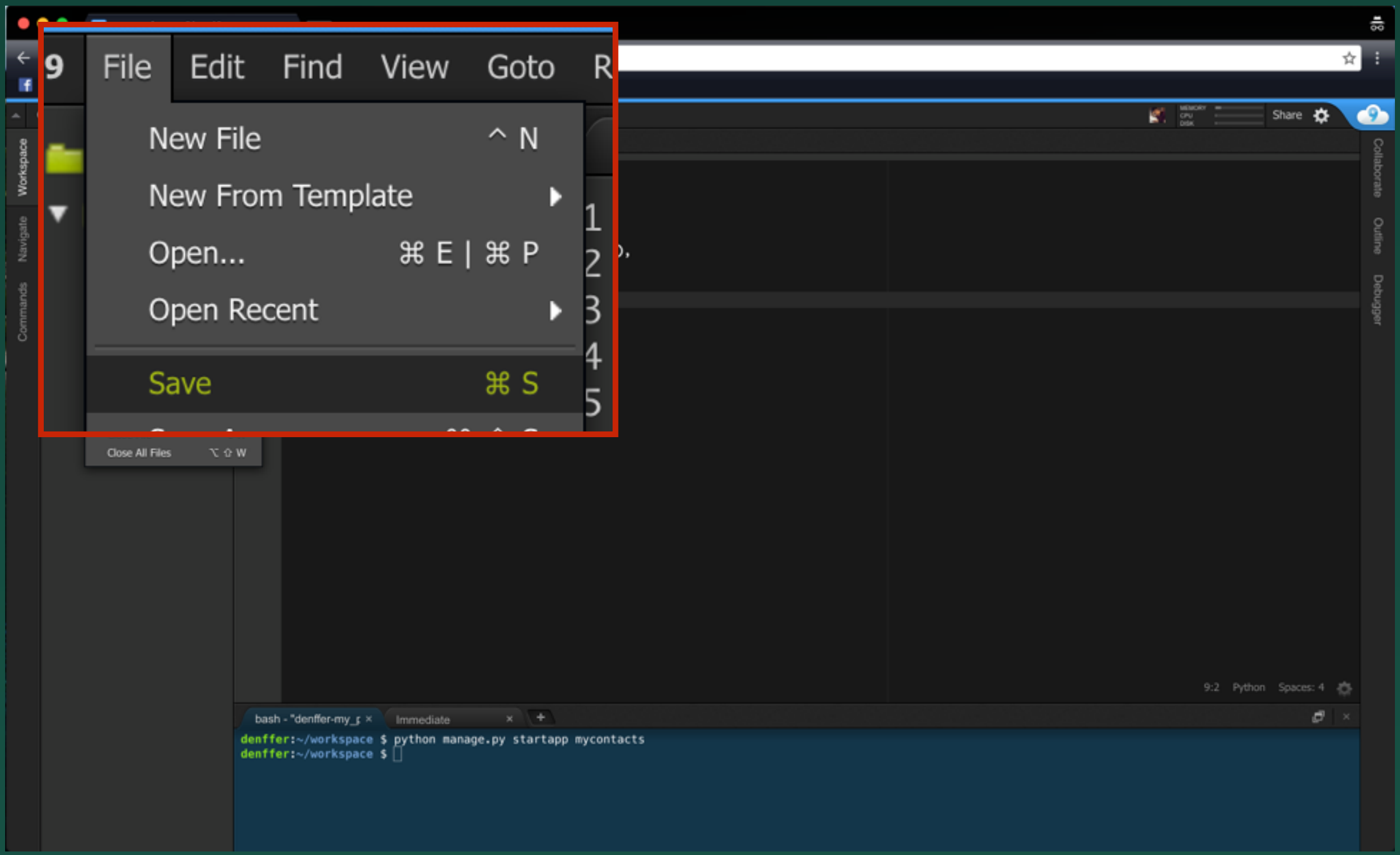


Do you see the dot over here?

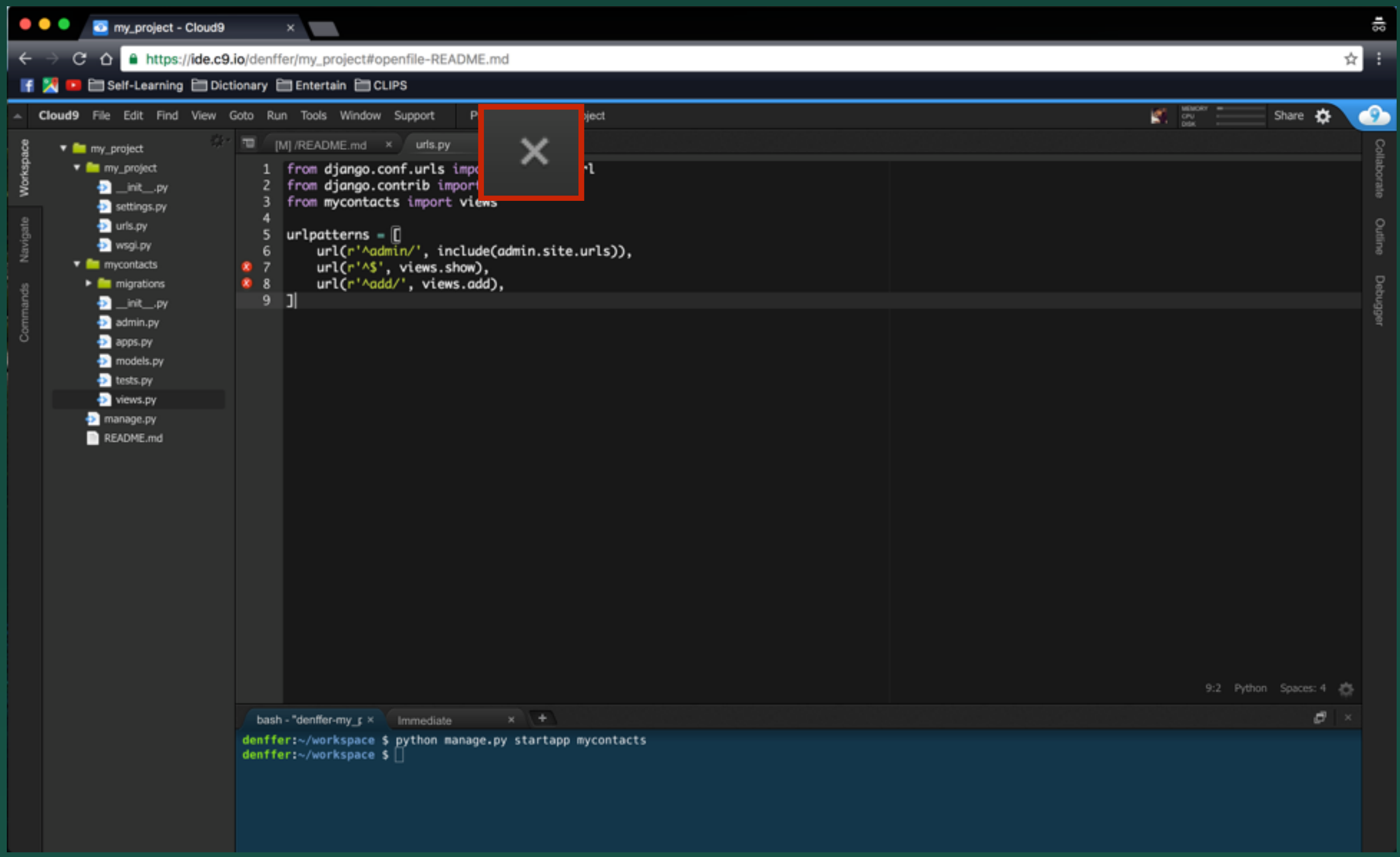
This means that the file is **NOT** saved



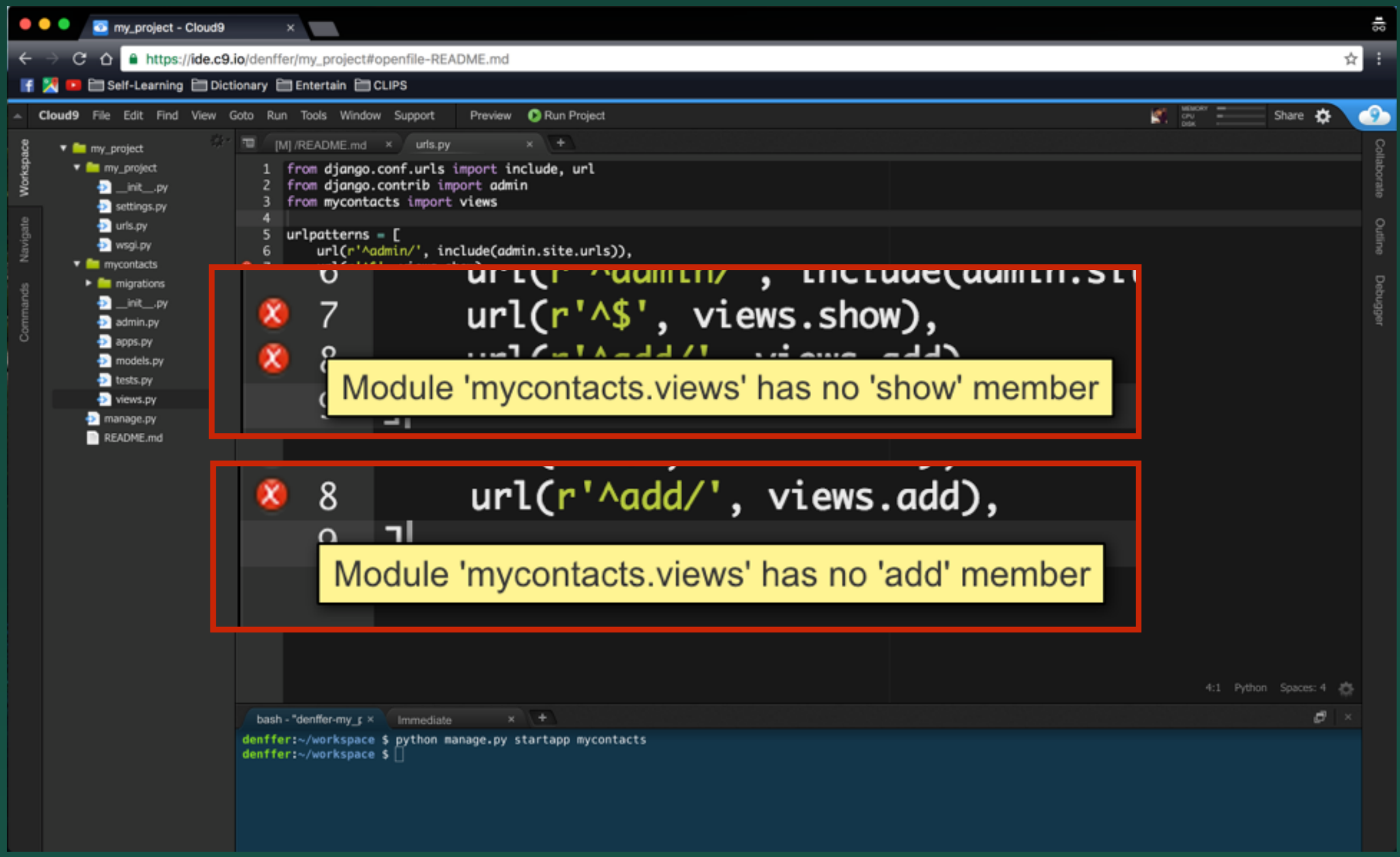
From here on, remember to save your file after every edition



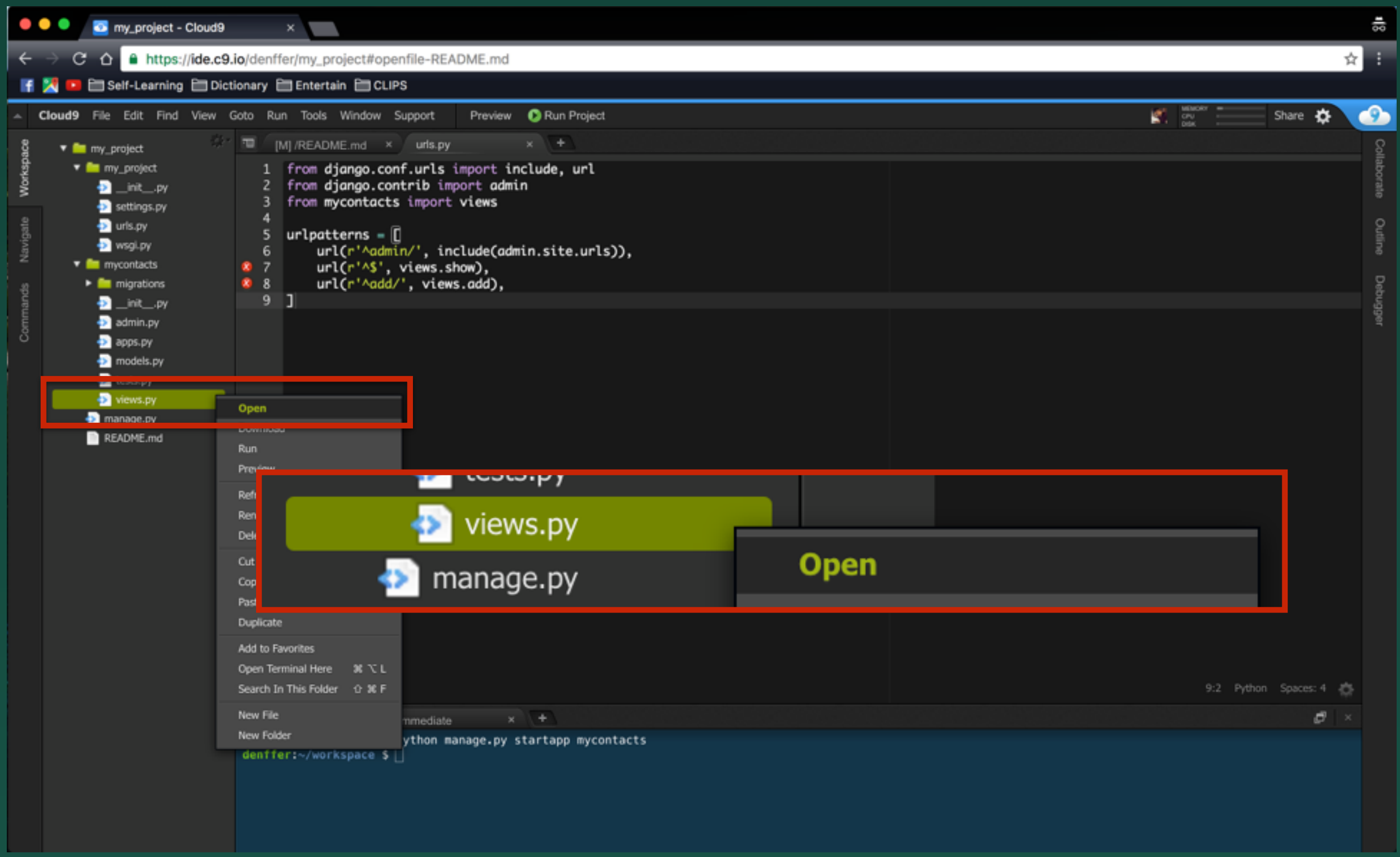
Great! The cross shown here means that you have already saved the file



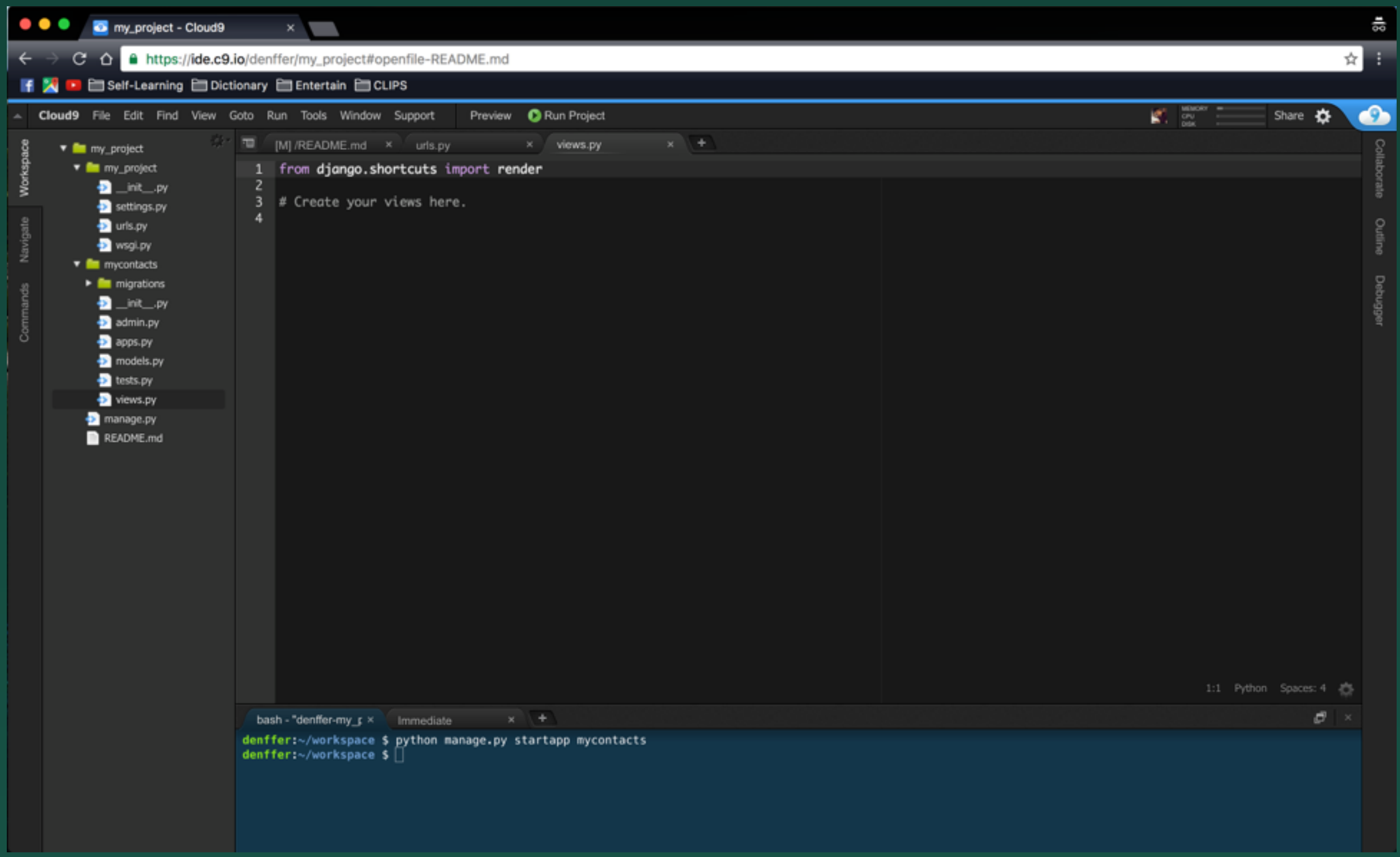
Whoops! It says there are some problems with your “View”



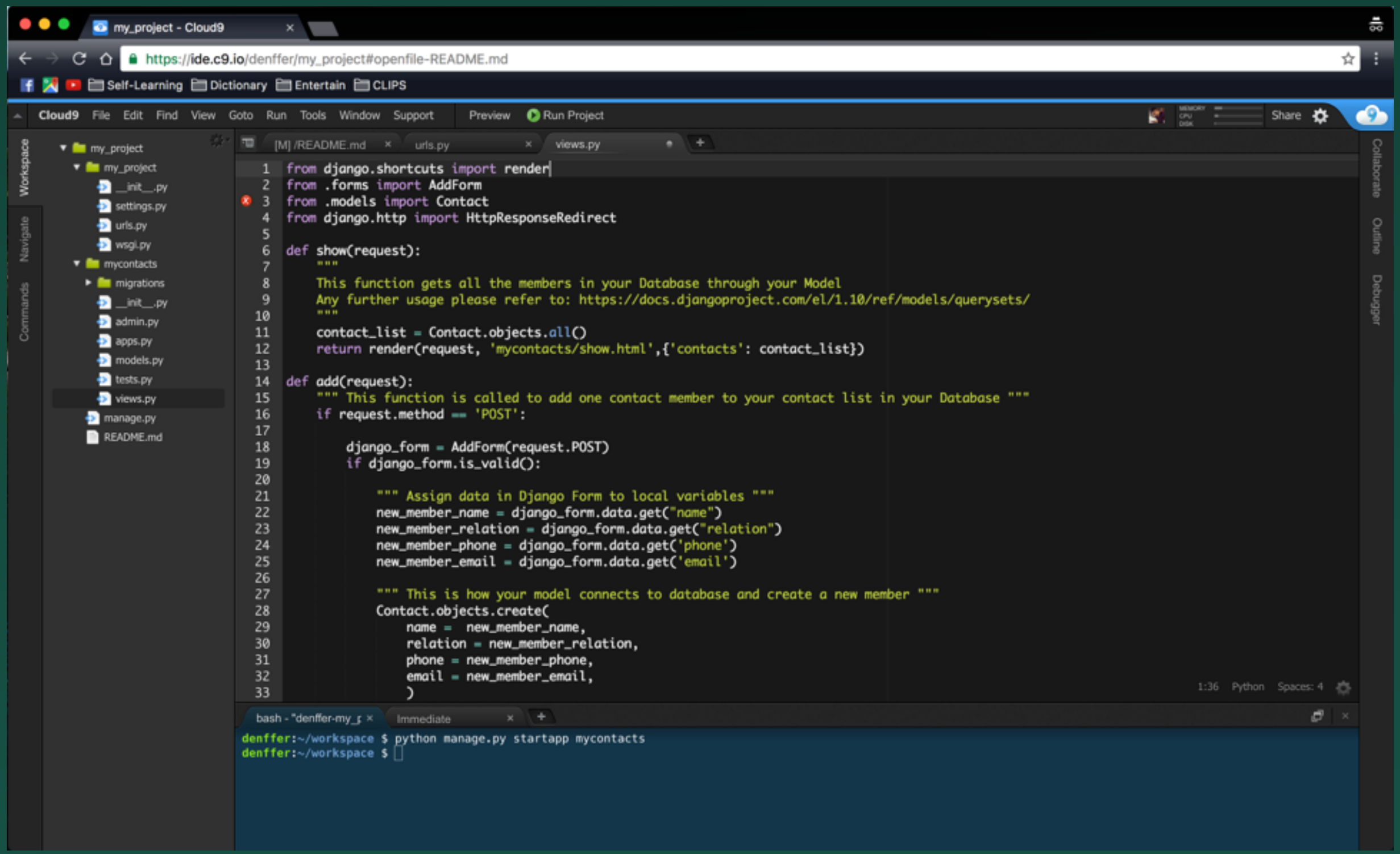
Let’s deal with it. Now go to **views.py**



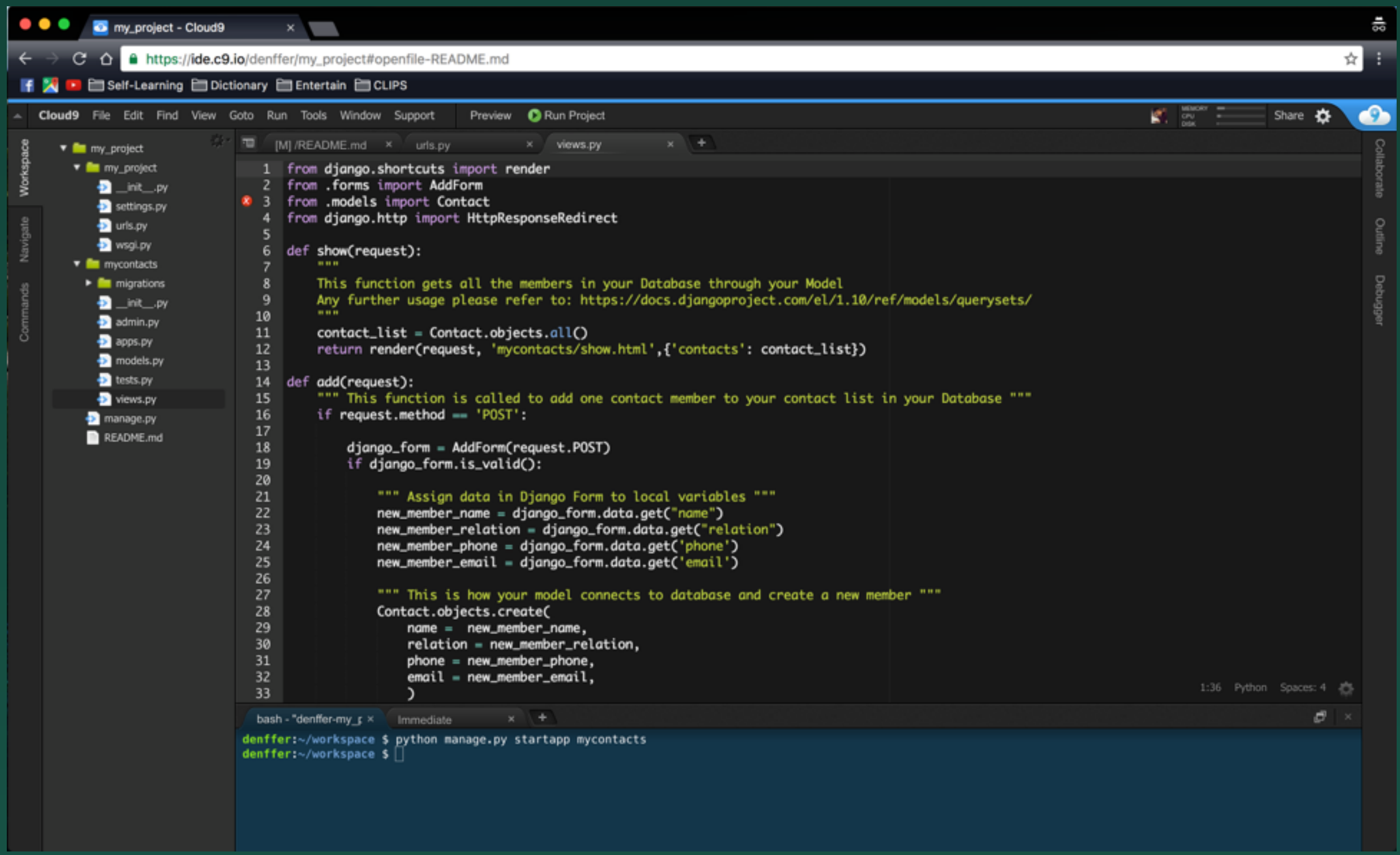
Remember? This is where you put all the functions



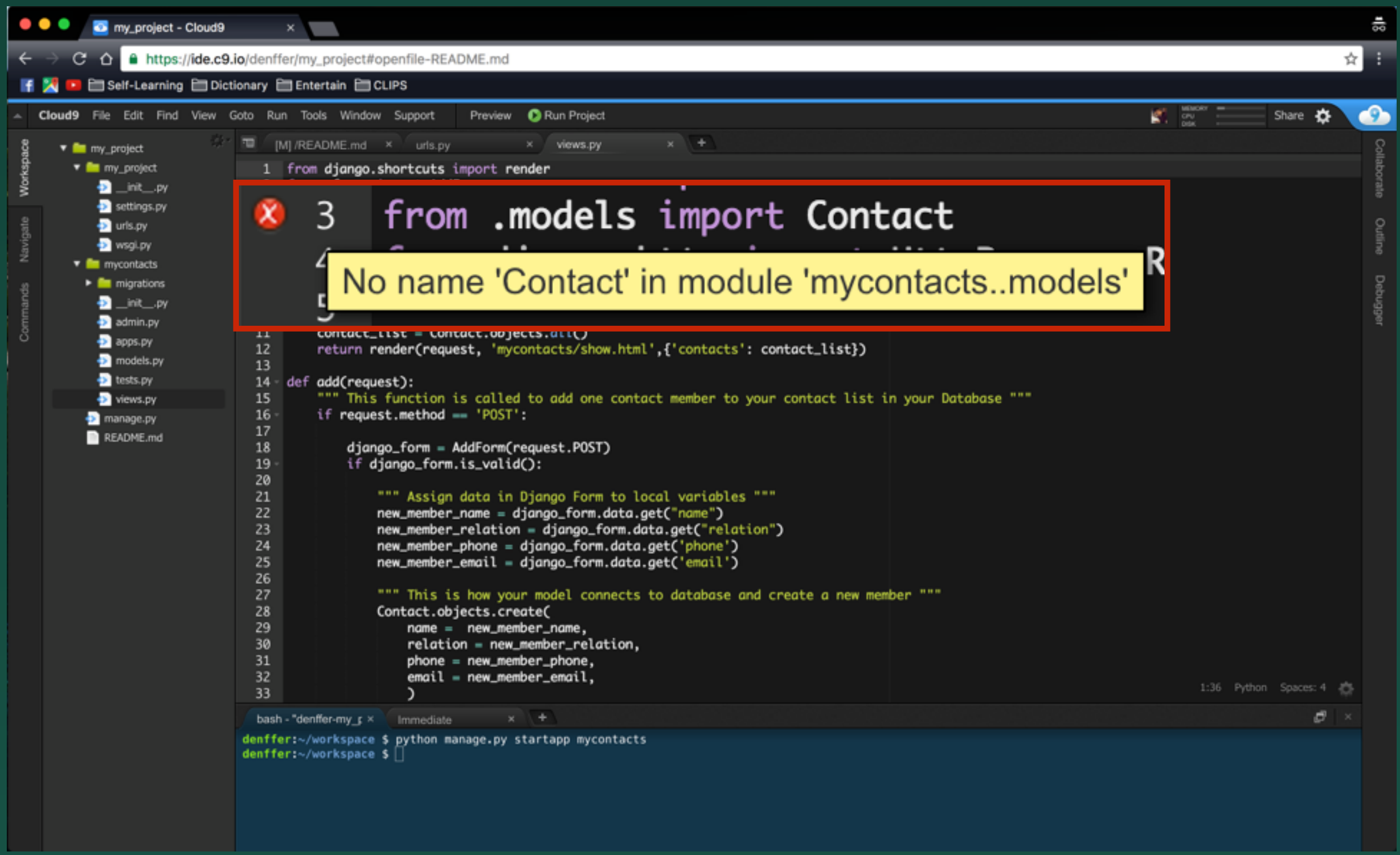
Now, copy and paste the codes in **views.py** from the repository you cloned from Github



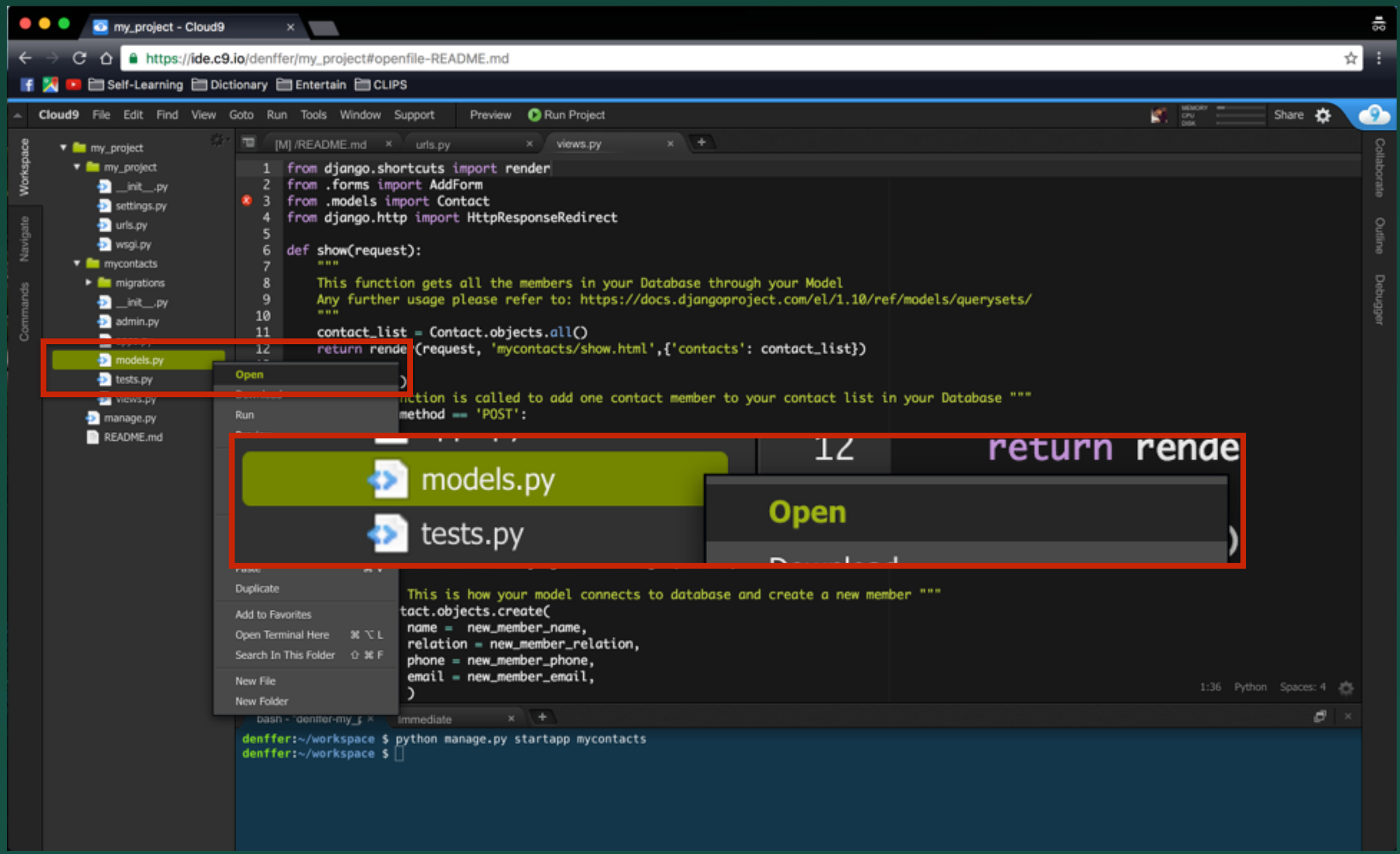
Remember to save your file



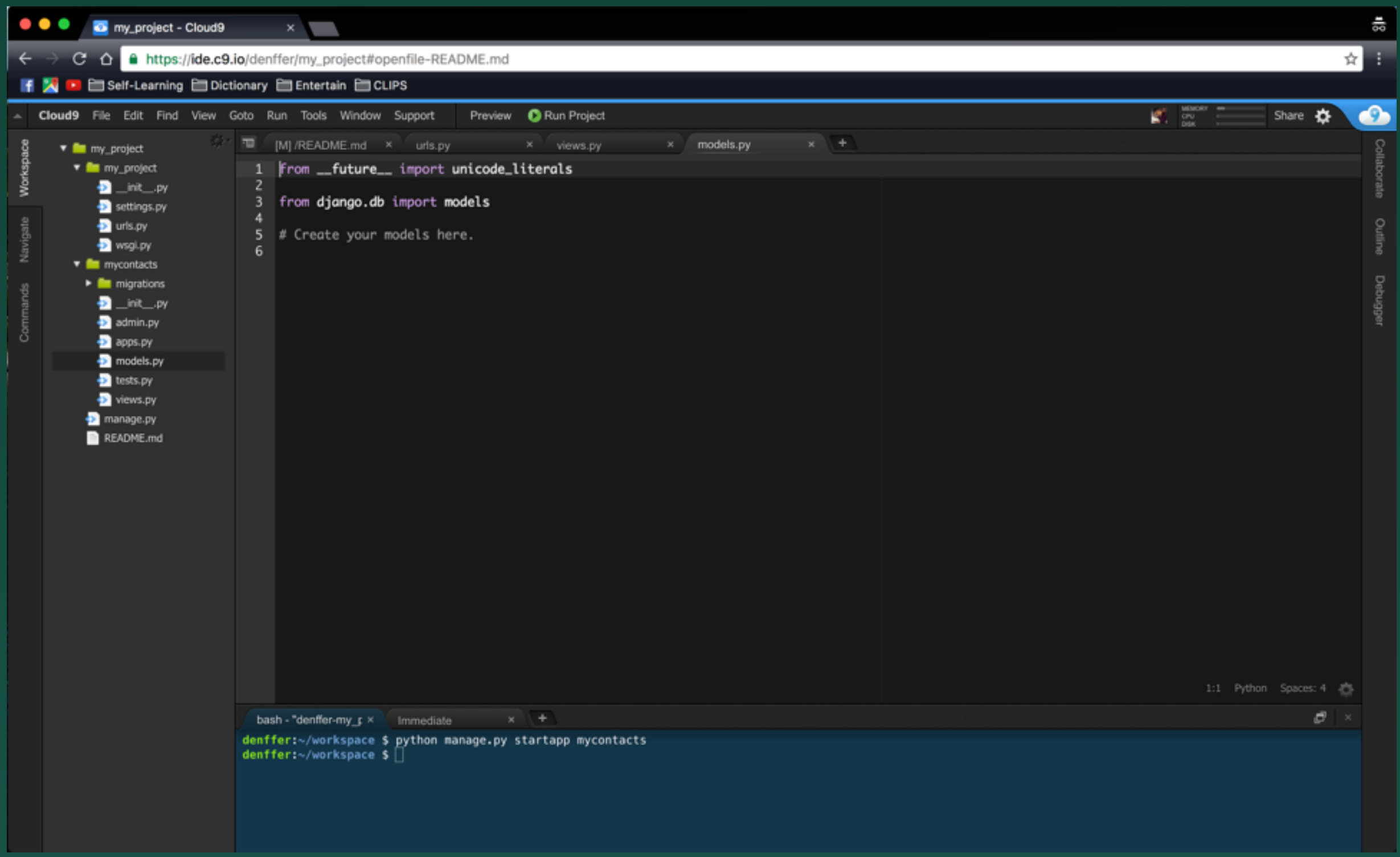
Whoops! It says there are some problems with your “Model”



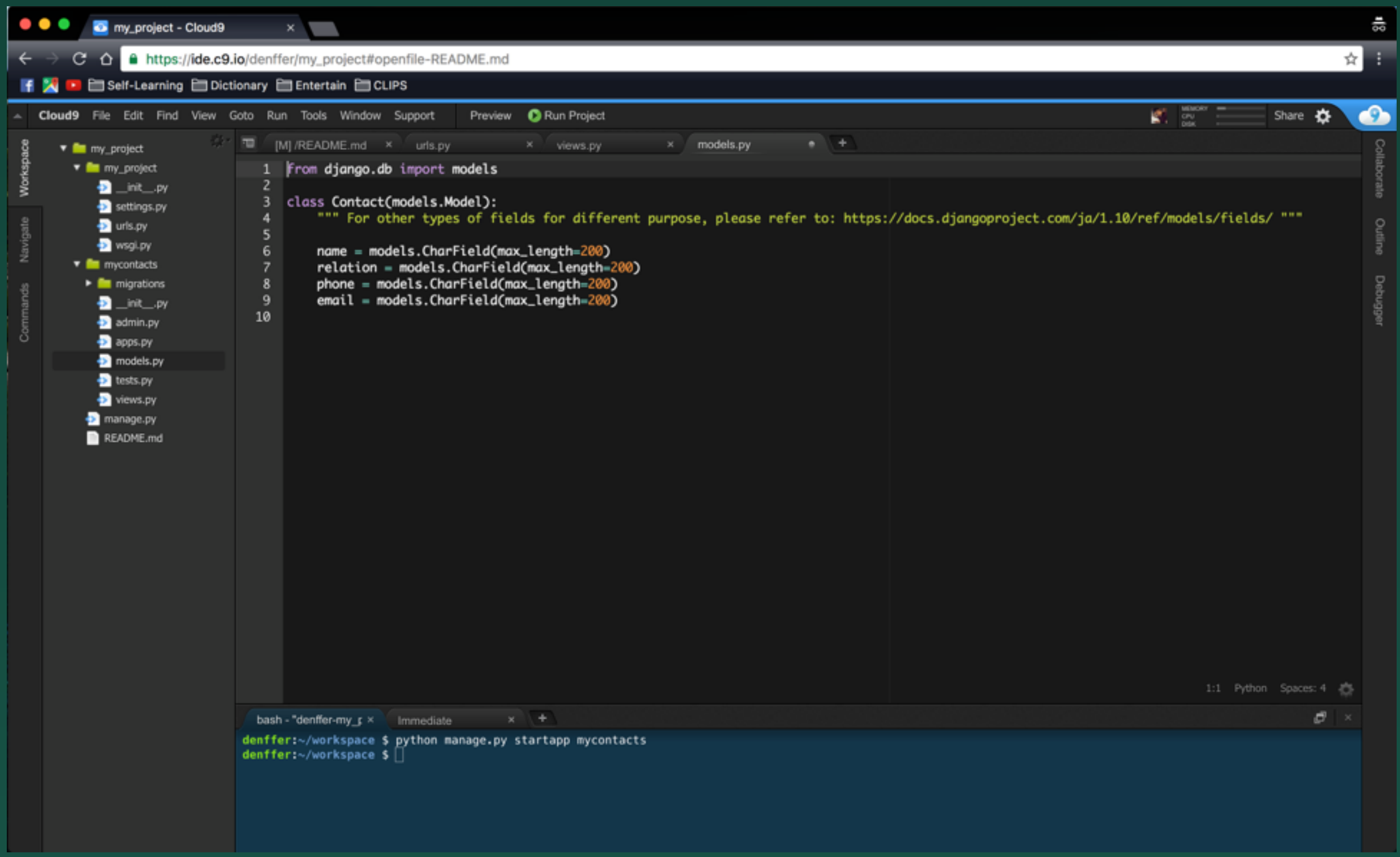
Let’s deal with it. Now go to **models.py**



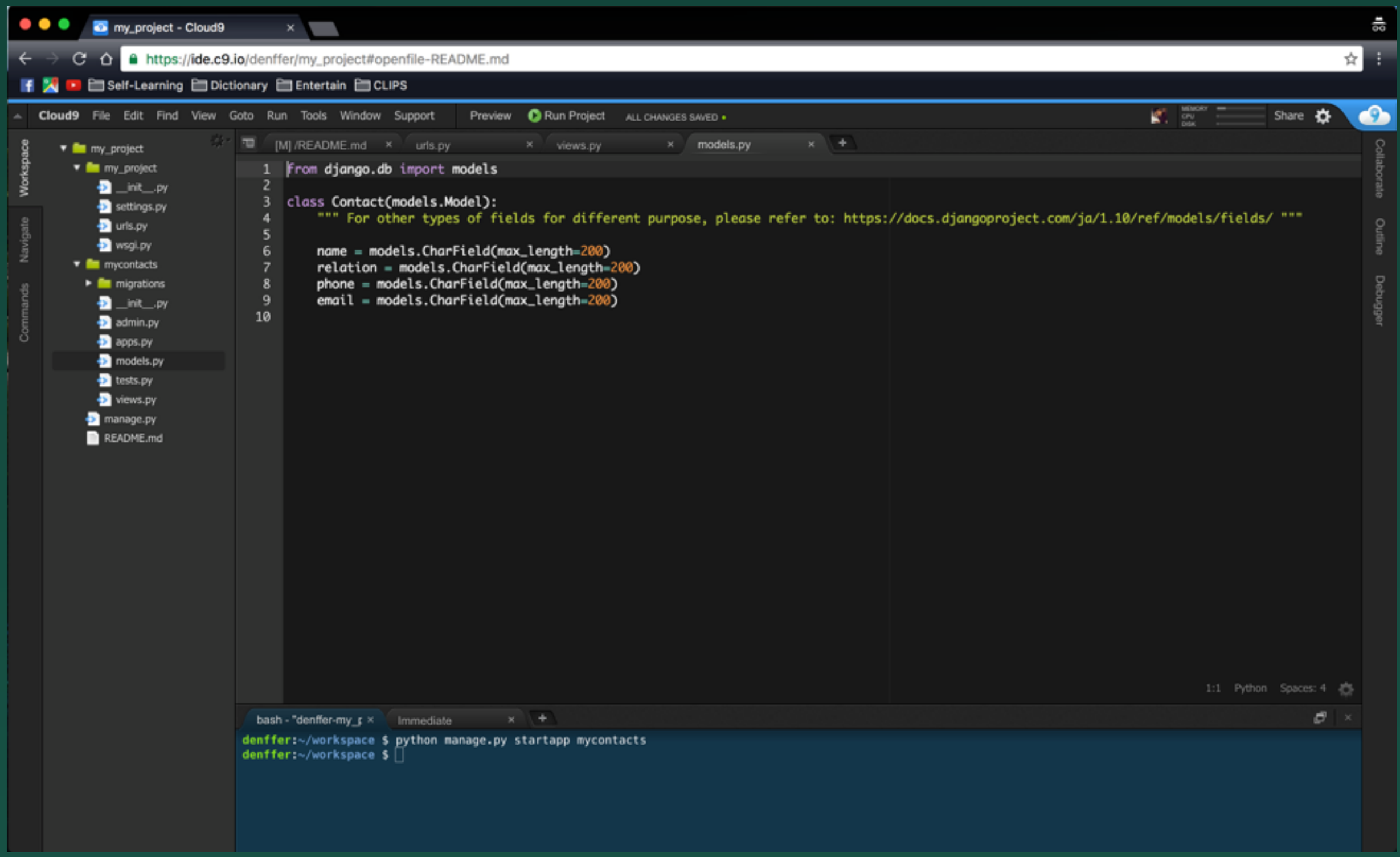
Remember? You have to construct your model so it can really connects to your database



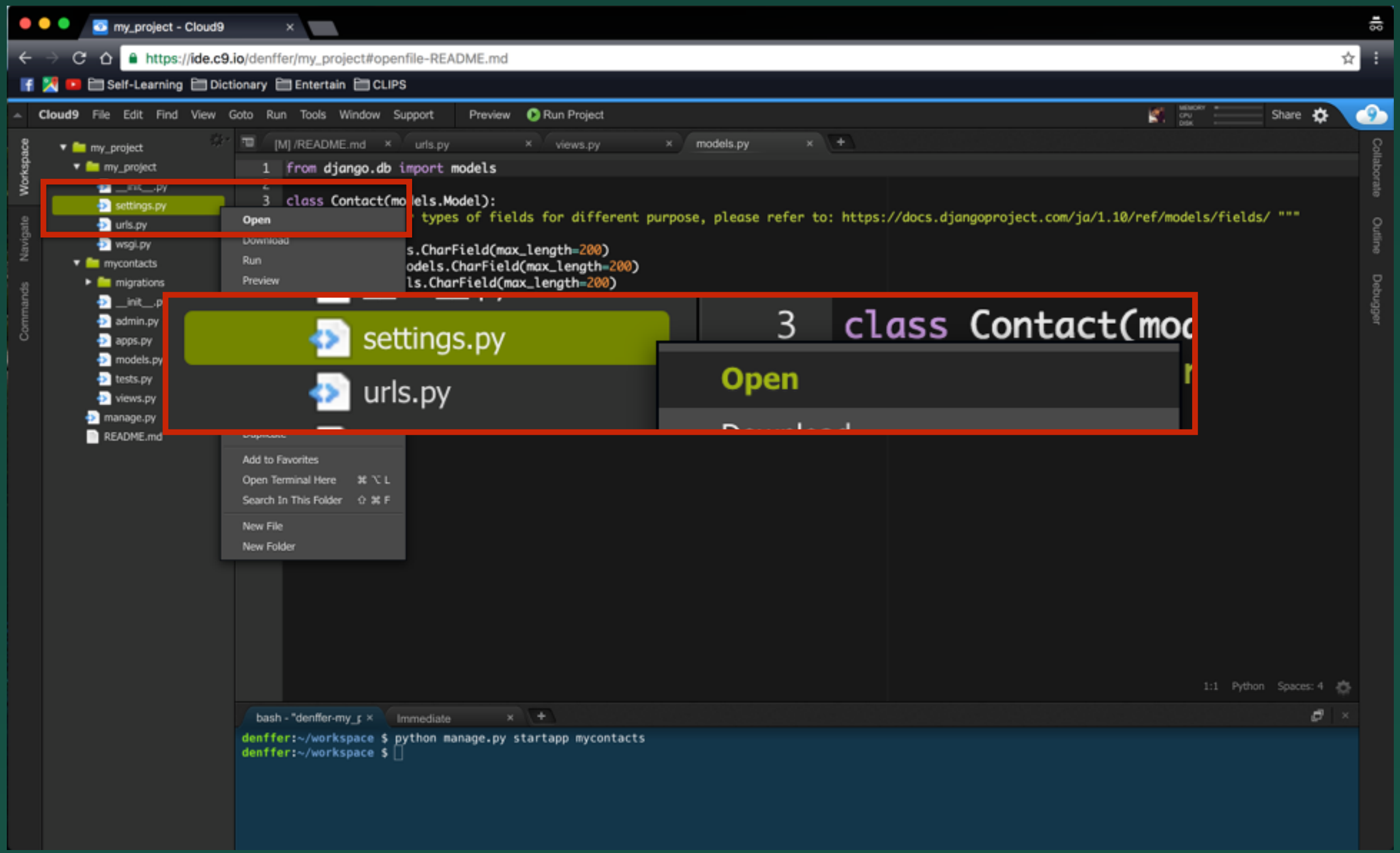
Now, copy and paste the codes in **models.py** from the repository you cloned from Github



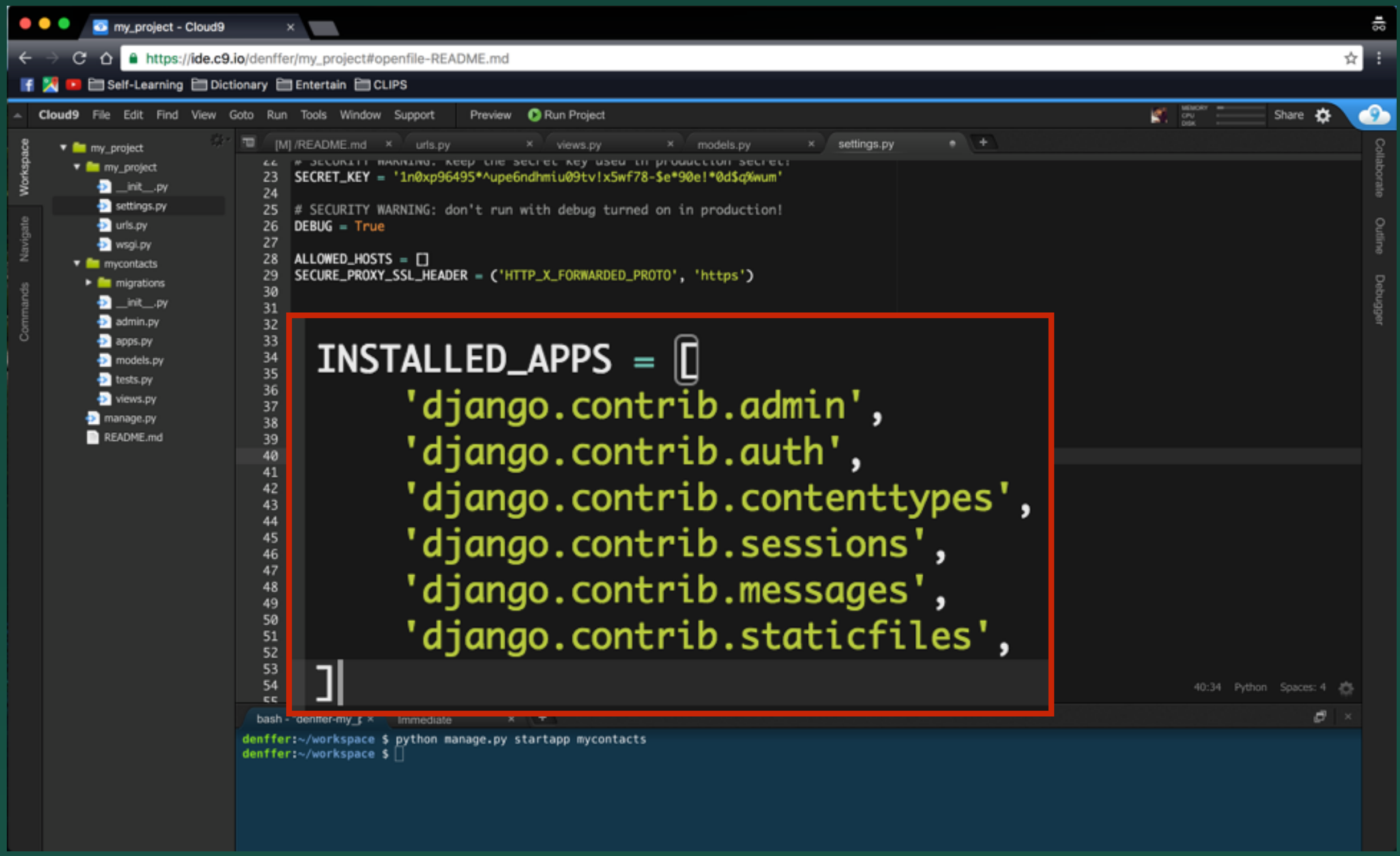
Remember to save your file



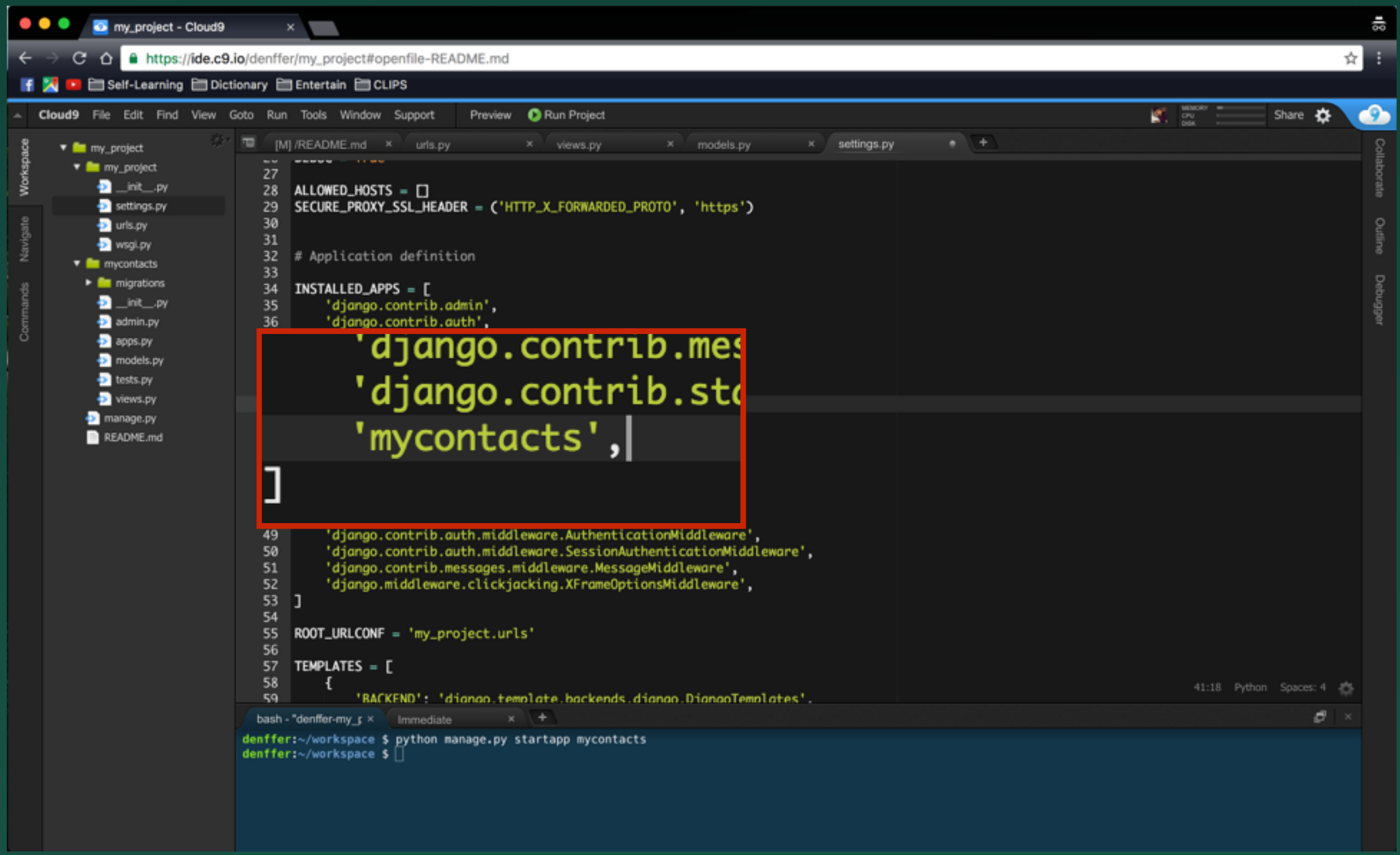
Let’s go to **settings.py**



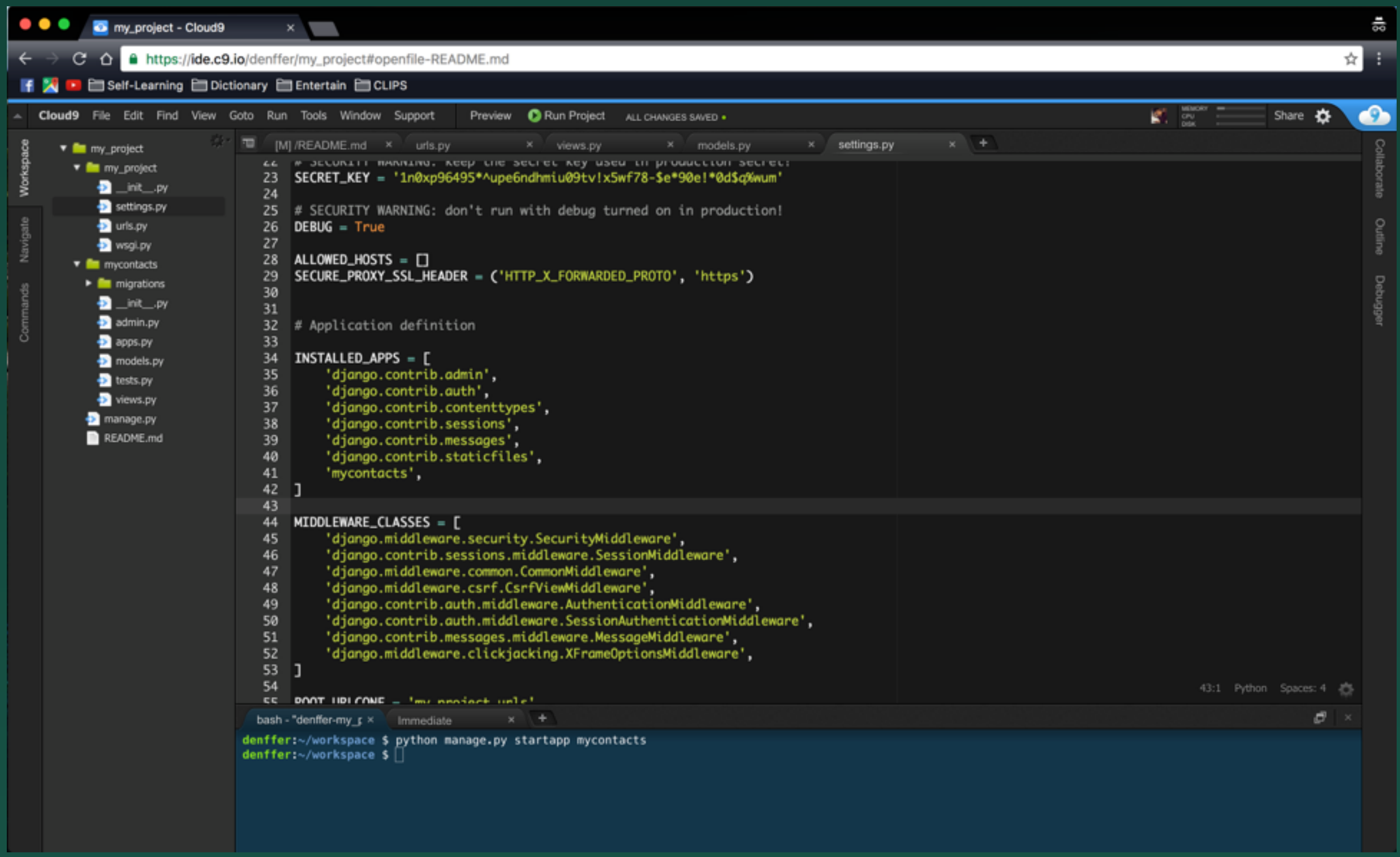
Find ‘INSTALLED_APPS’



State your application in INSTALLED_APPS



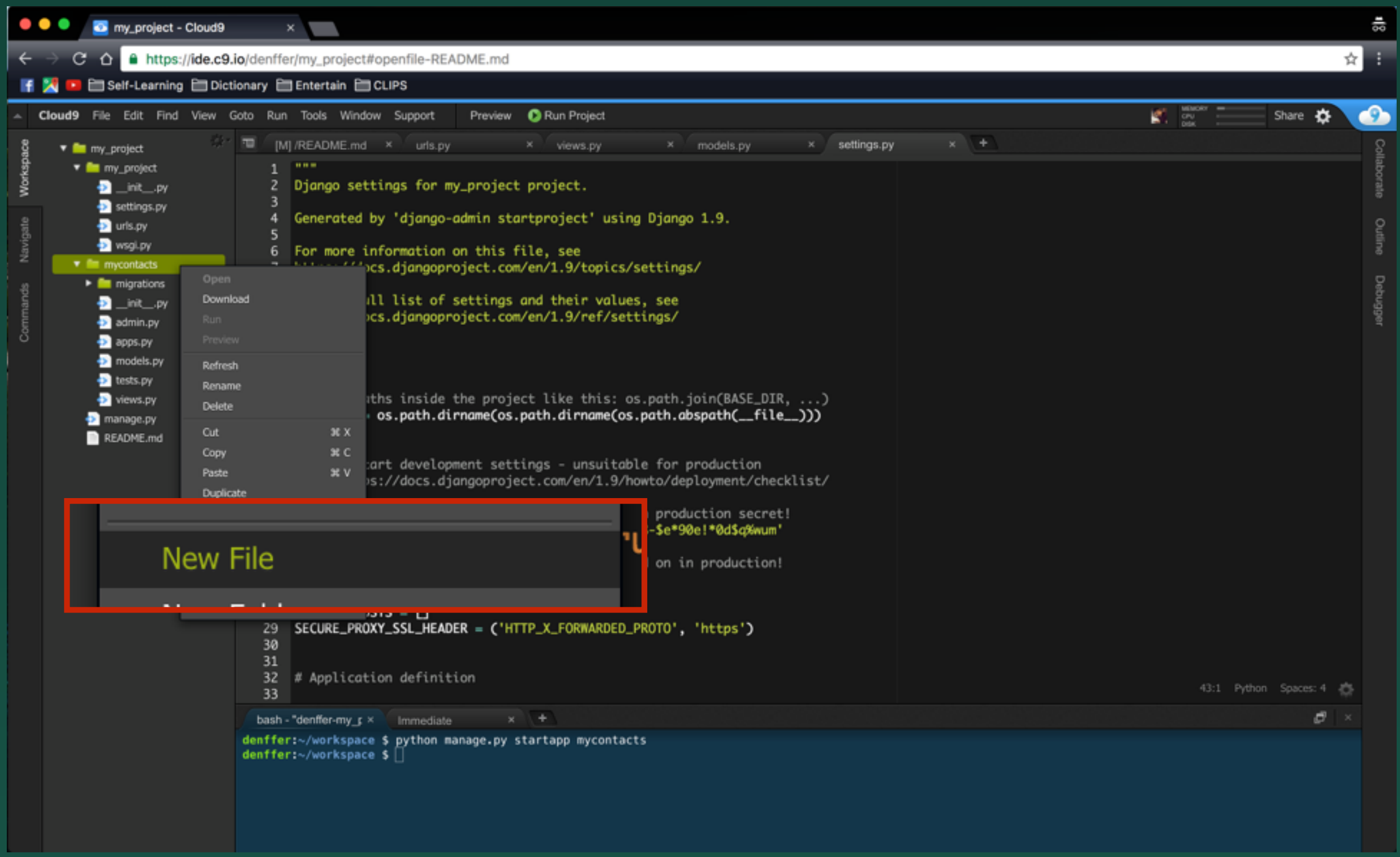
Remember to save your file



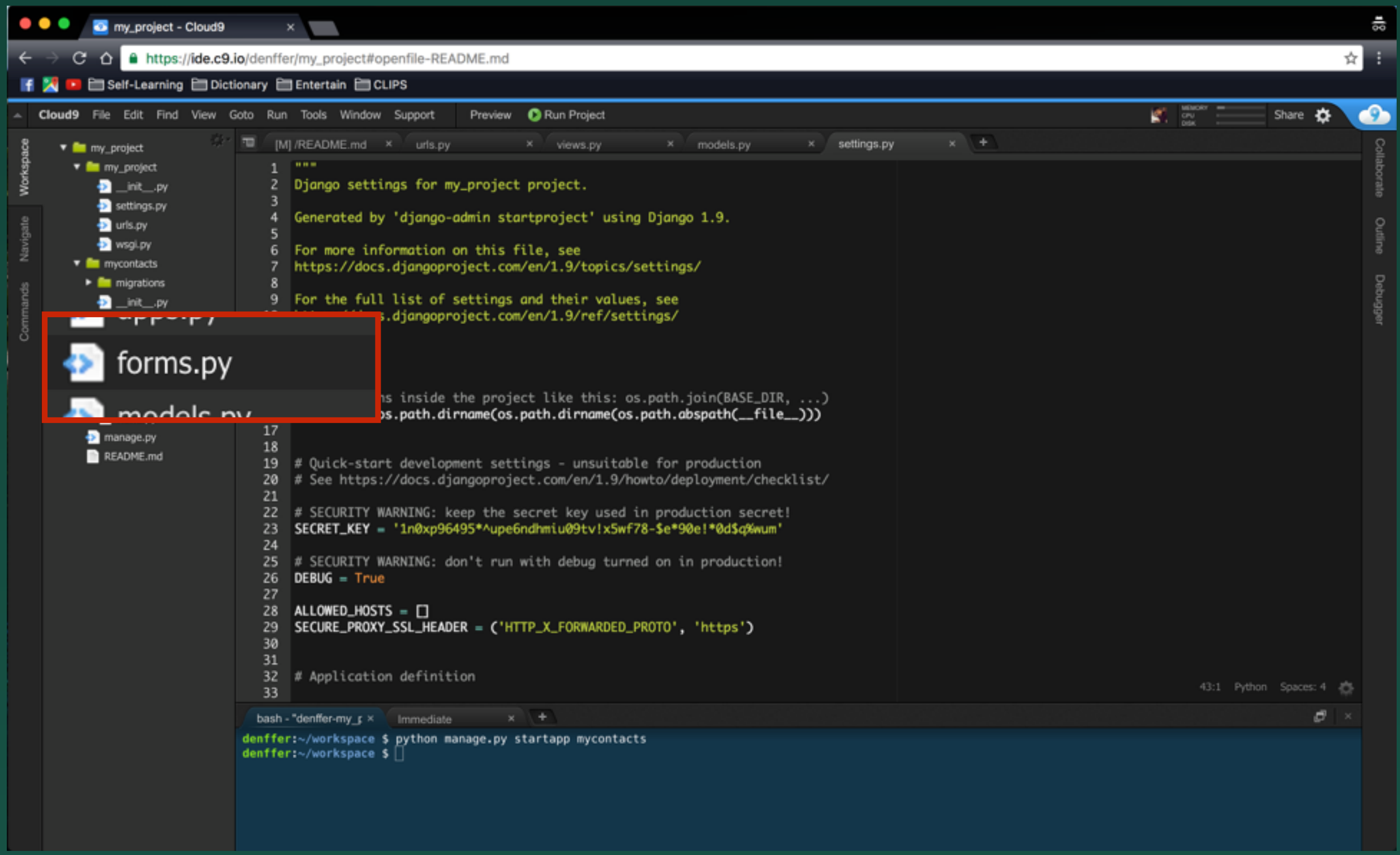
Django does not provide **Form** by default

So you have to create it by yourself

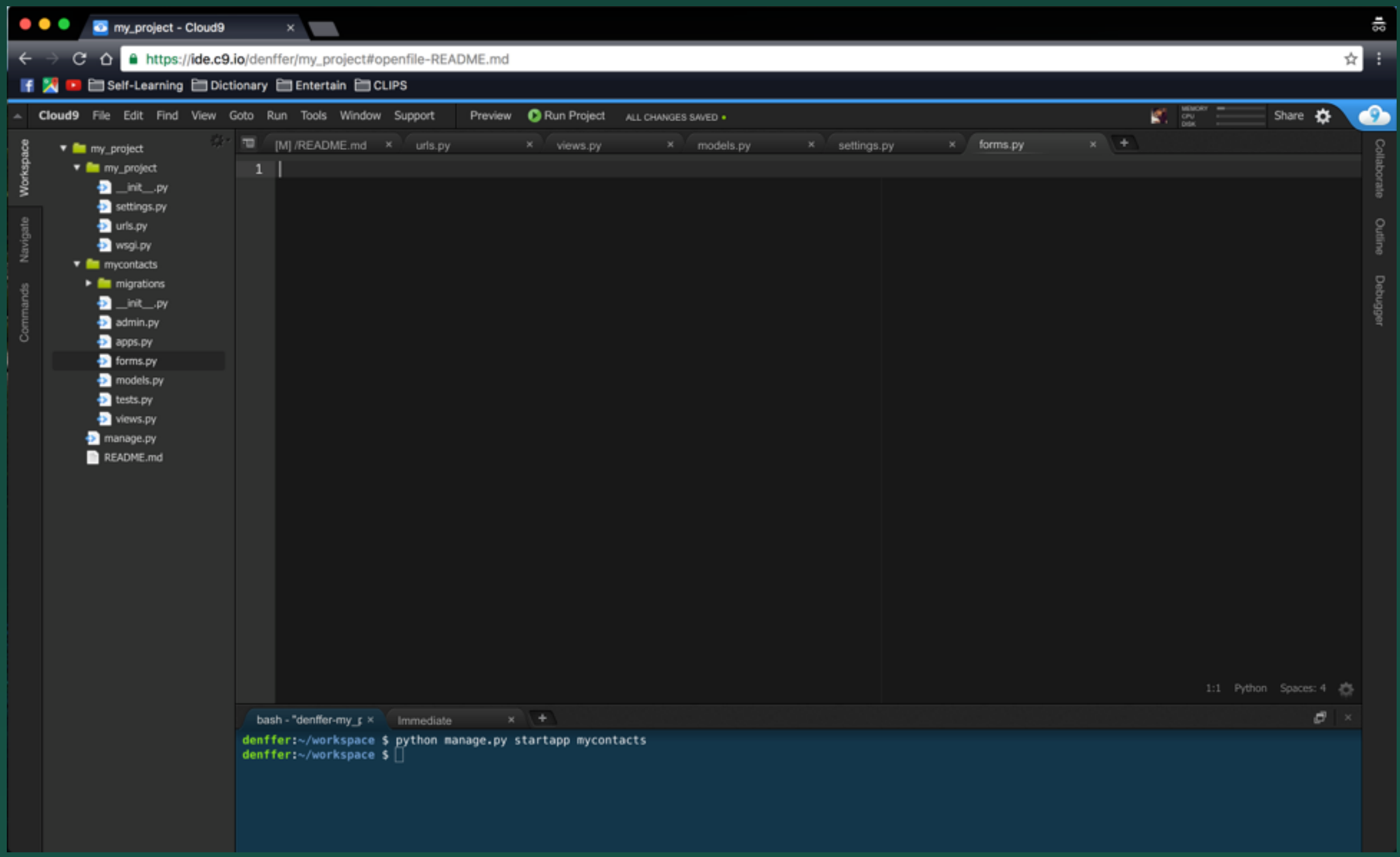
Create a file named ‘forms.py’ in your application ‘mycontacts’



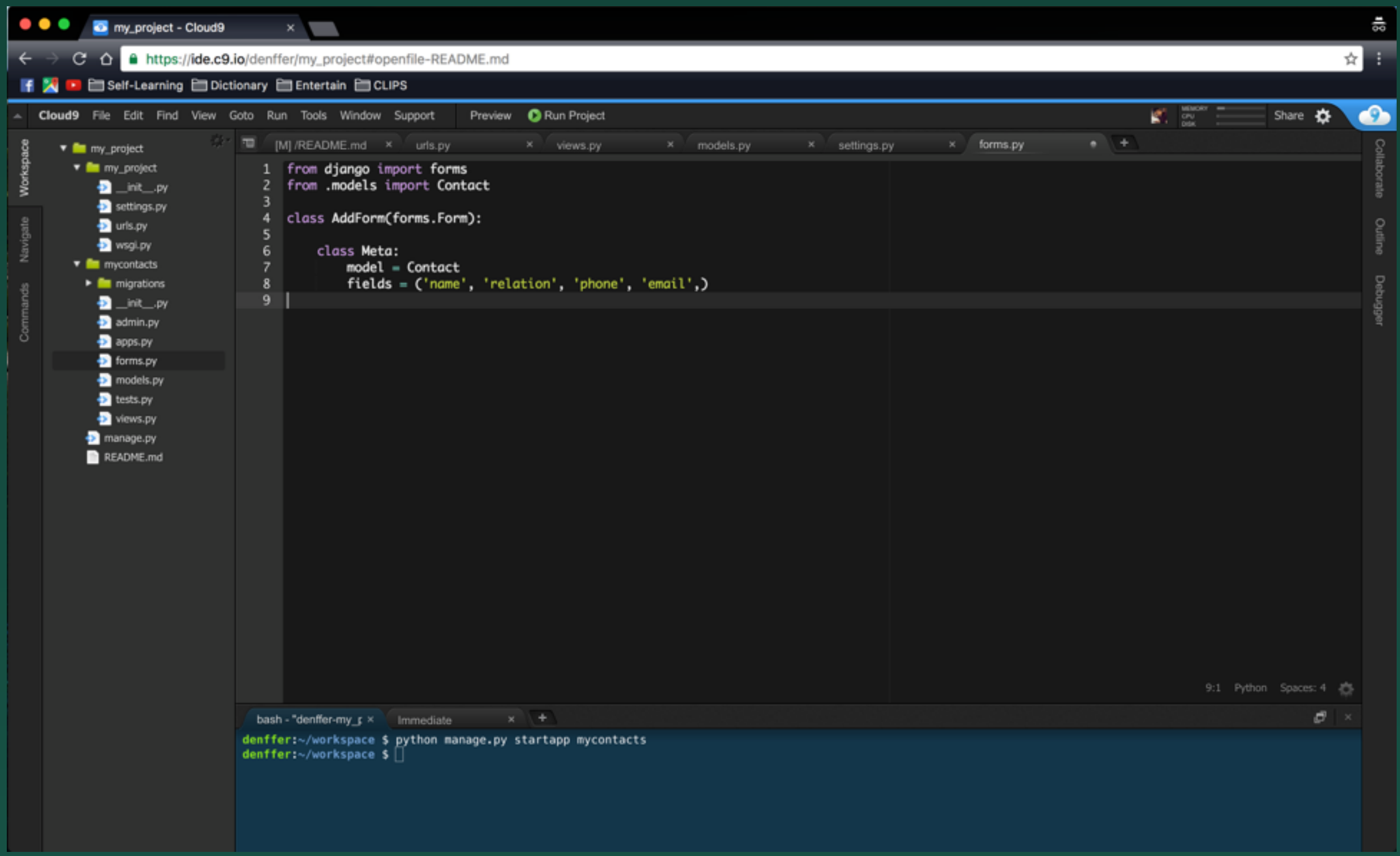
There you go. Open it.



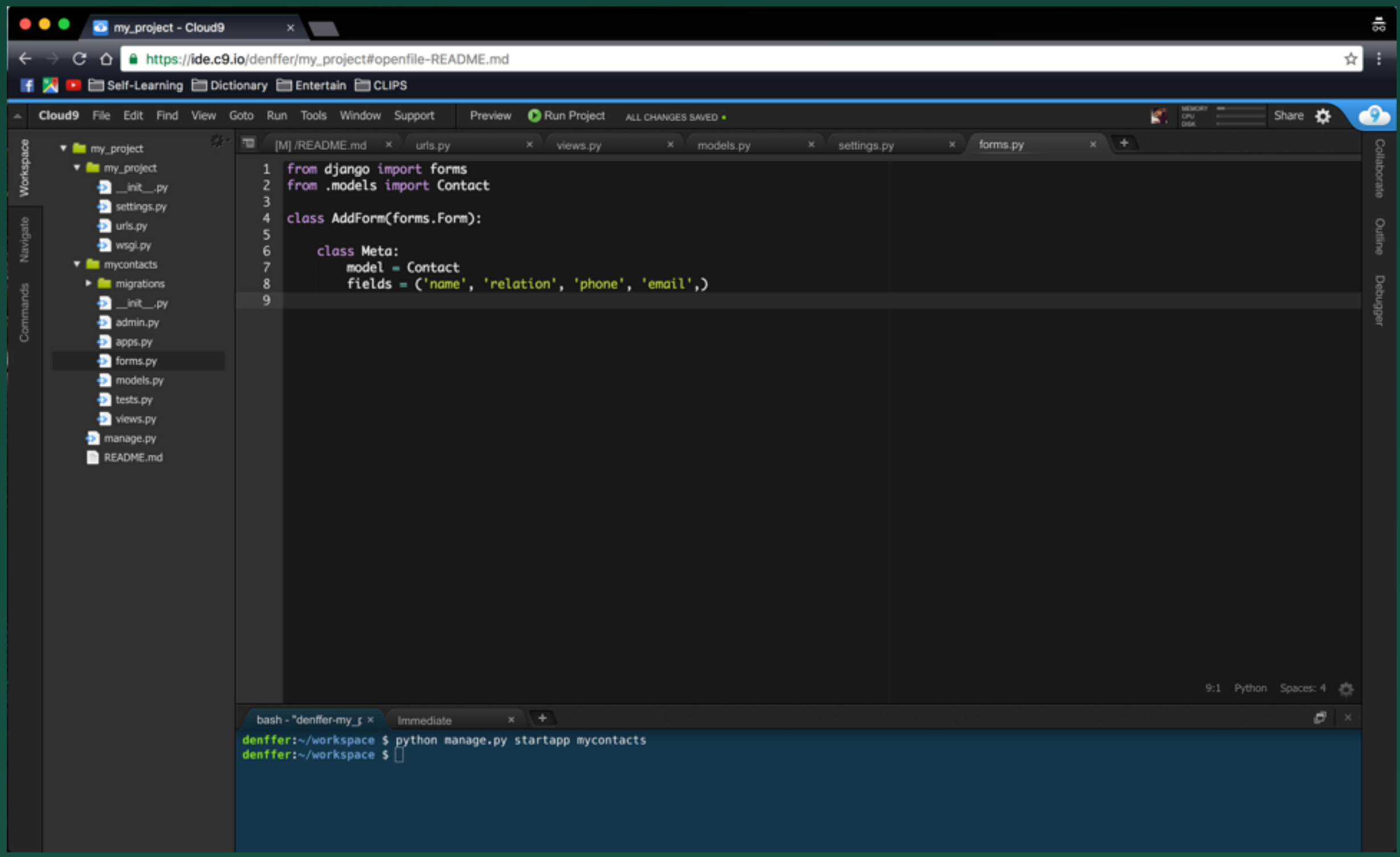
Remember? Django Form is used to fetch data from Html Form



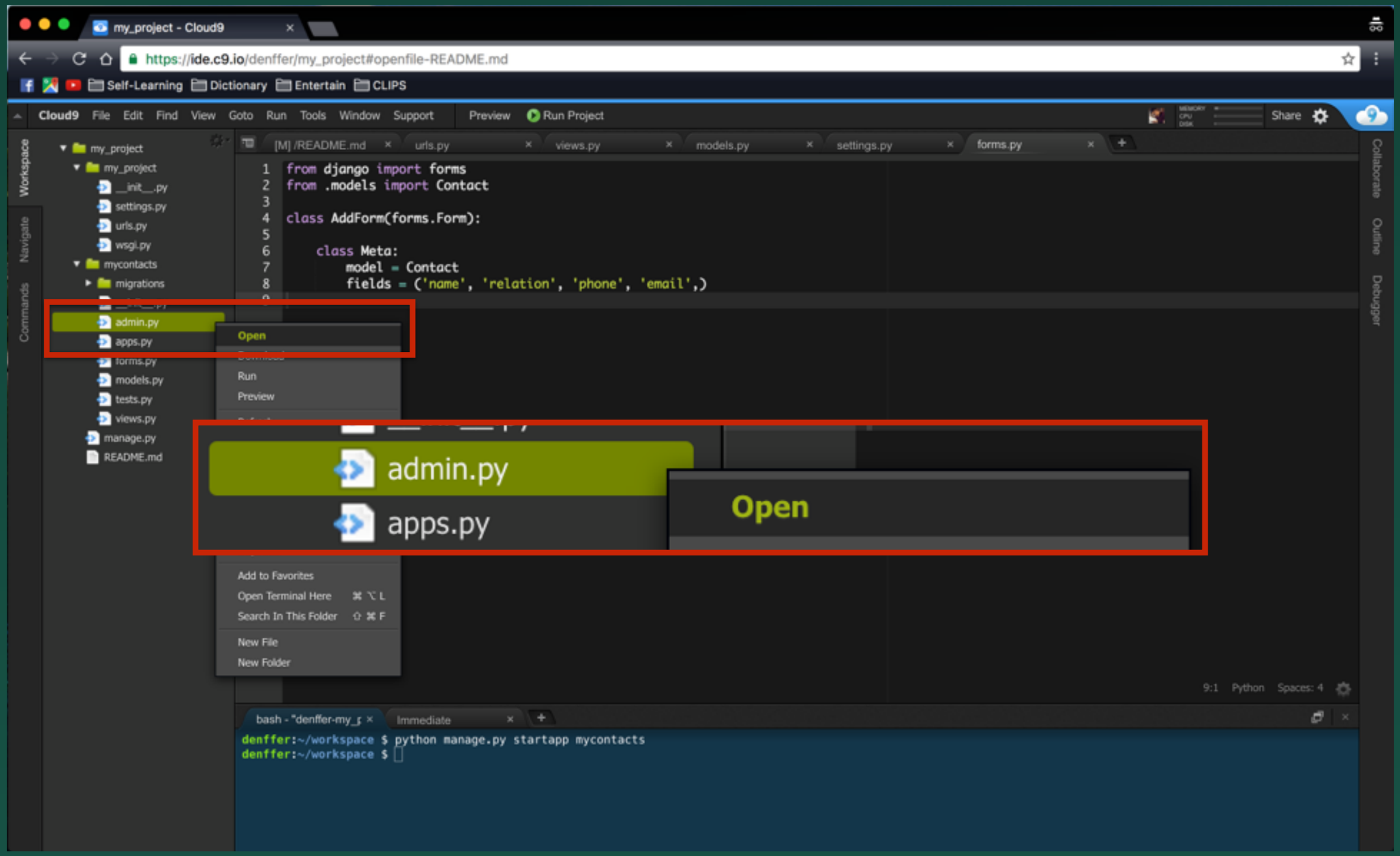
Now, copy and paste the codes in **forms.py** from the repository you cloned from Github



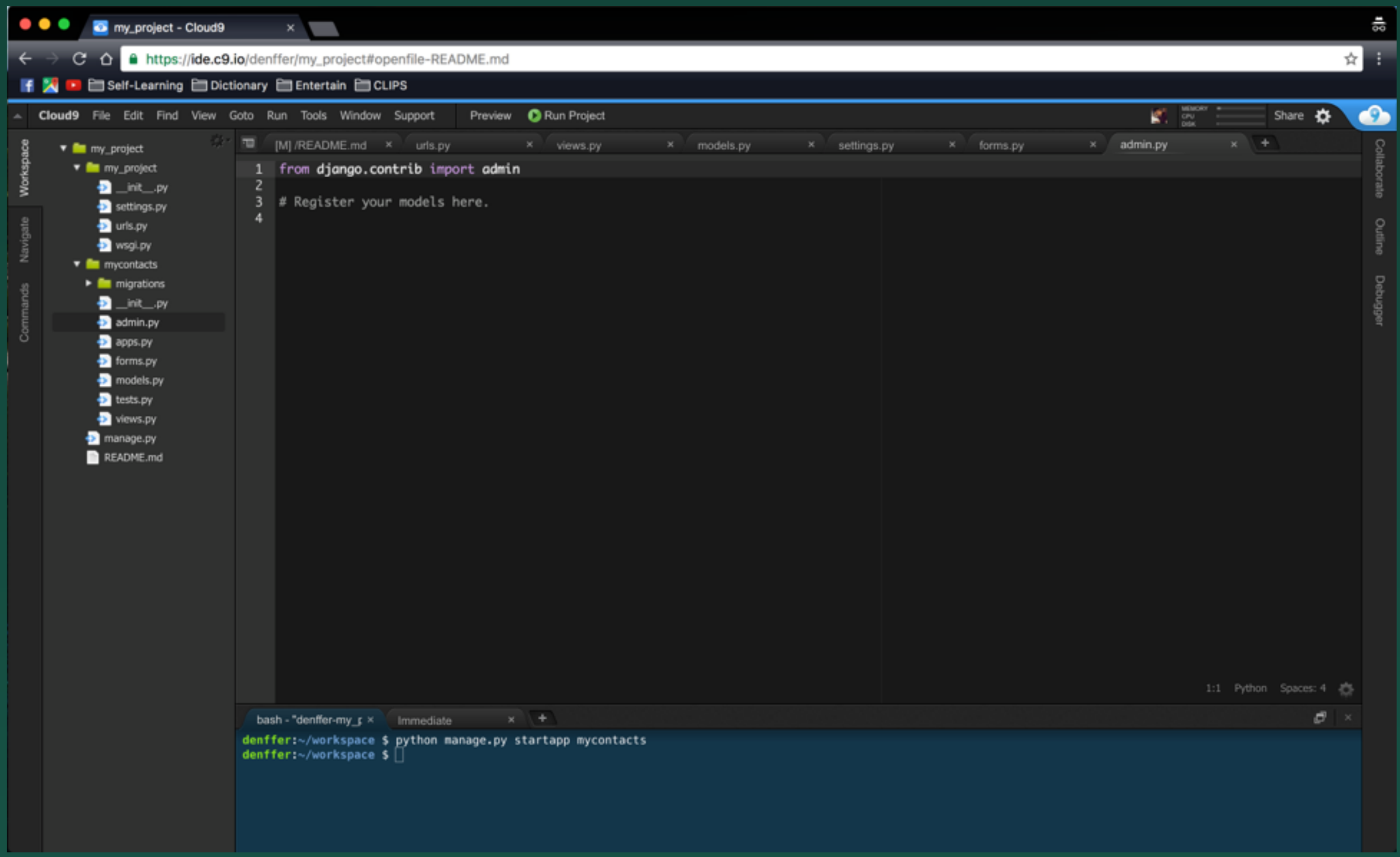
Remember to save your file



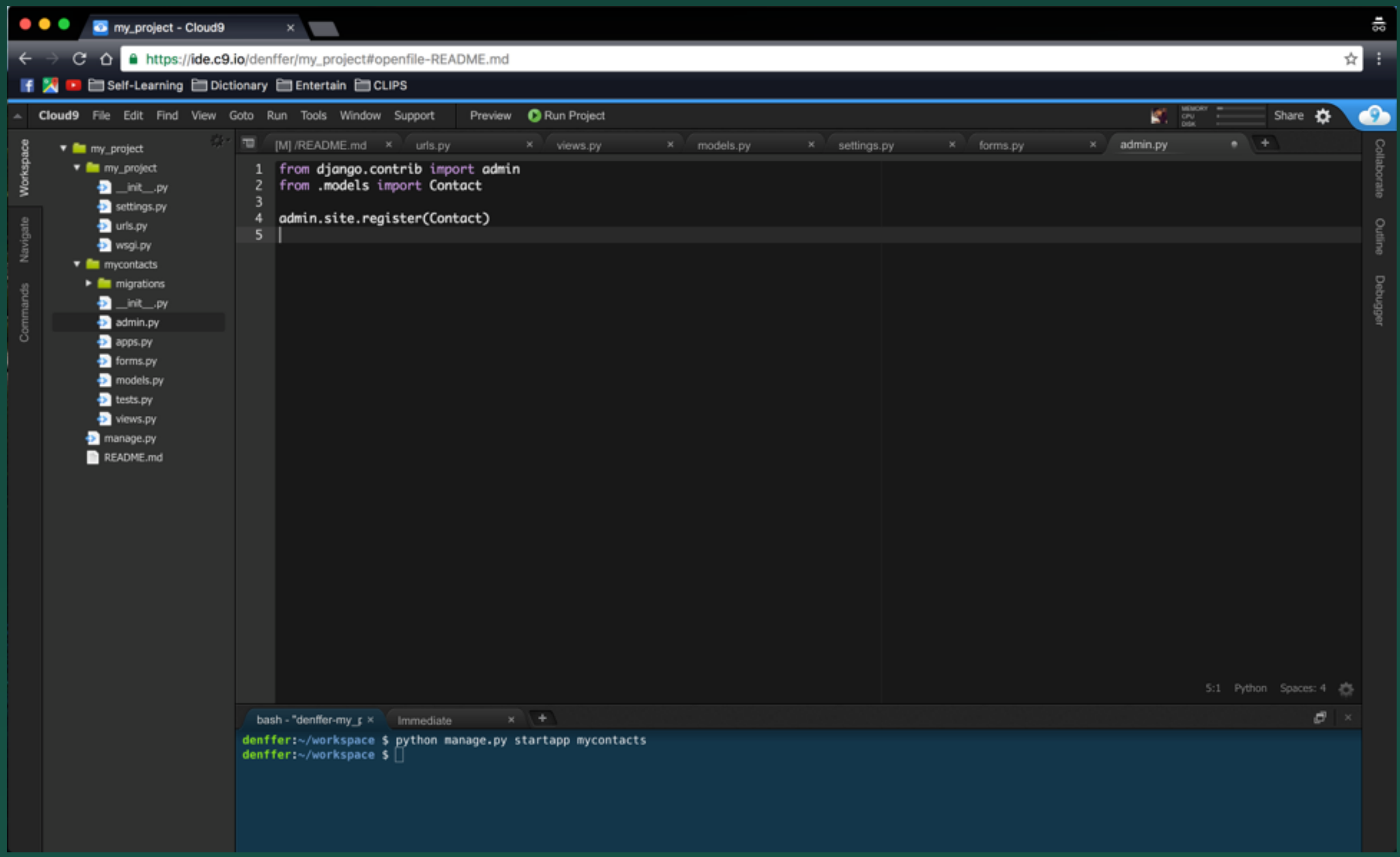
One more step. Let's go to **admin.py**



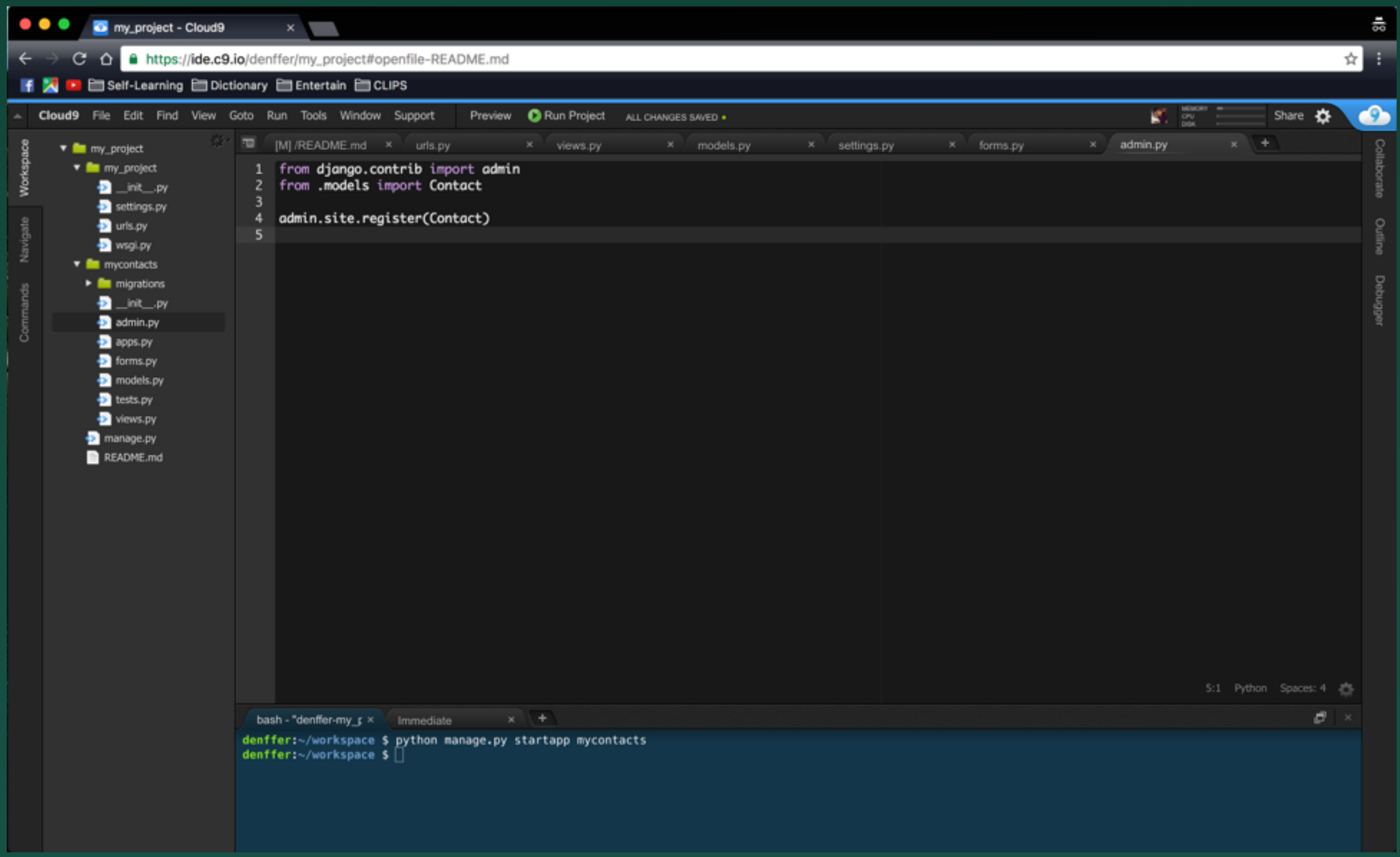
Remember? **Admin** is used to register your **Model** in your **Database**



Now, copy and paste the codes in **admin.py** from the repository you cloned from Github



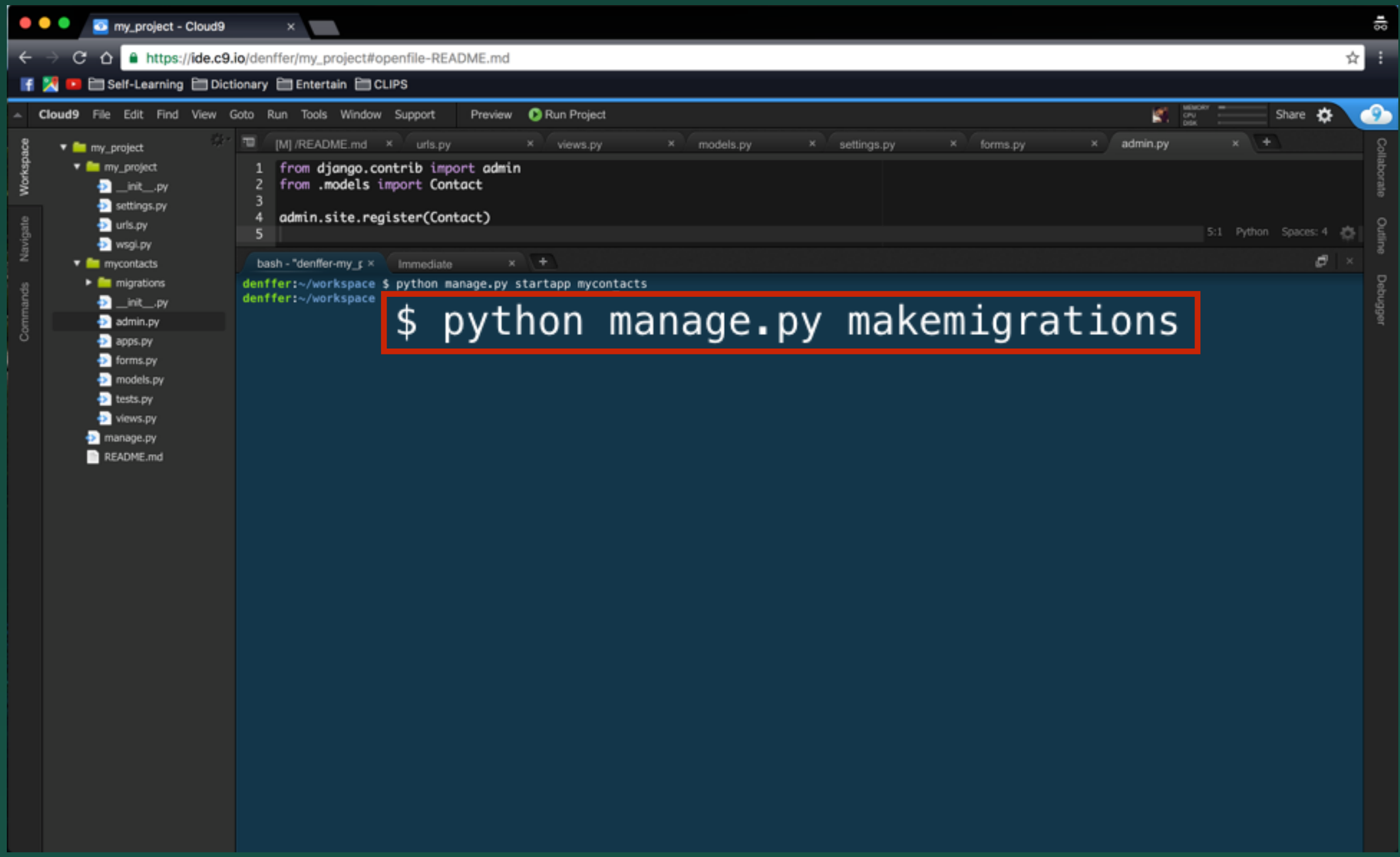
Remember to save your file



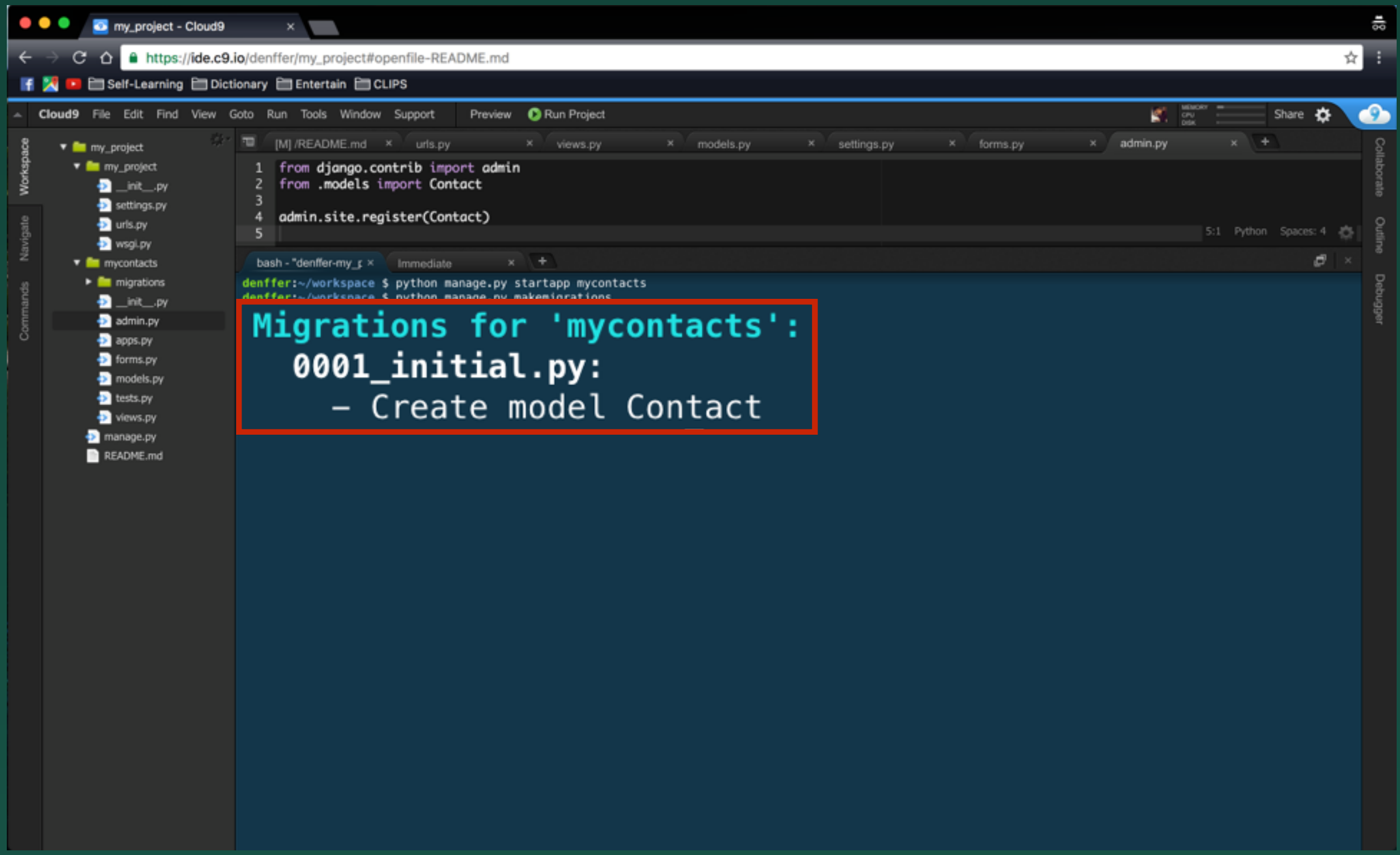
Great!

Now you have to make a migration so that your **Model** and be recognized and stored in **Database**

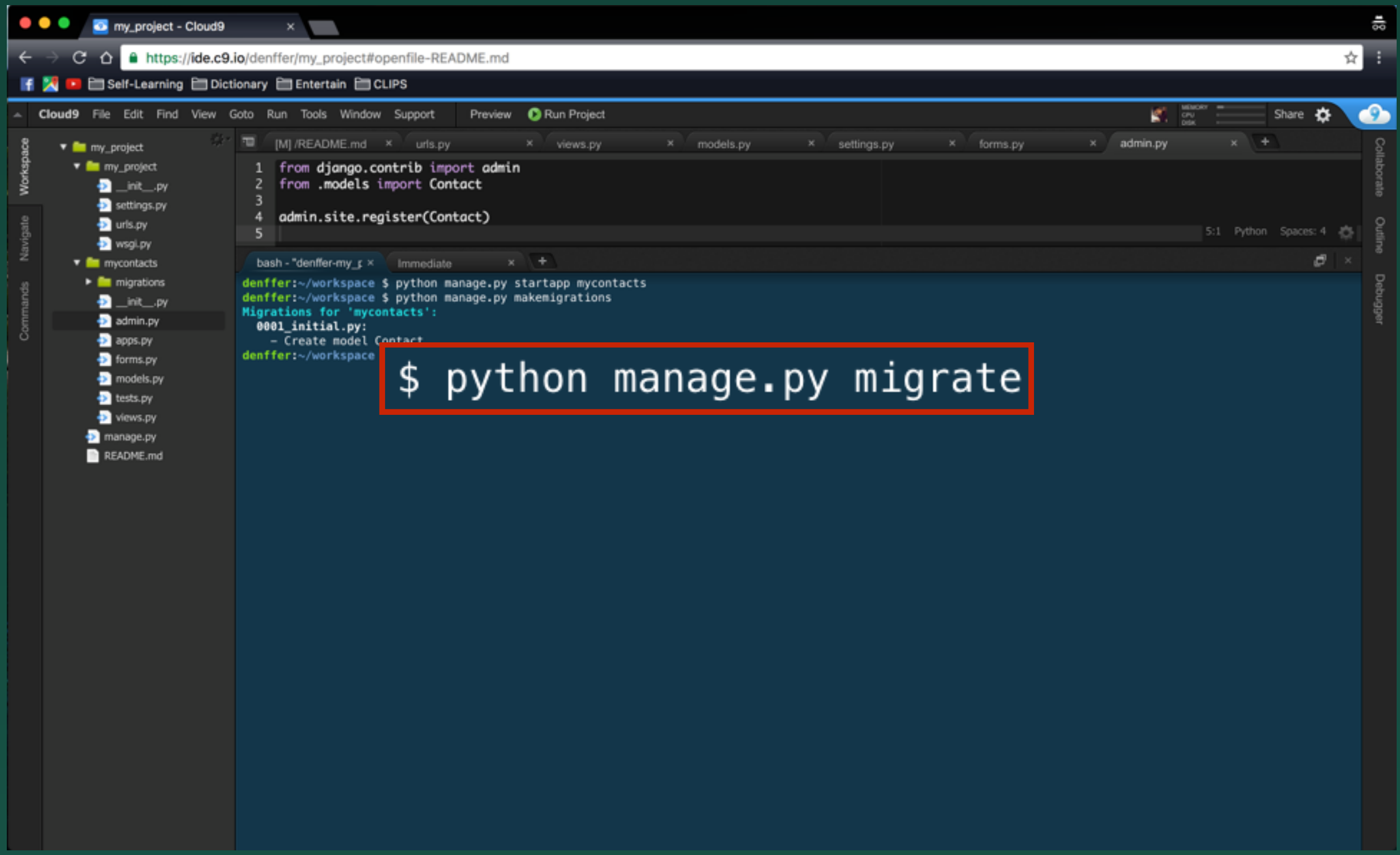
Go to command line and type in the command



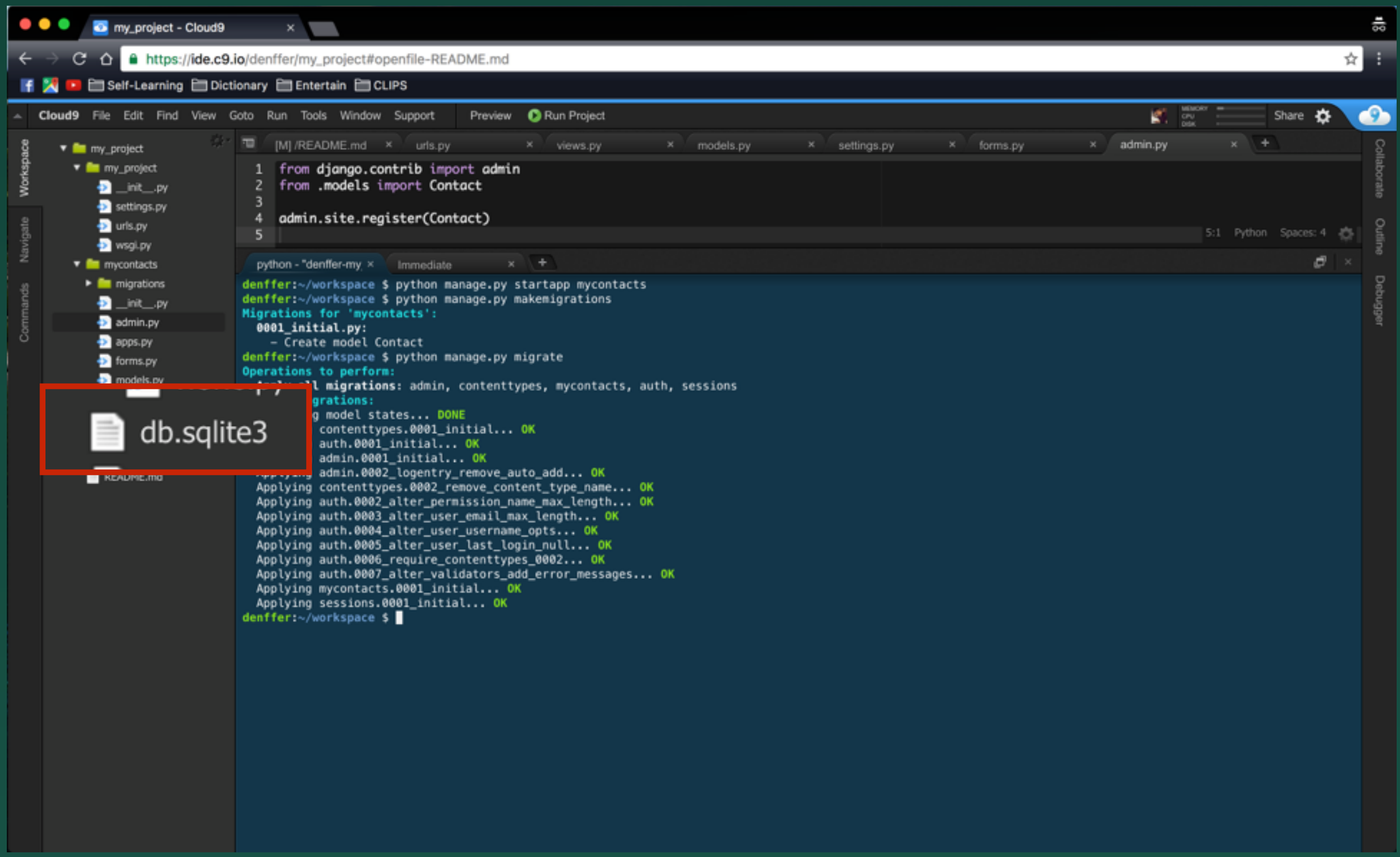
Great! You have just created your first model



Now, type in the command to write your model into your **Database**



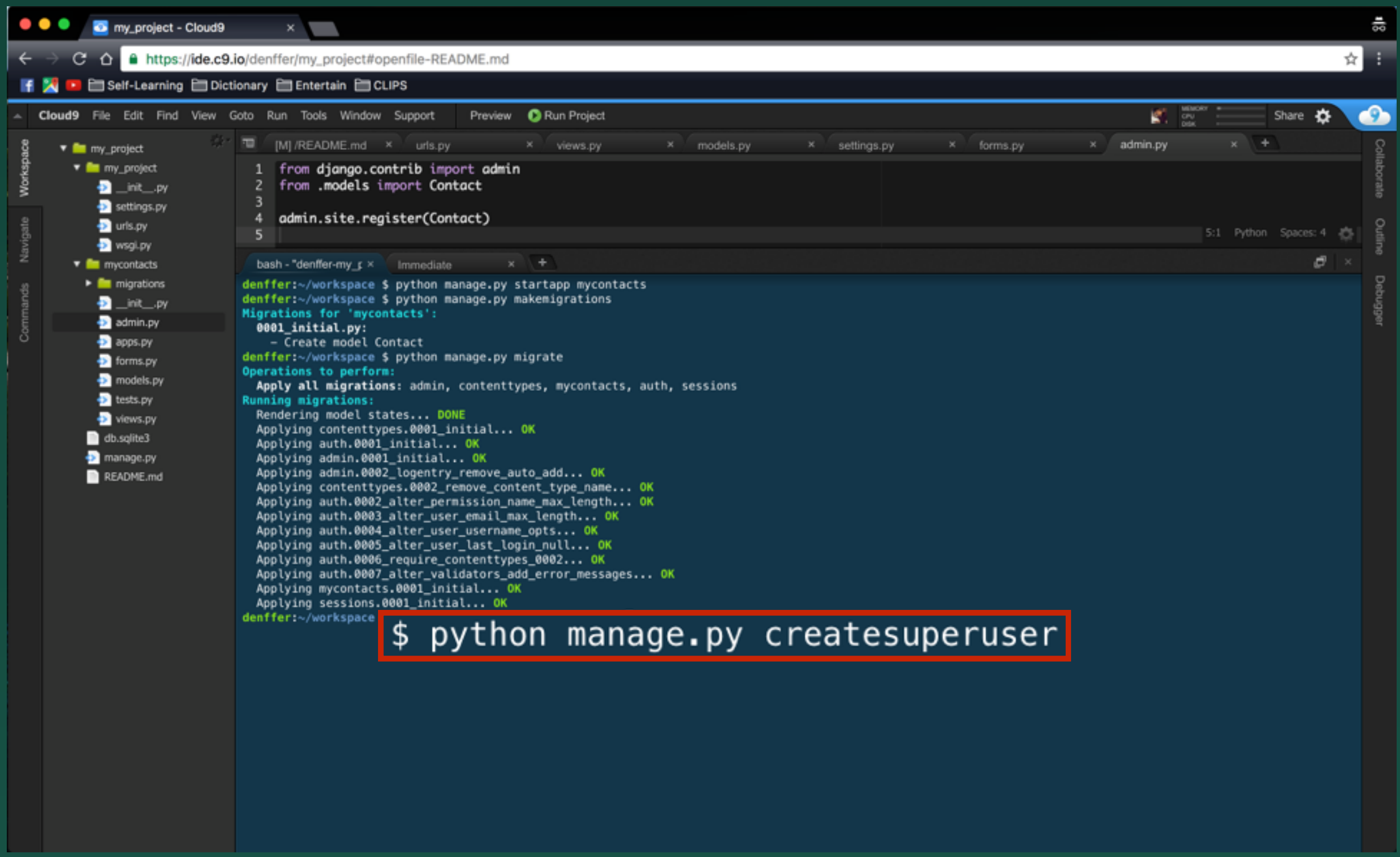
Great! Your first database is just created with your model written in it



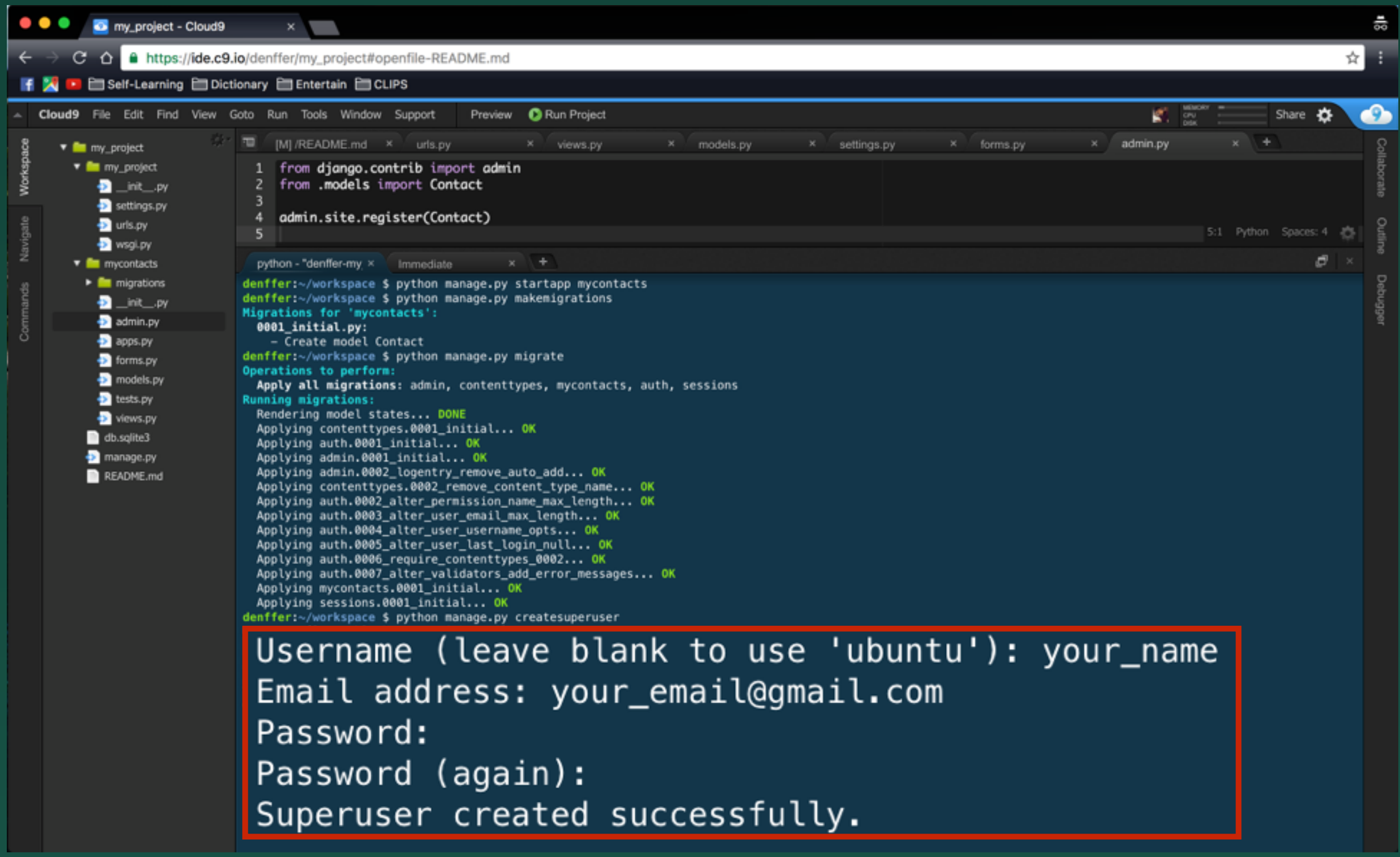
Let’s check on our database
and see how it looks like

However, before that, we have to
create a superuser account

Type in the command to create a superuser



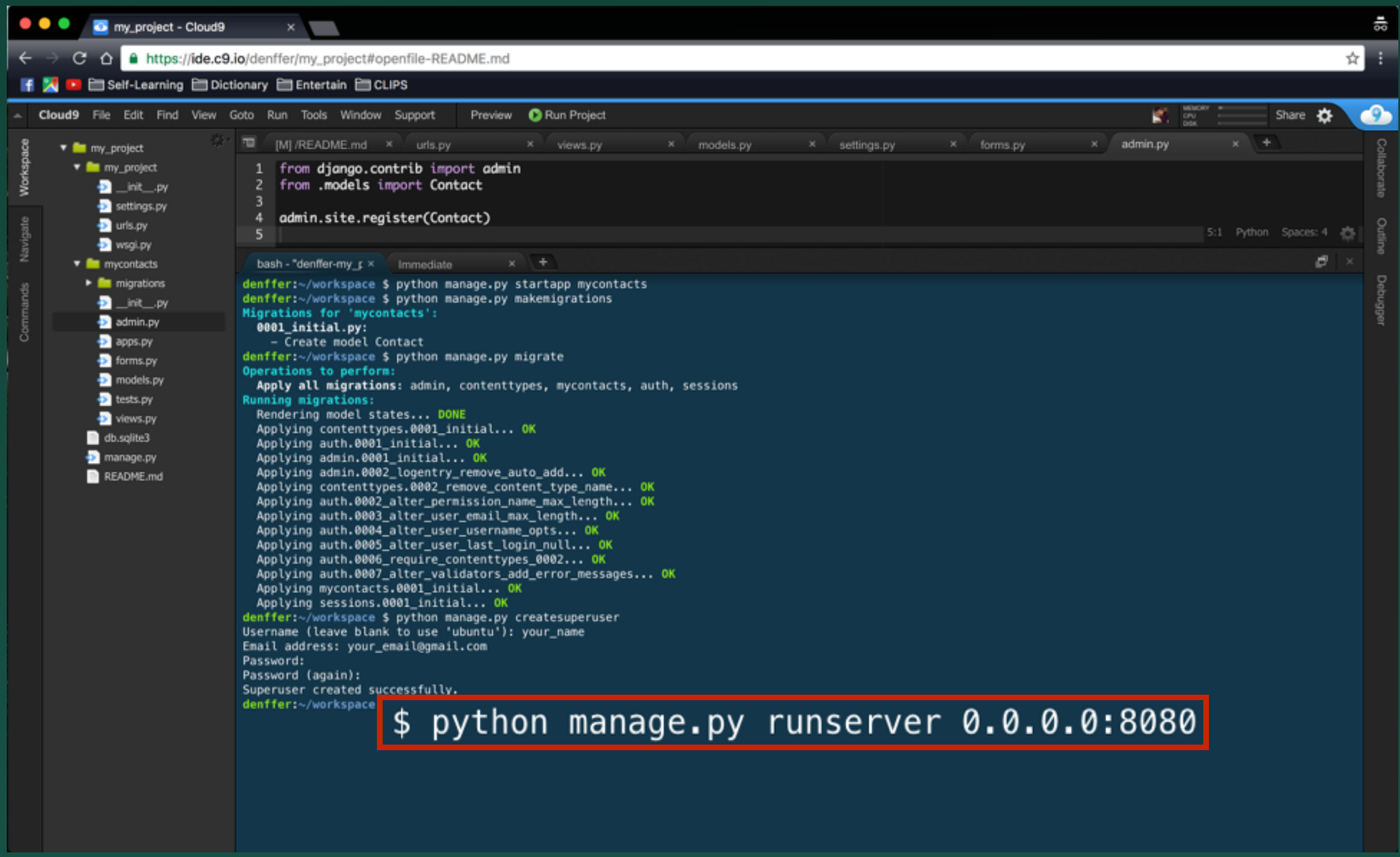
This should be easy



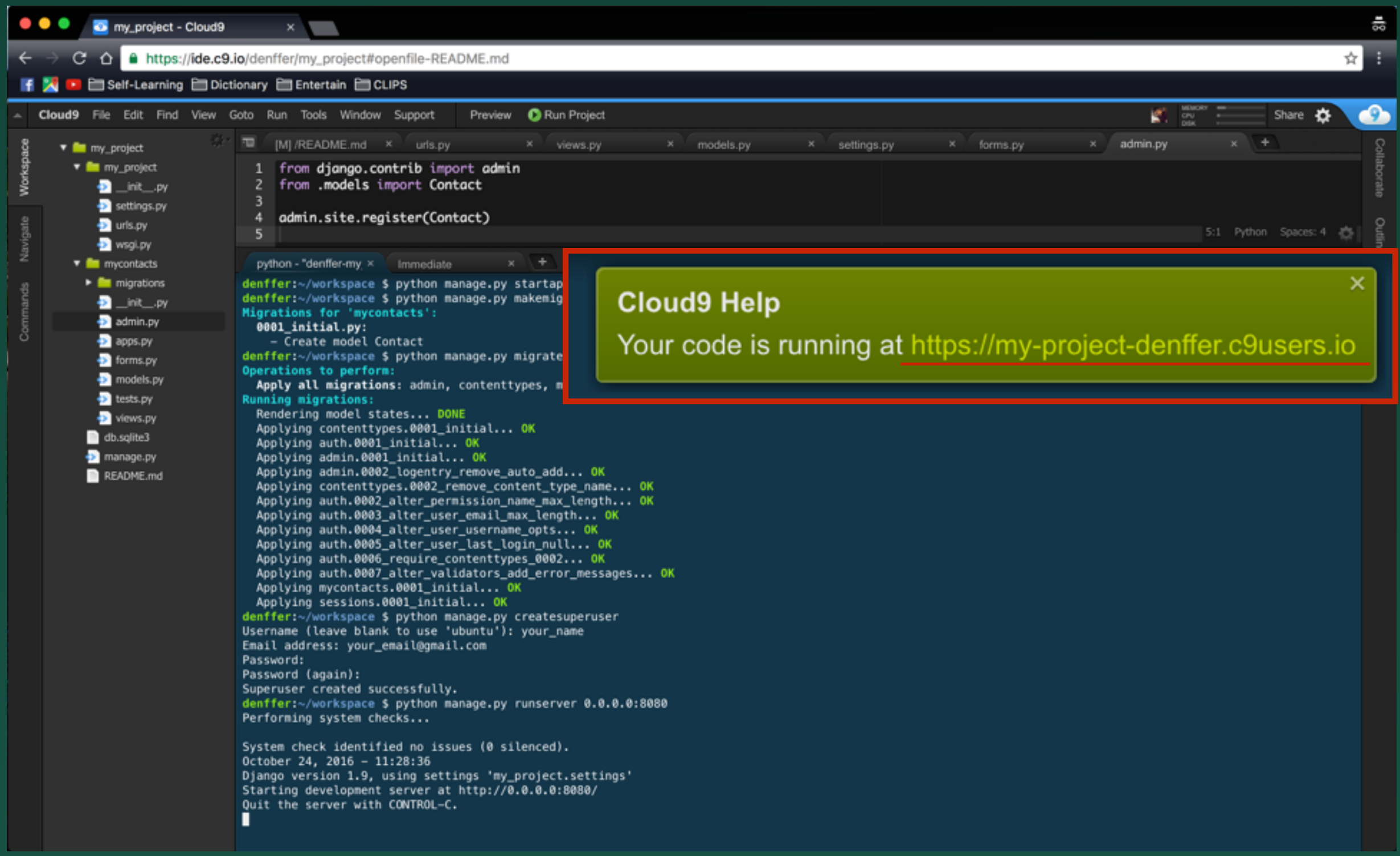
Now comes the interesting part!

Let’s us launch our server!

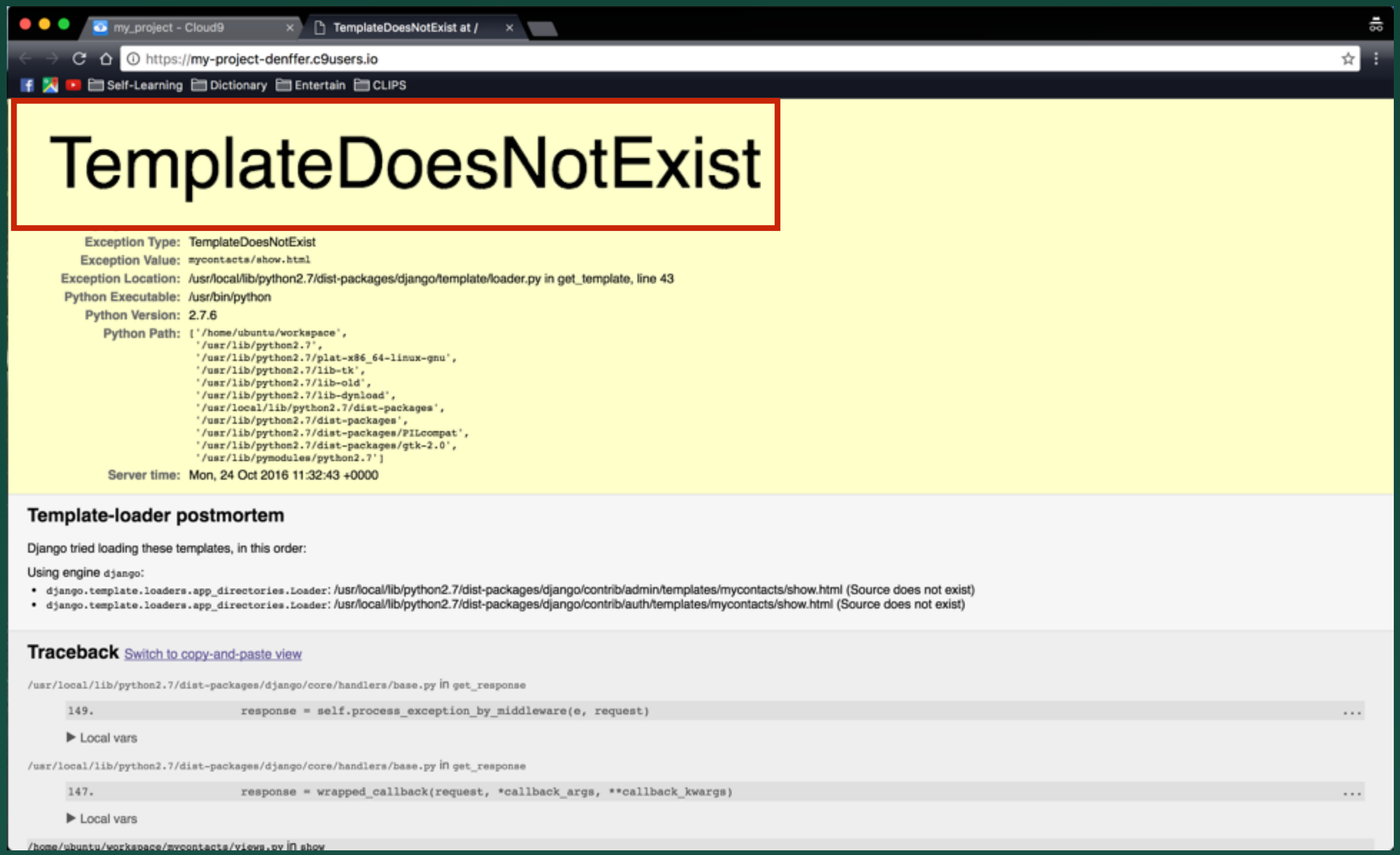
Type in the command to launch the server



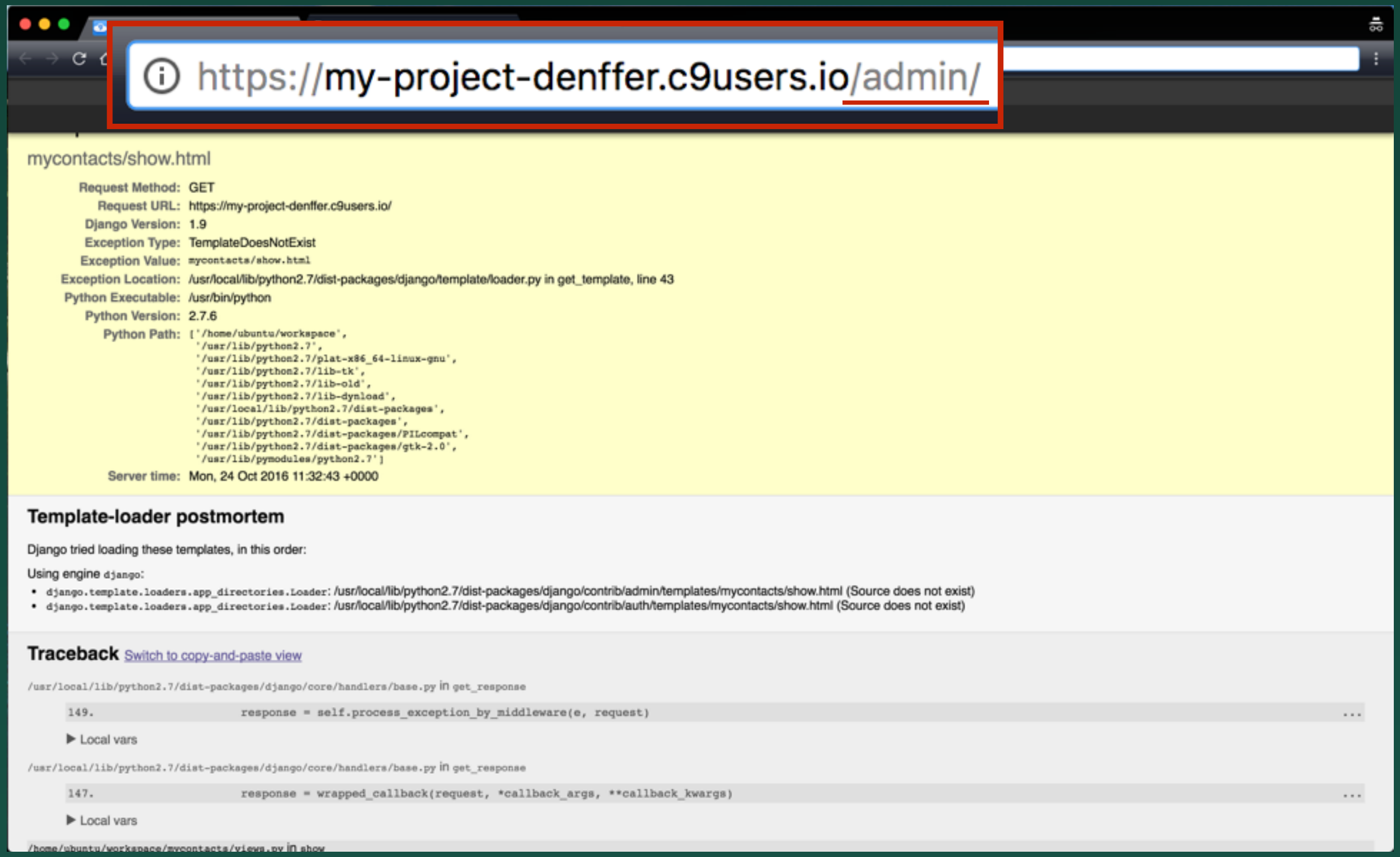
Click on the link to your website!



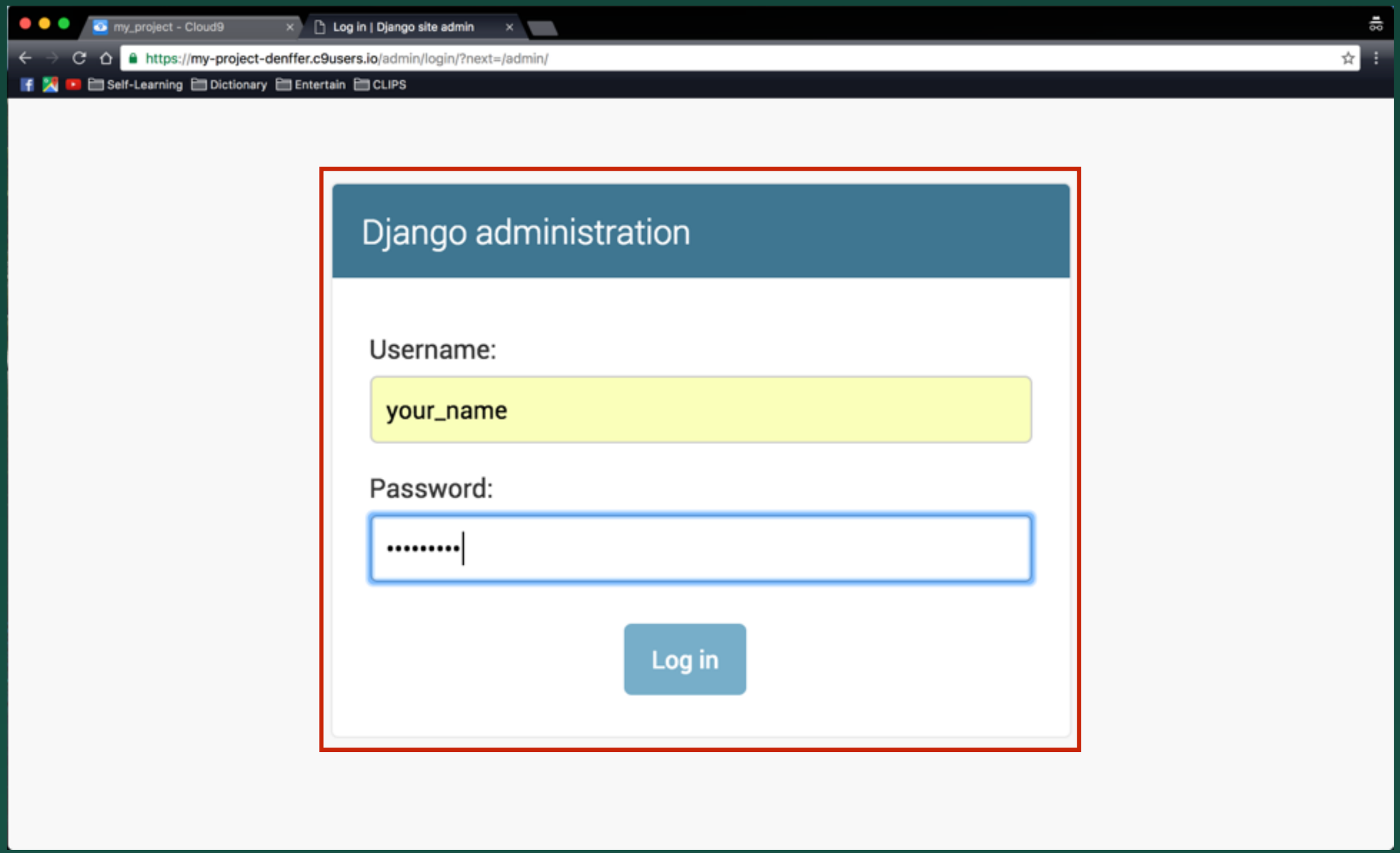
Welcome to your first Django error



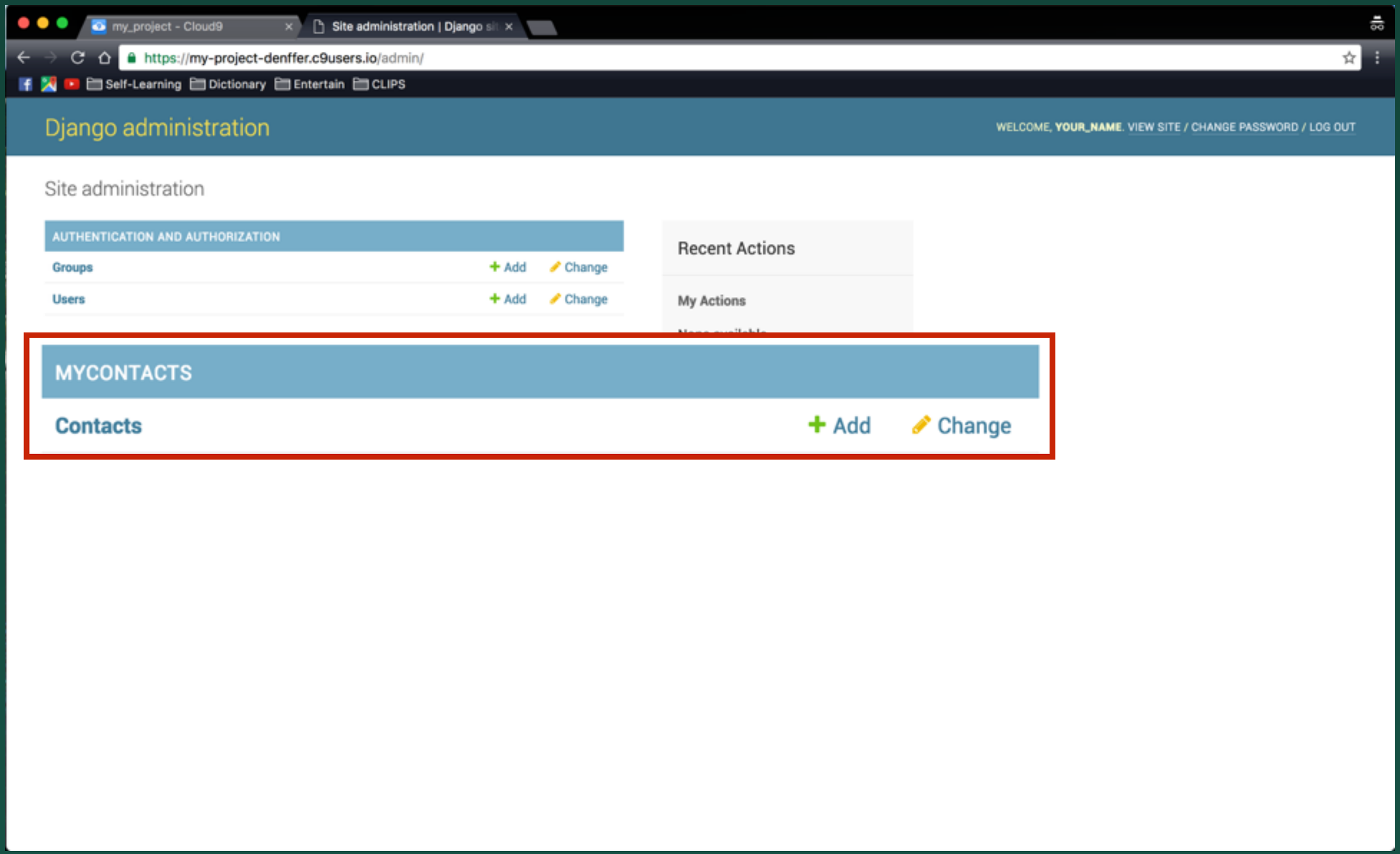
Don’t worry. Let’s edit and add ‘/admin/‘ to your URL



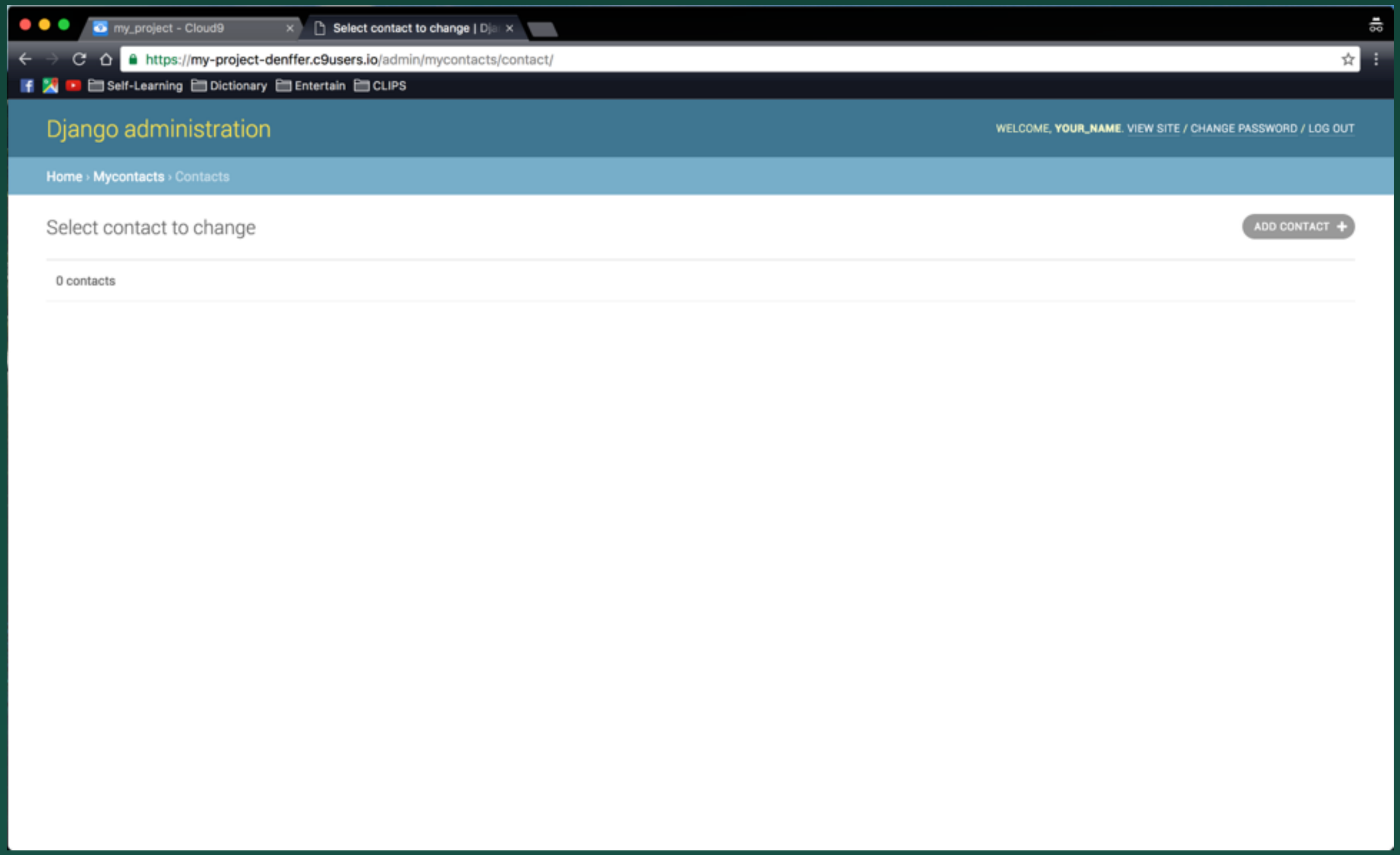
Log in with your superuser account



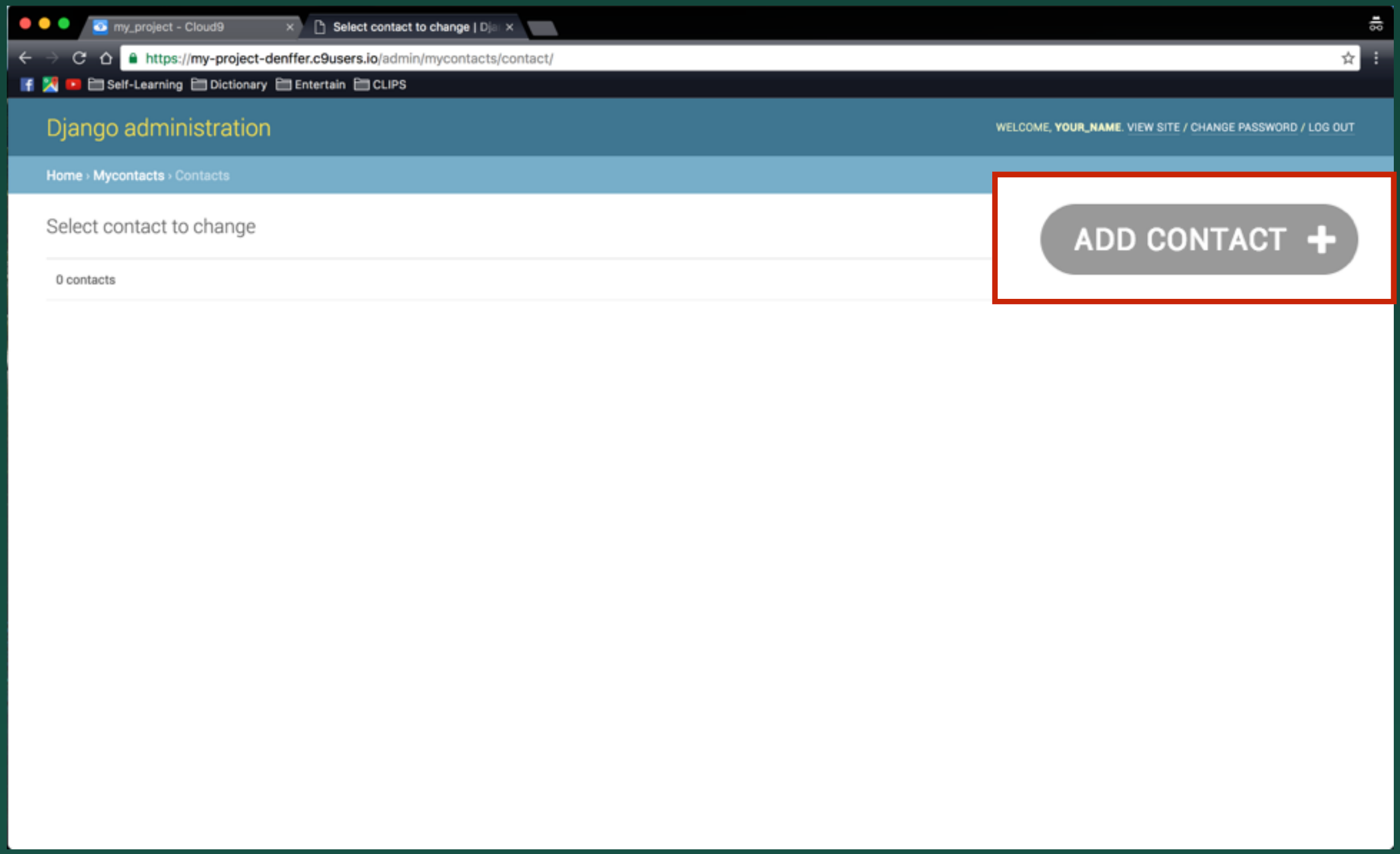
You should be able to see your model



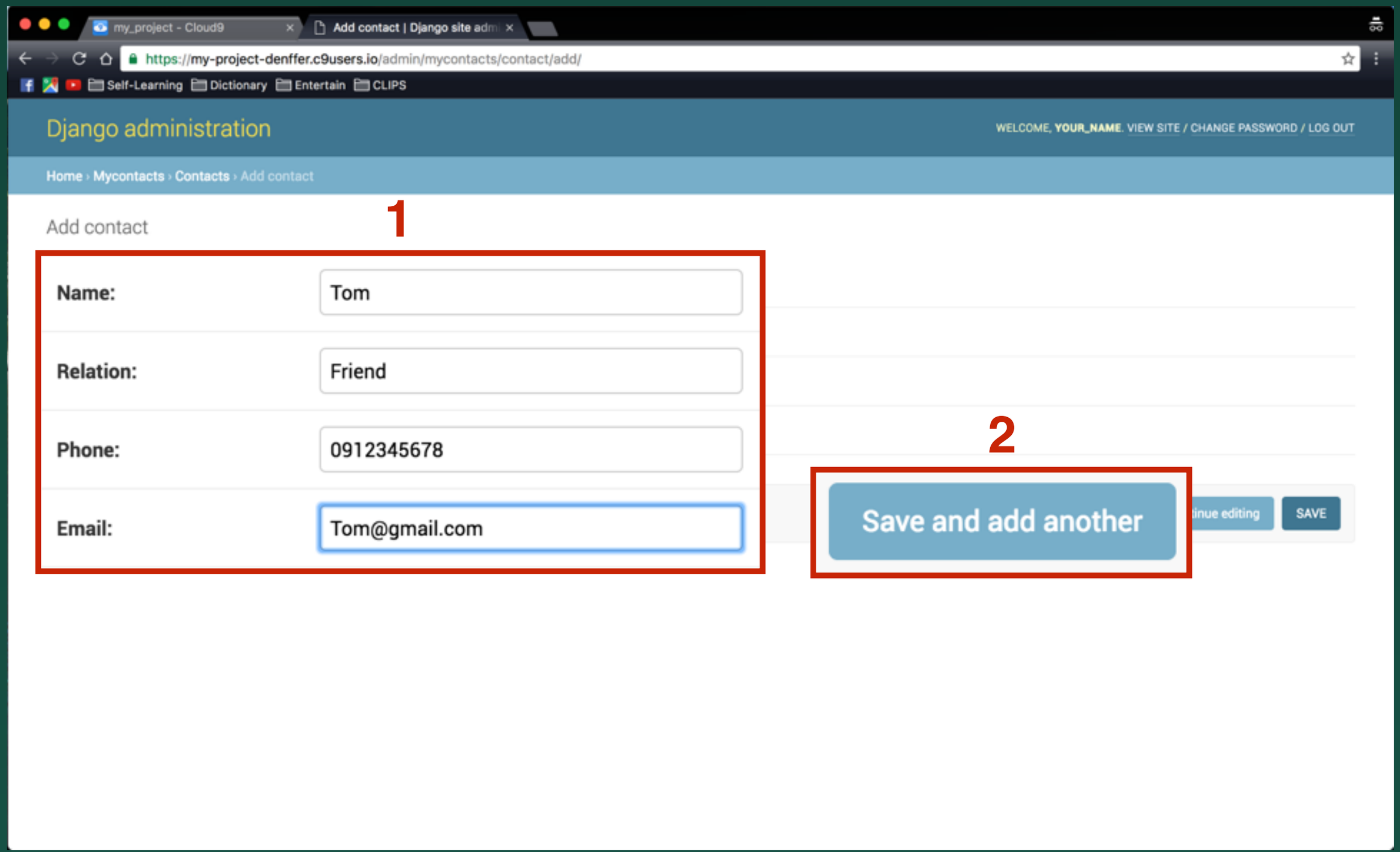
But it is now empty



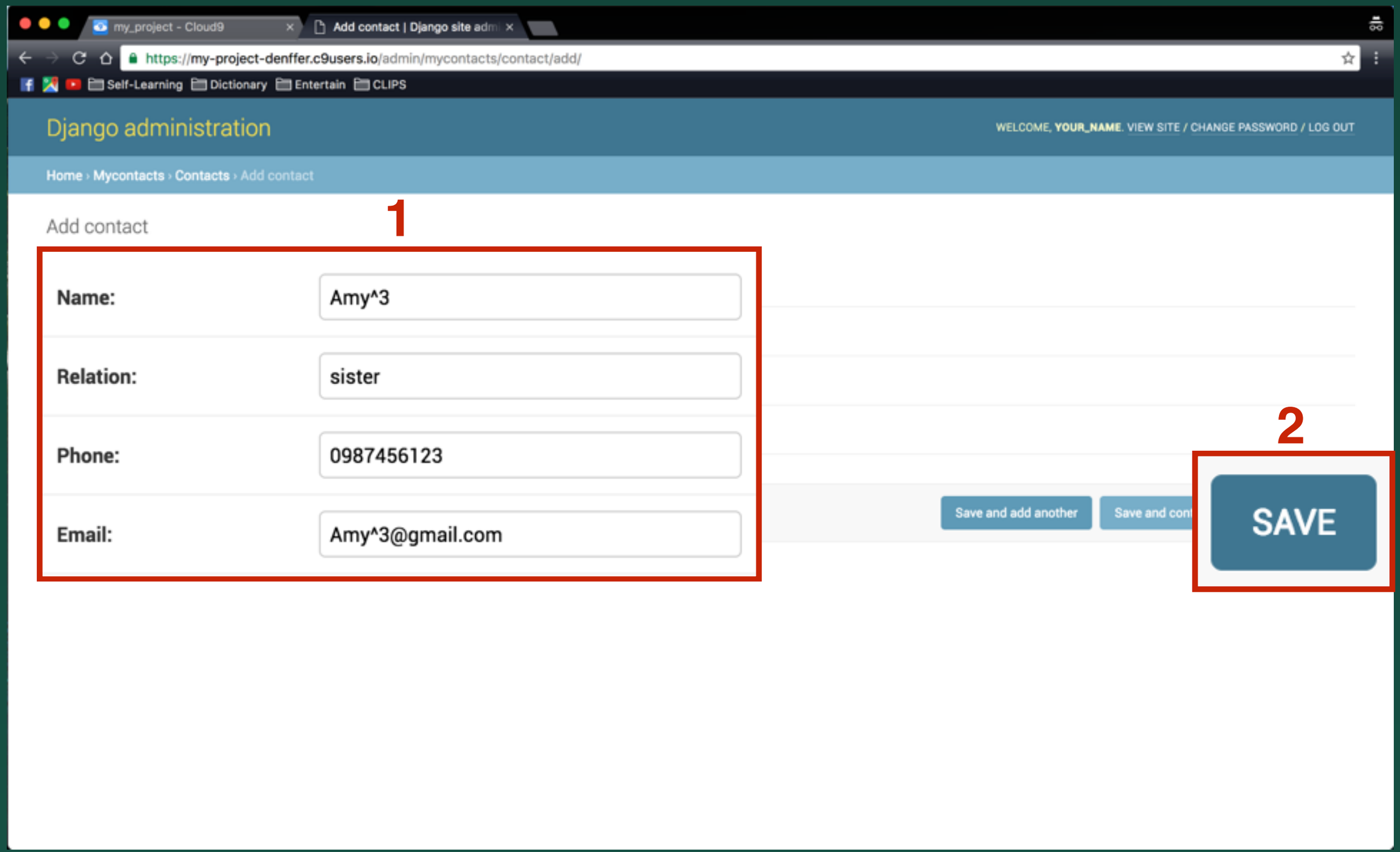
Click on ‘ADD CONTACT’ button



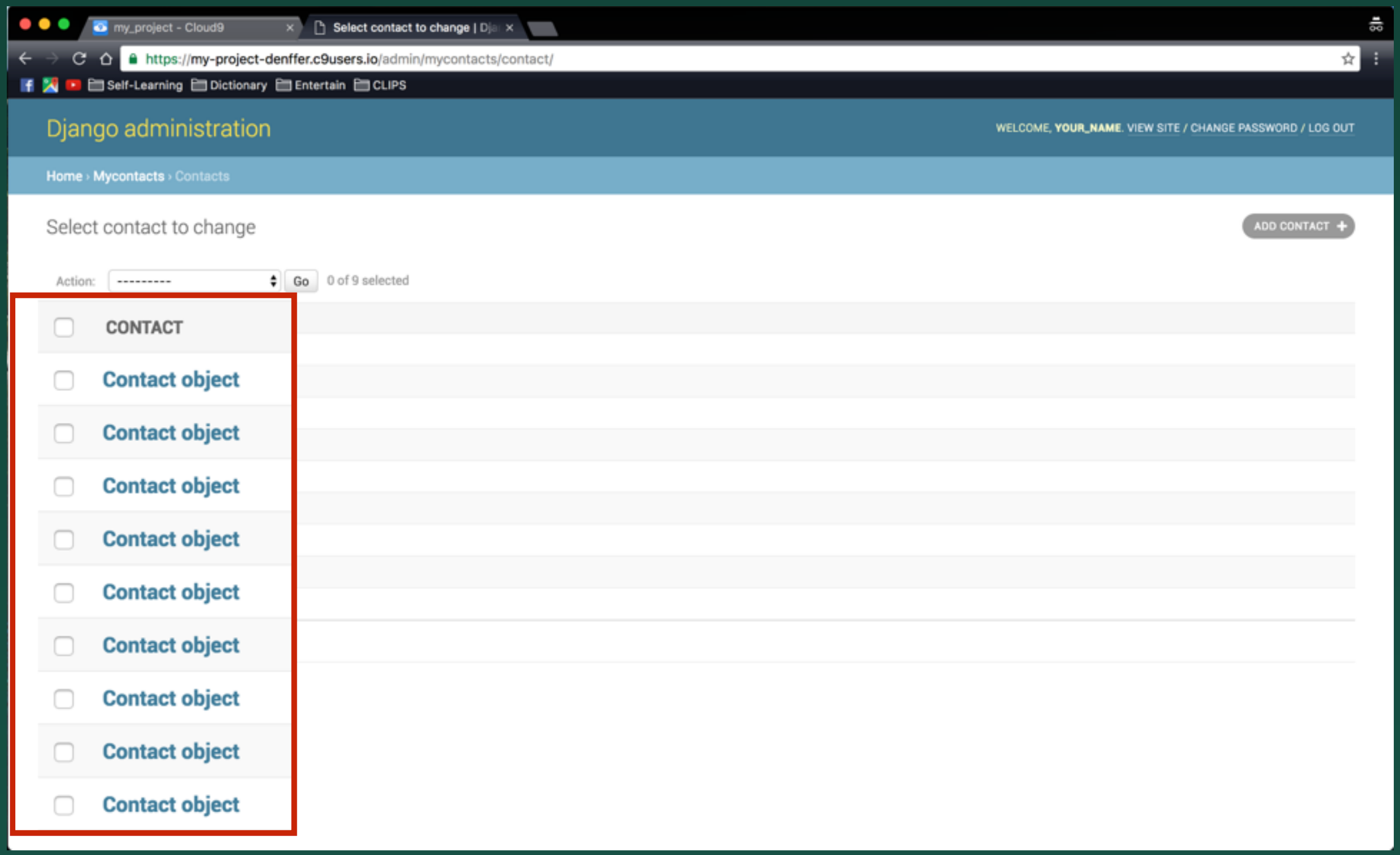
Let's add **9** contact members



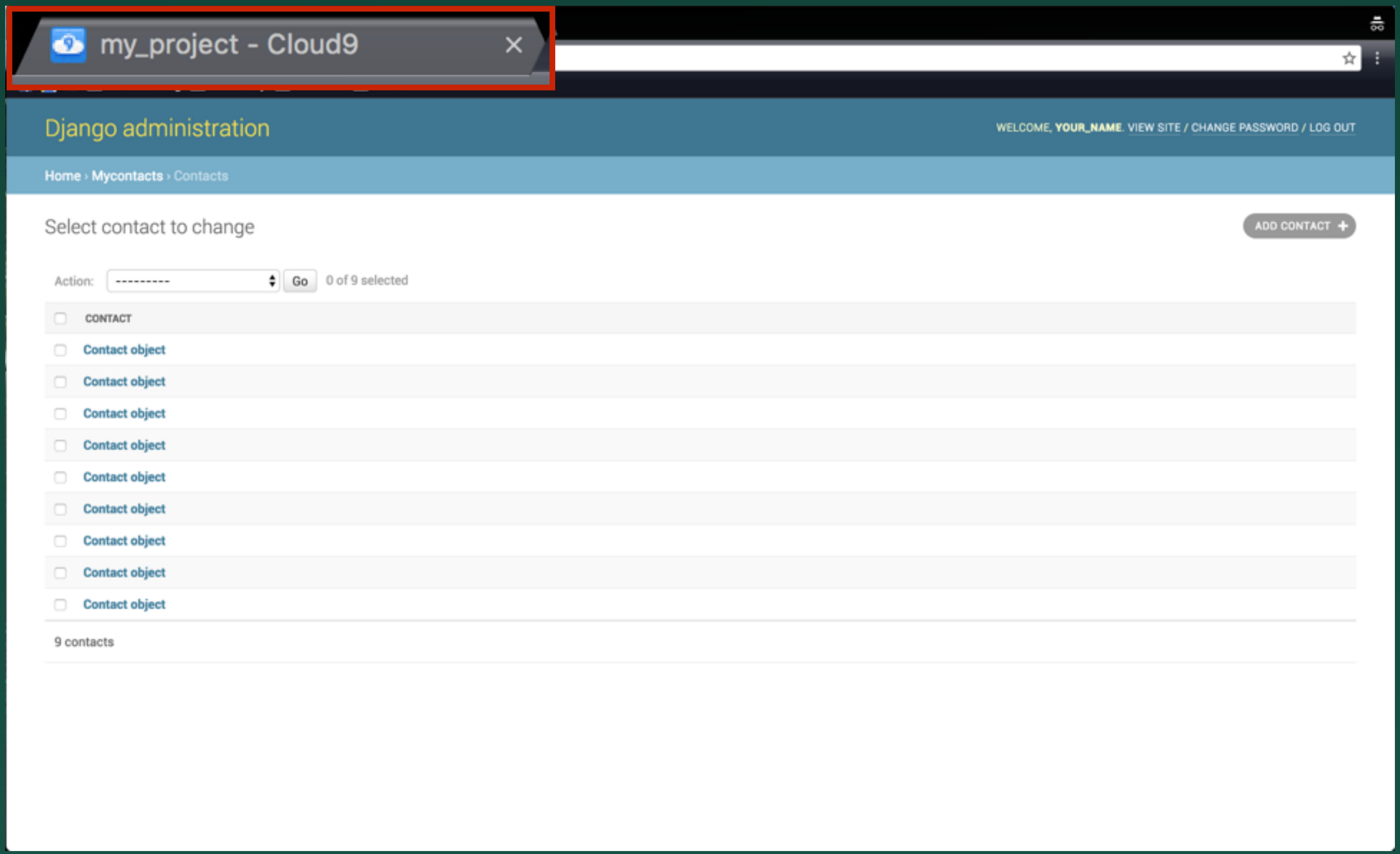
Click ‘SAVE’ button on the last contact member



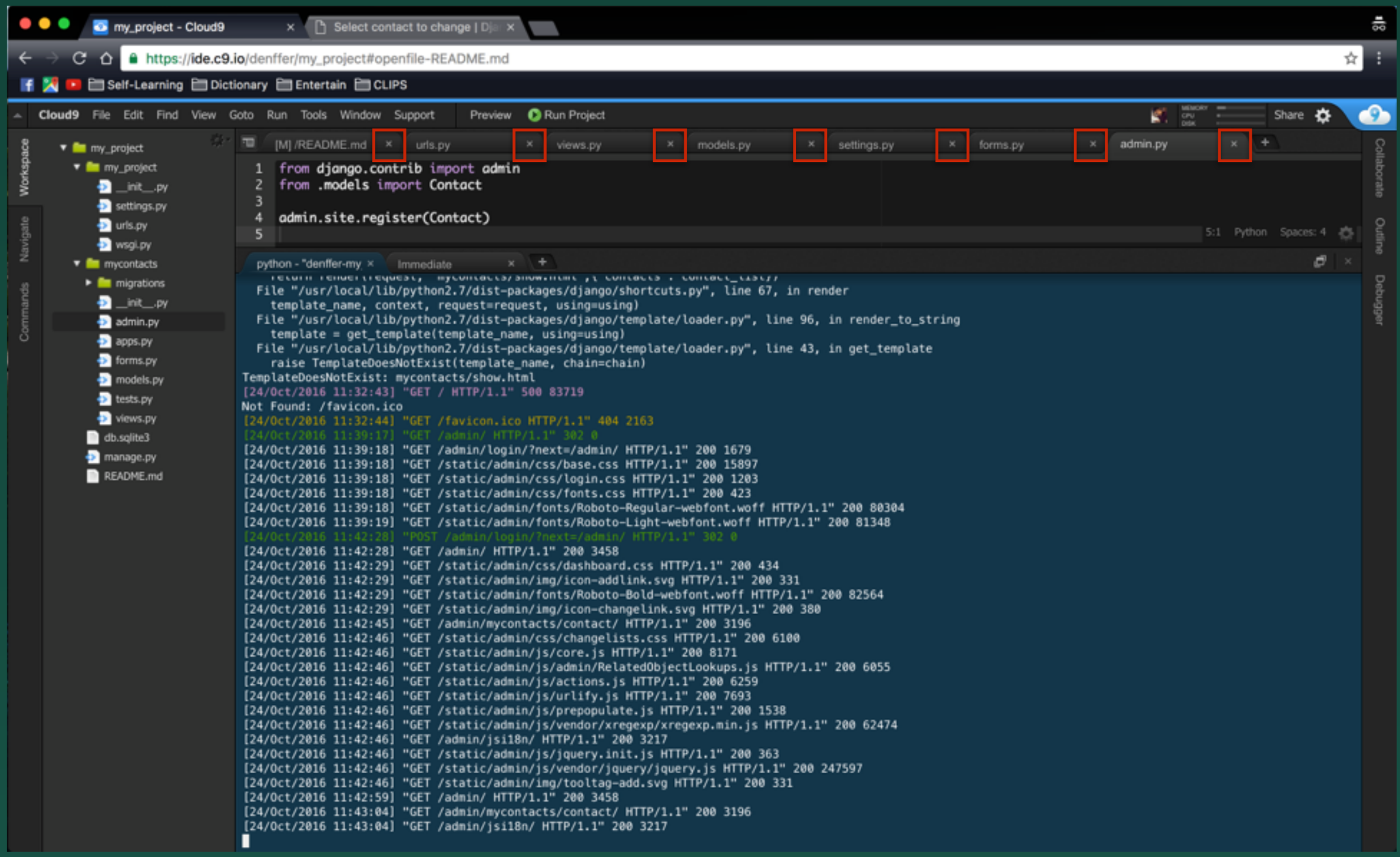
Now you should see 9 contact members



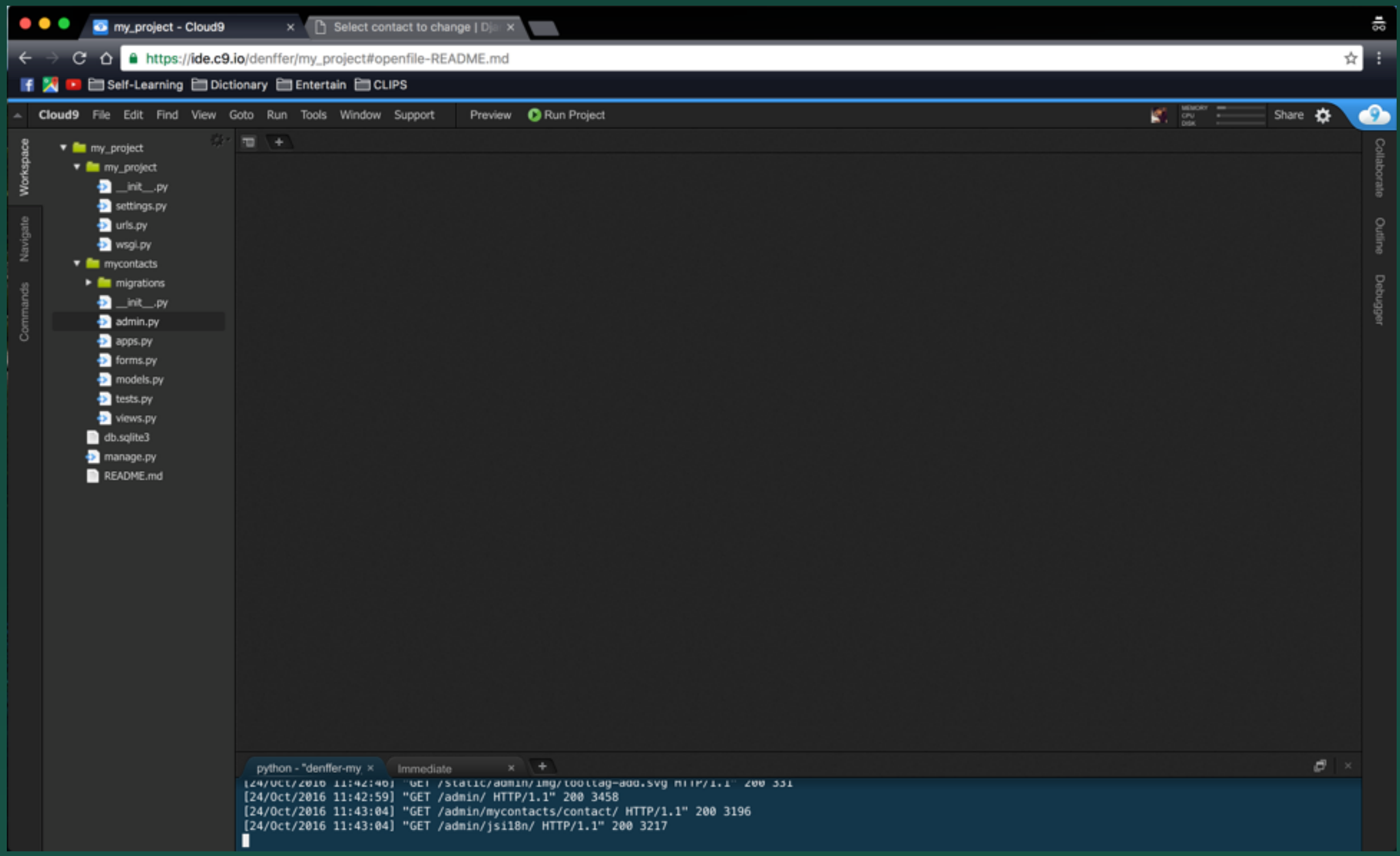
Let’s go back to ‘my_project’



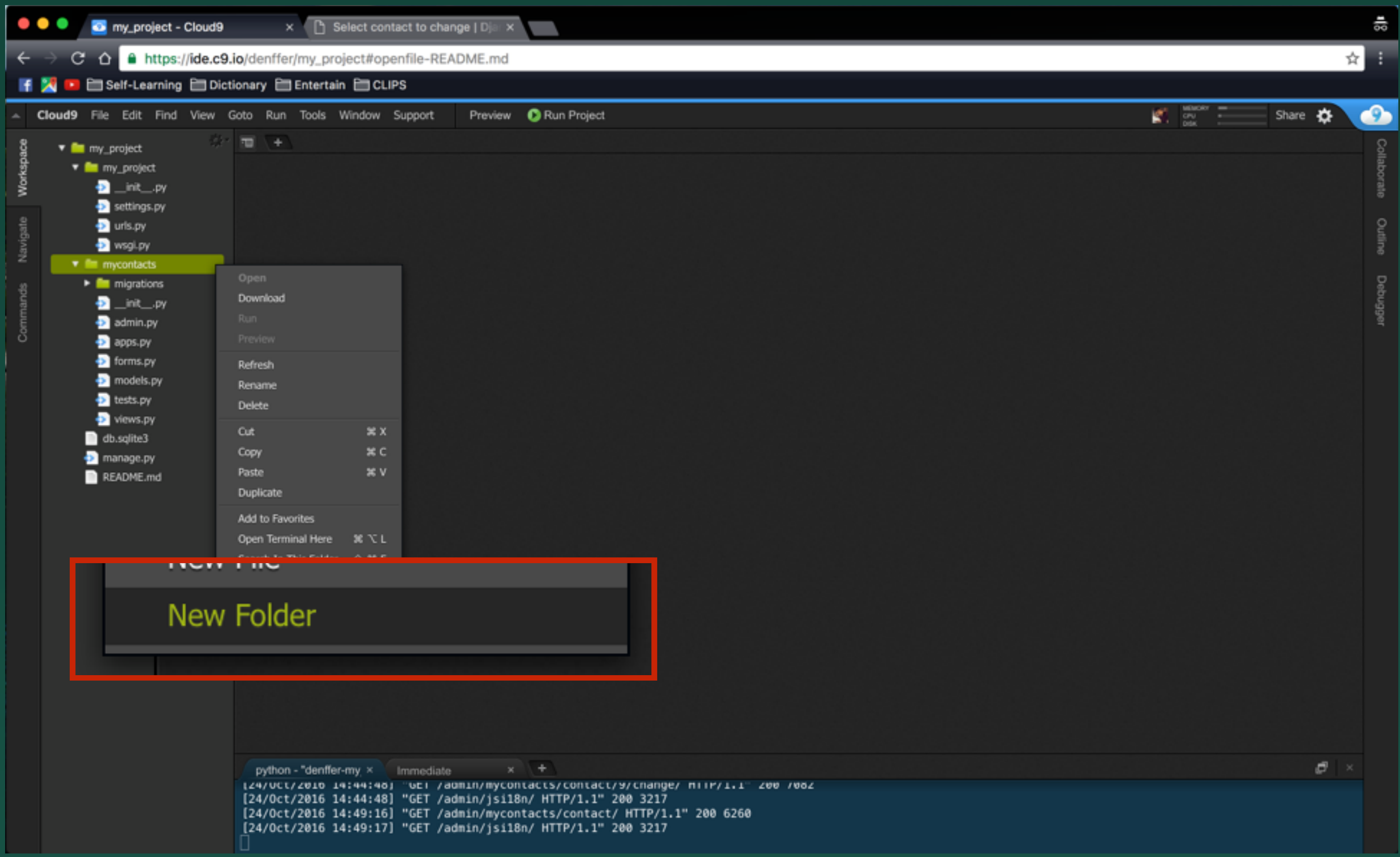
Too messy. Let's kill all the tabs



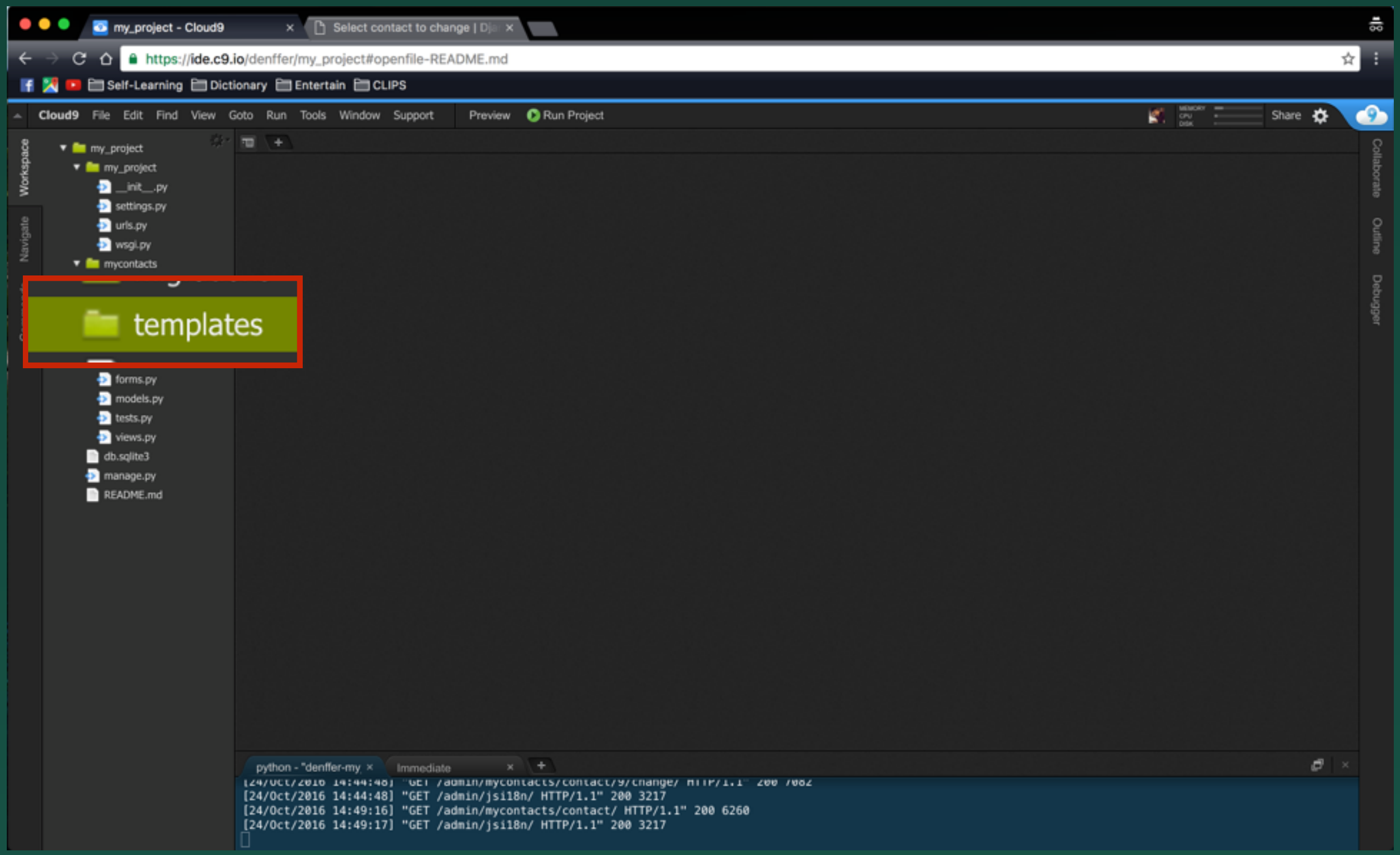
Great! Now let’s deal with the ‘template’ problem



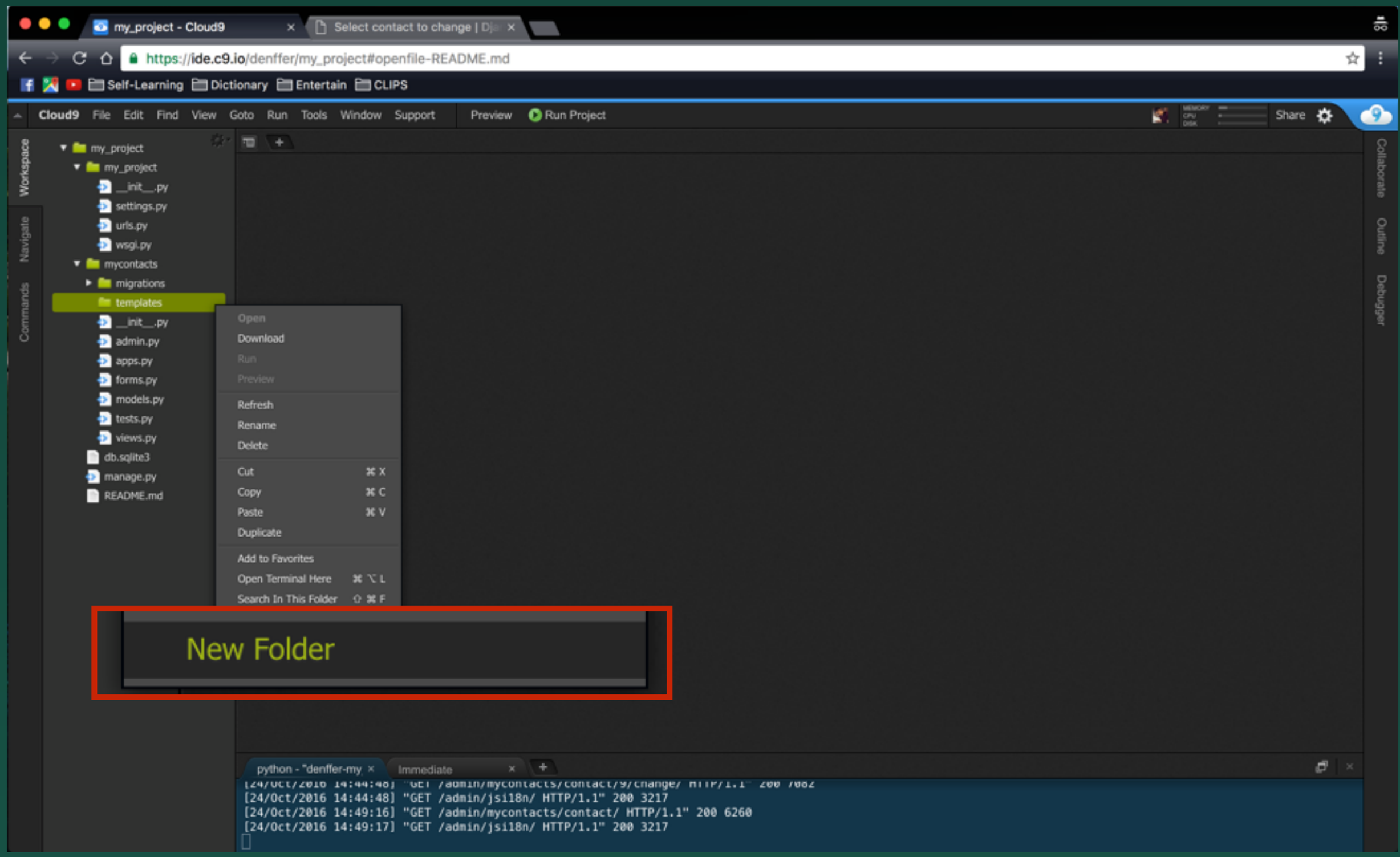
Create a folder named ‘templates’ under your application ‘mycontacts’



Remember? templates are used to contain all of your **Html** files

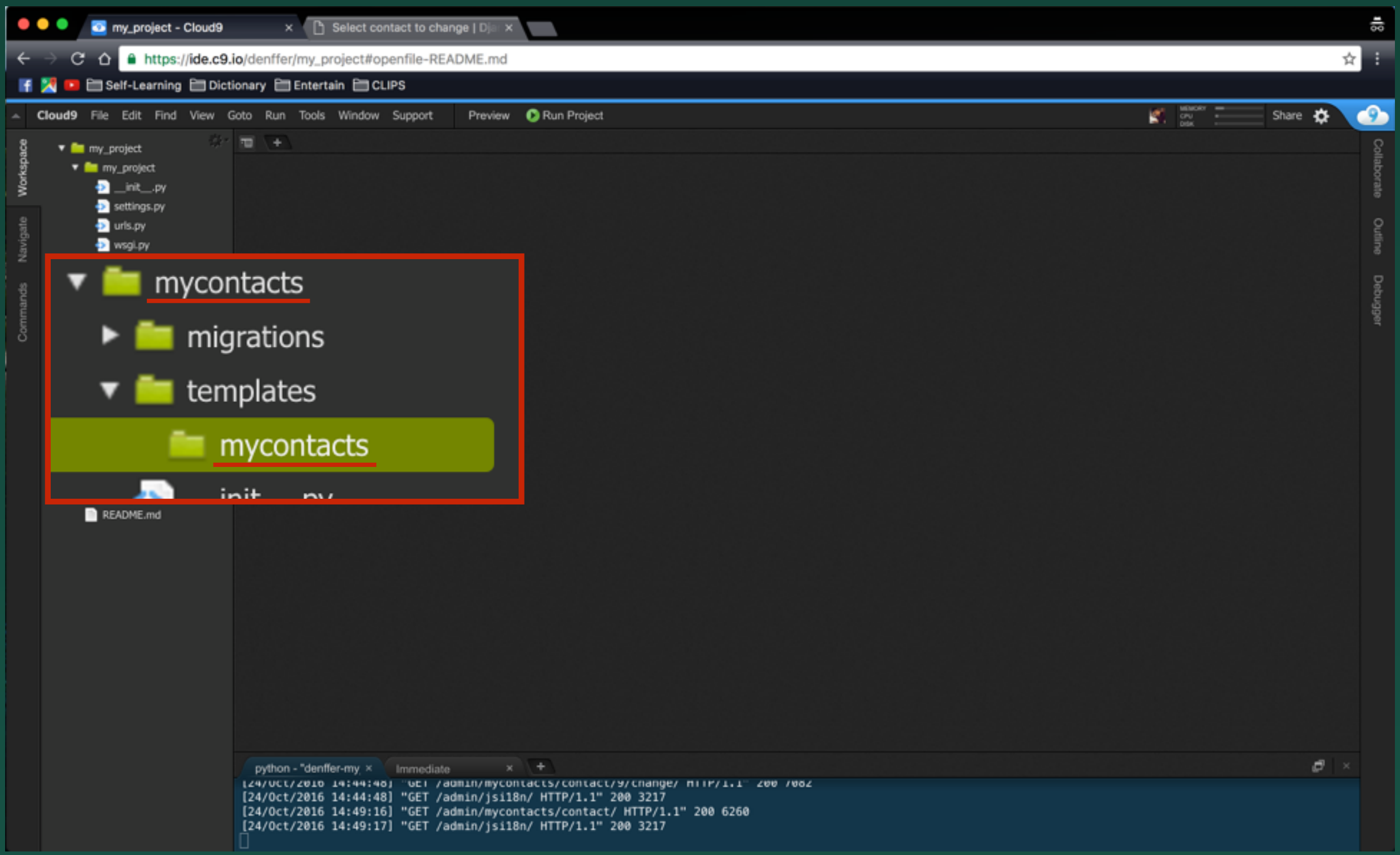


Now create another folder named ‘mycontacts’ in the folder ‘templates’

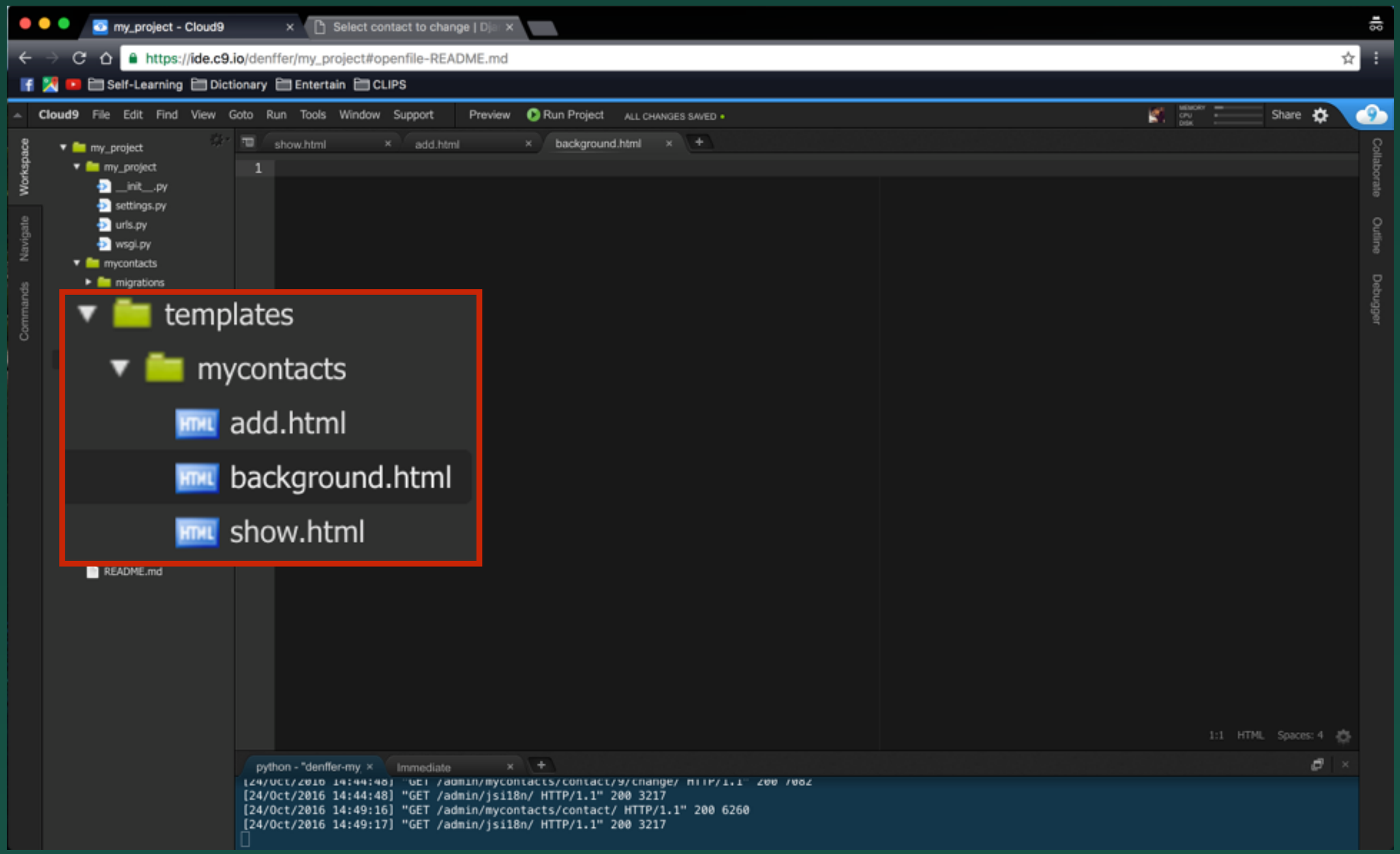


Remember the name of this folder should be the

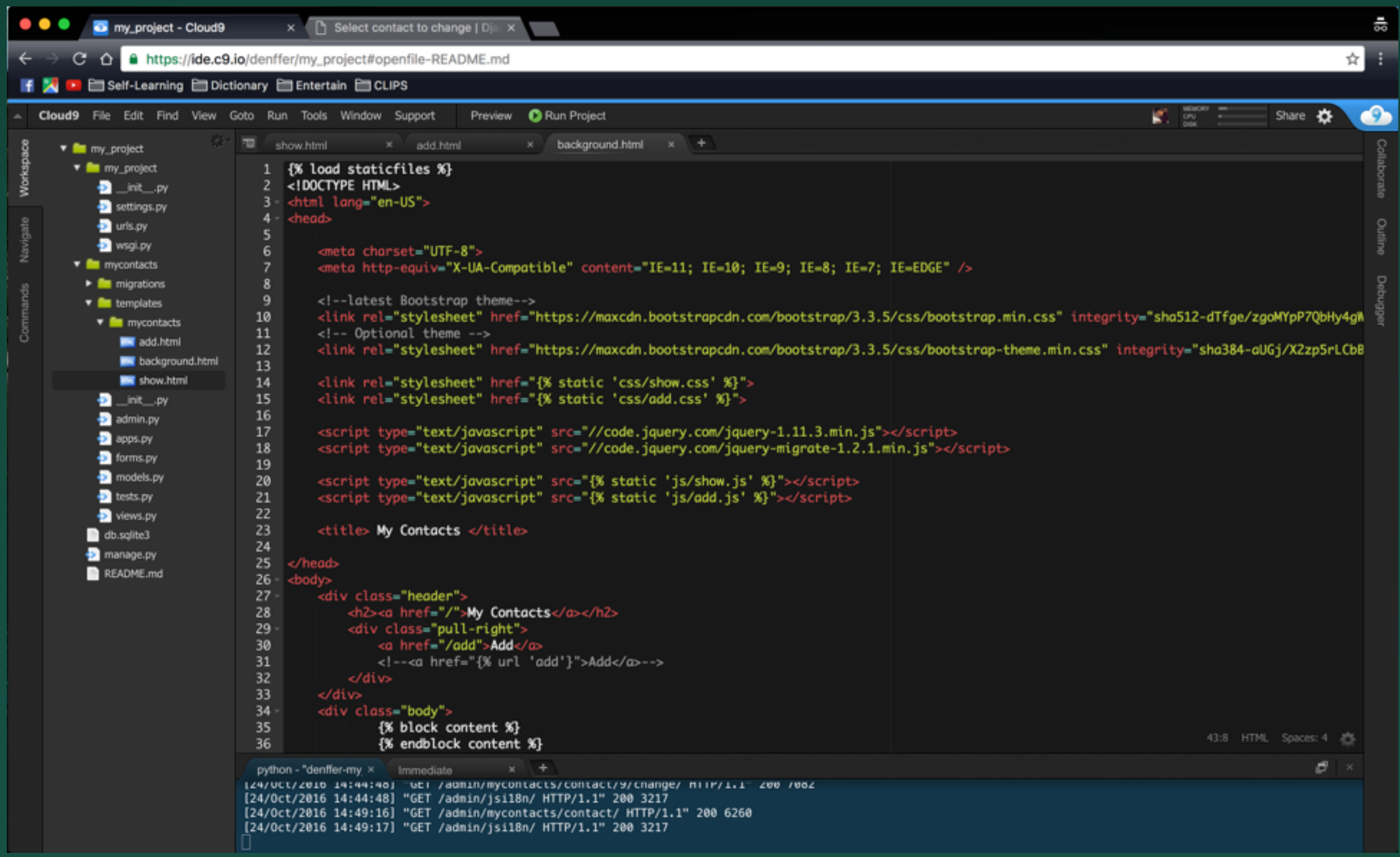
same as the name of your application



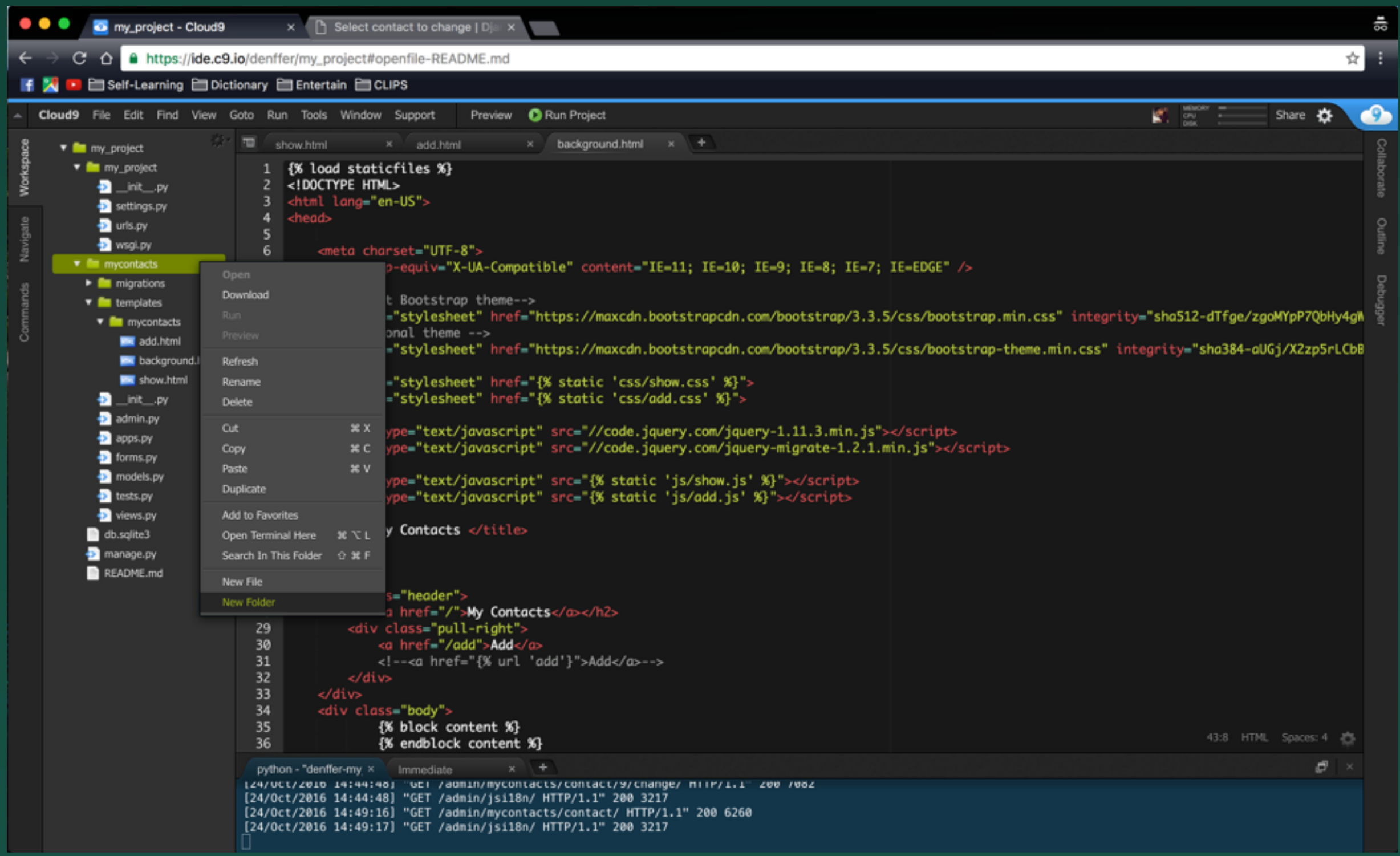
Create ‘show.html’, ‘add.html’, and ‘background.html’ in the directory ‘mycontacts/templates/mycontacts’



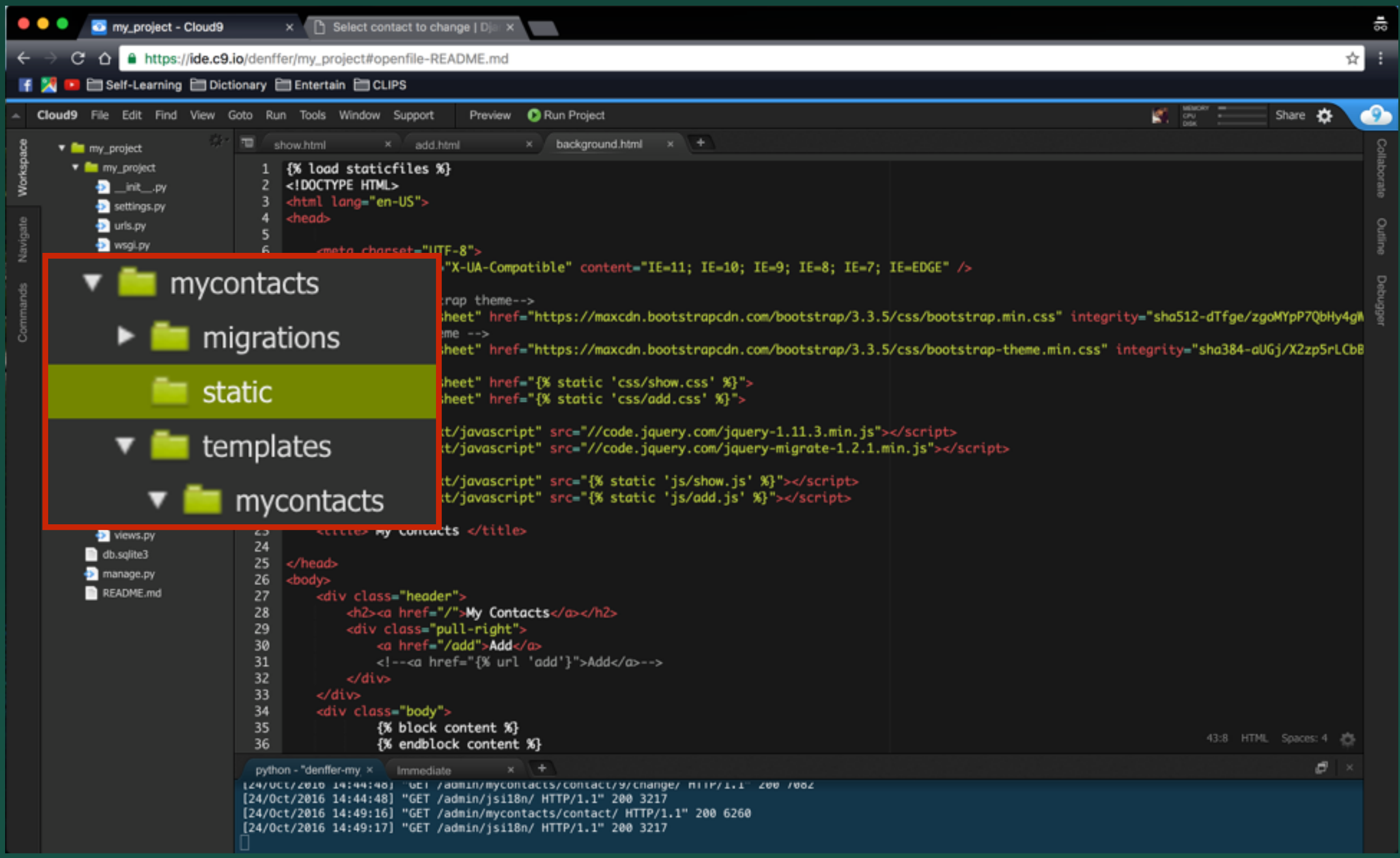
Respectively copy and paste the codes in the **html** files from the repository you cloned from Github



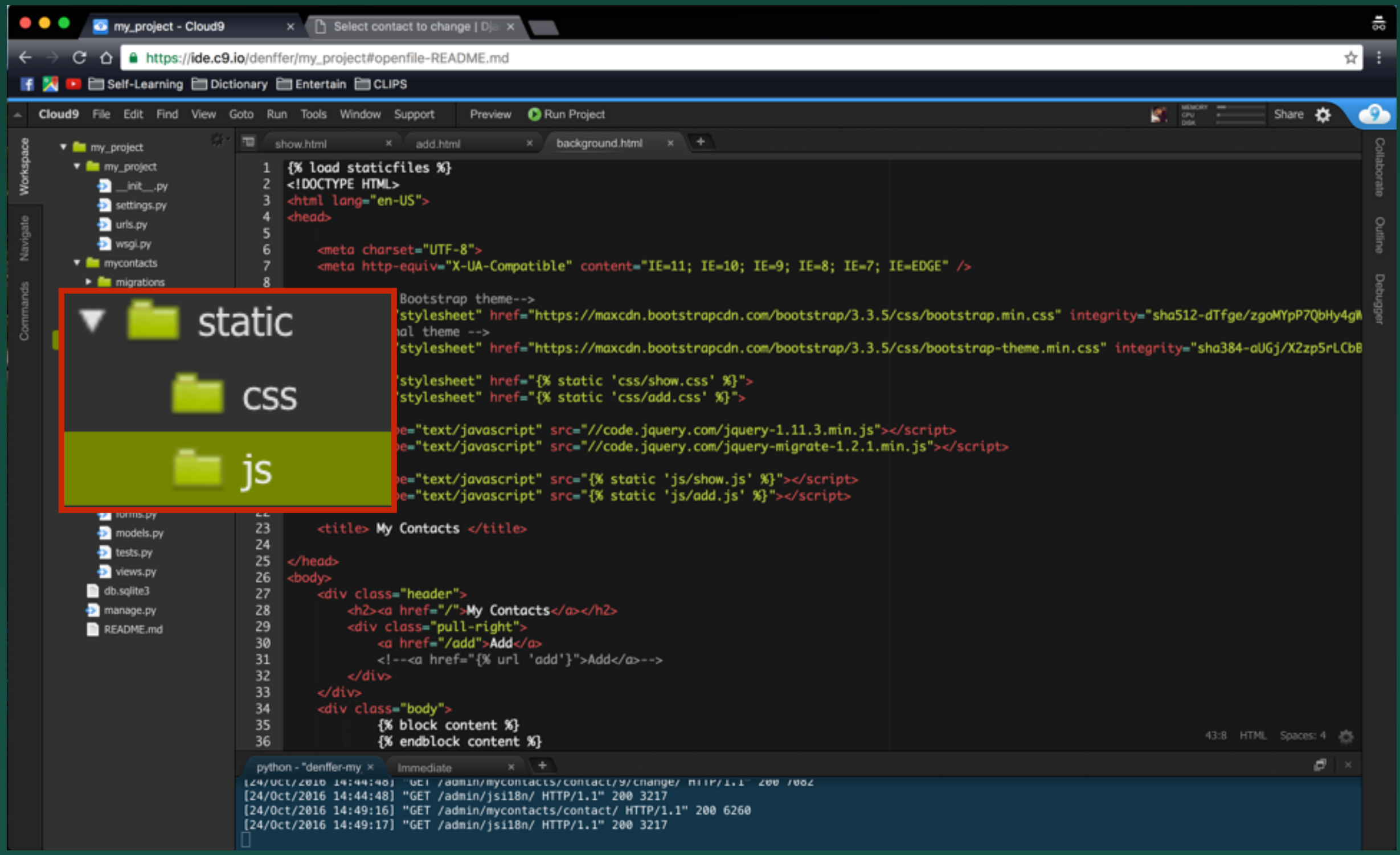
Next, create a folder named ‘static’ in your application



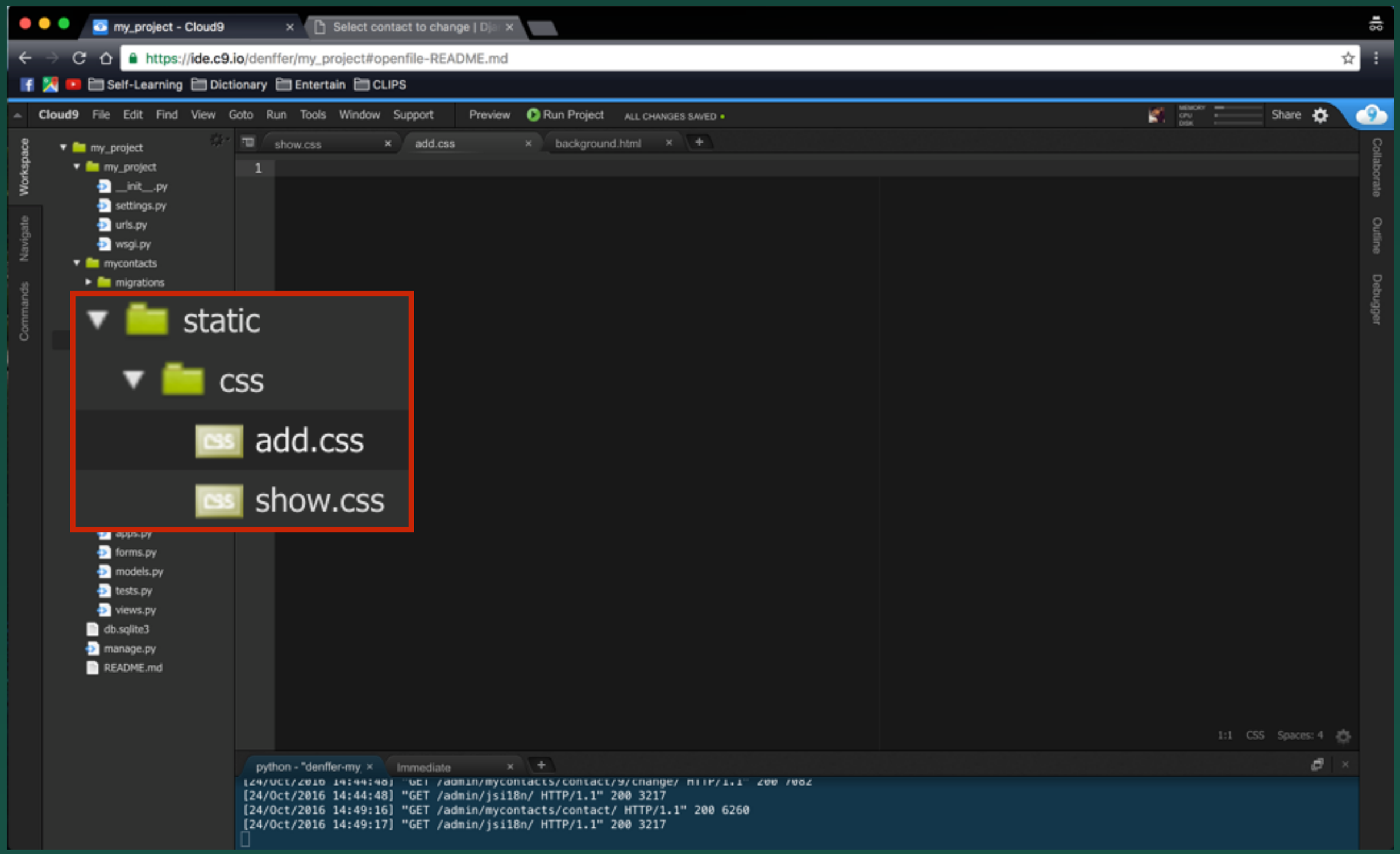
Remember? ‘static’ is used to contain all of your **CSS** files and **Javascript** files



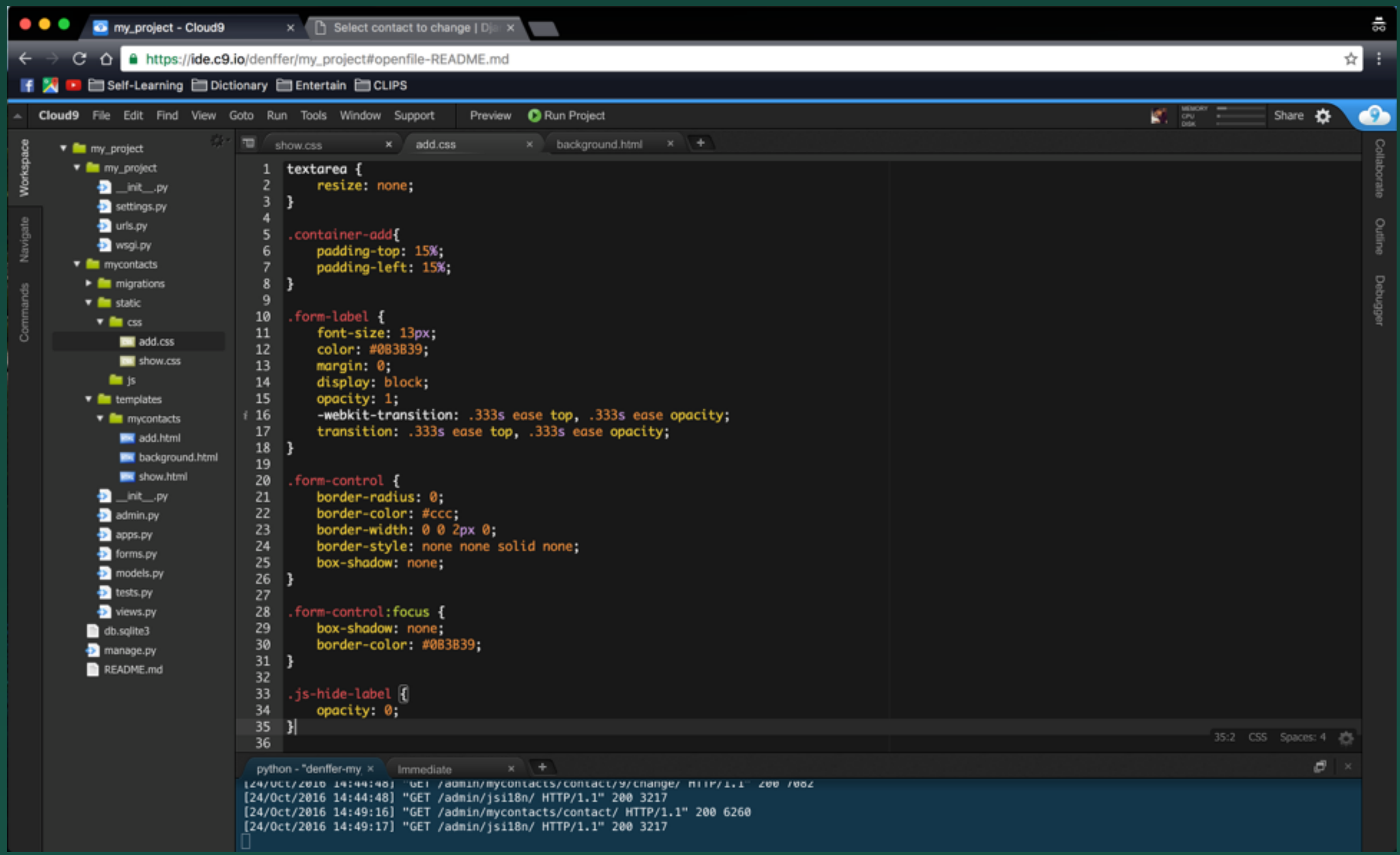
Next, create two folders named ‘css’ and ‘js’ in the folder ‘static’



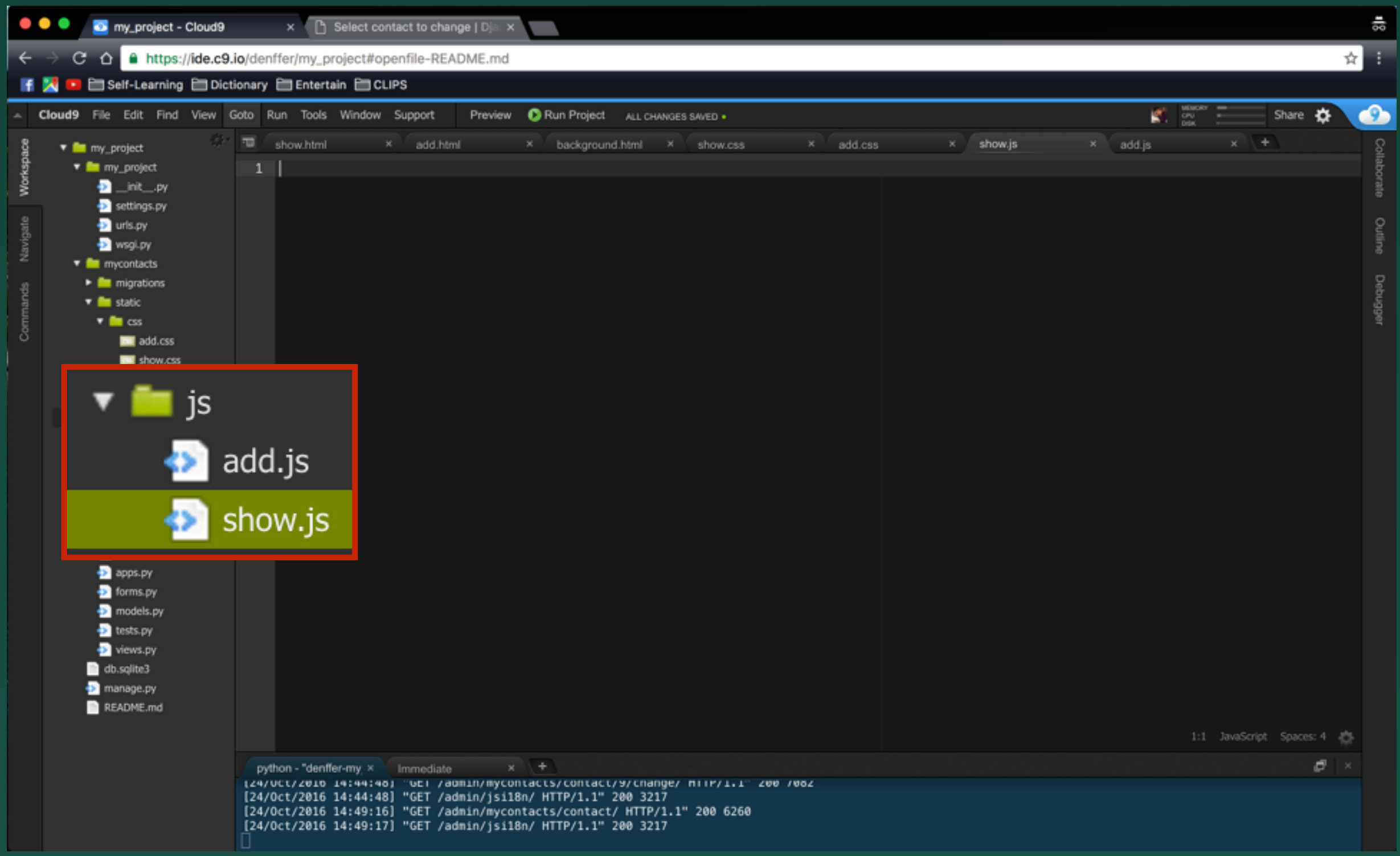
Create ‘show.css’ & ‘add.css’ in the folder ‘css’



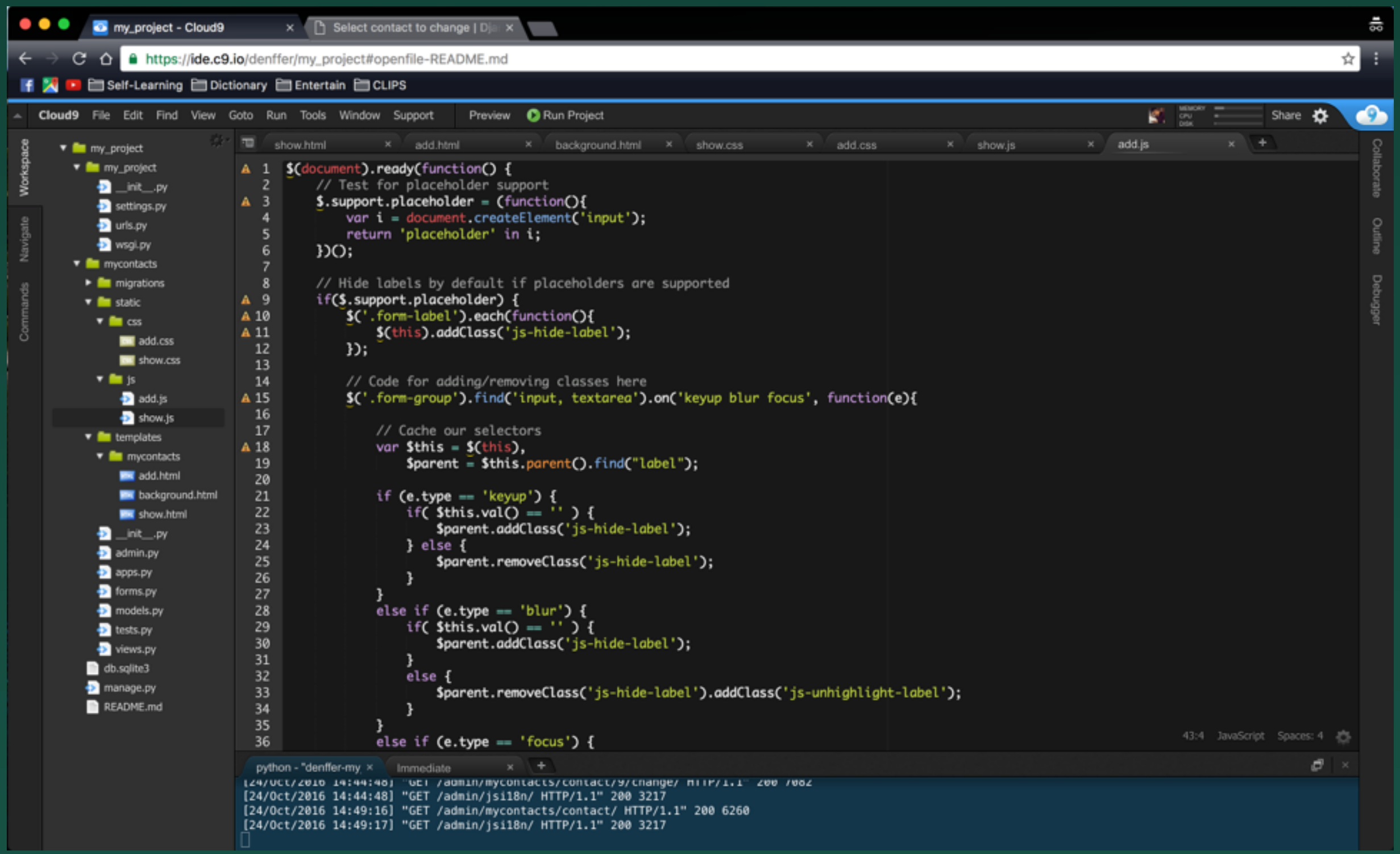
Respectively copy and paste the codes in the **CSS** files from the repository you cloned from Github



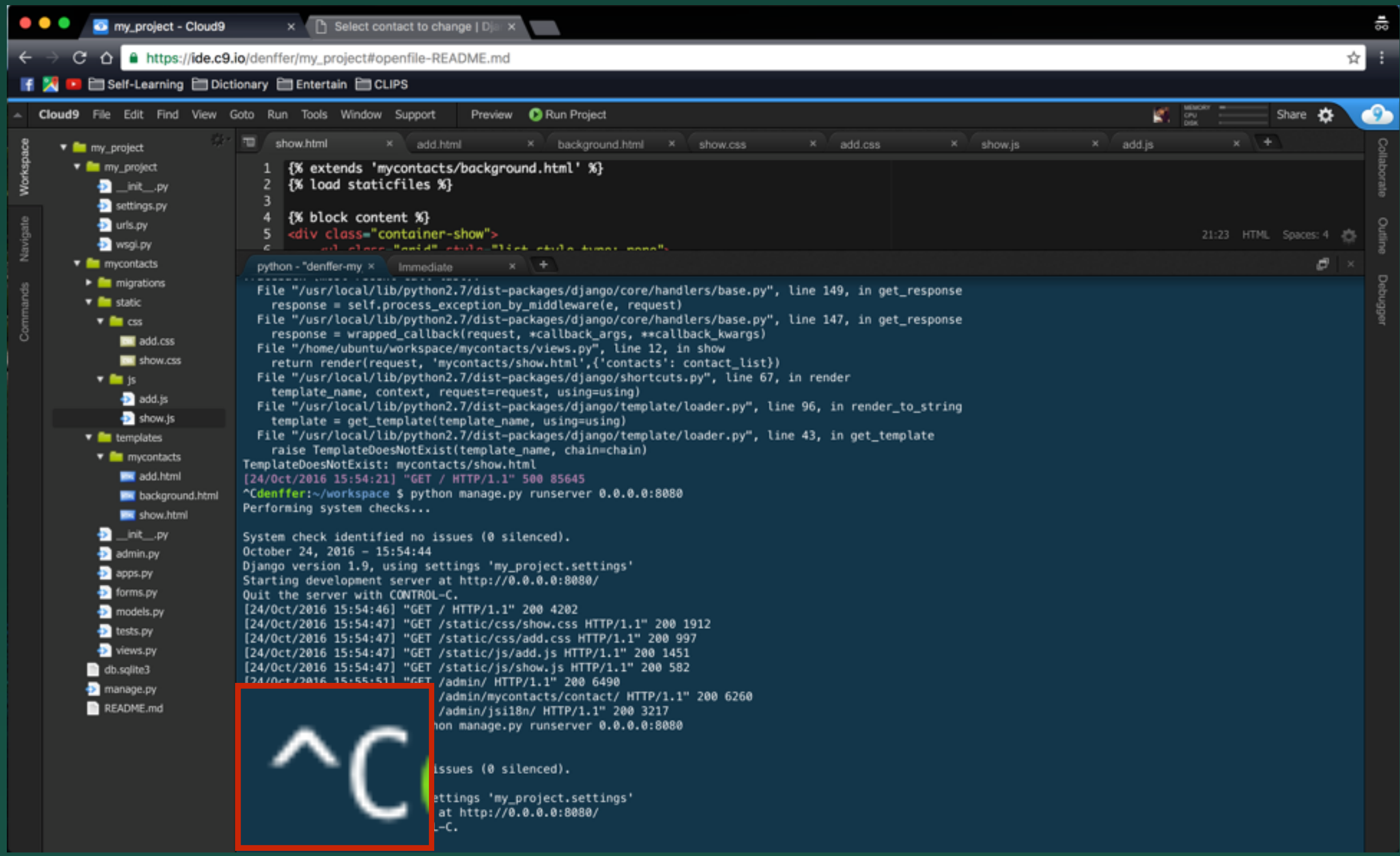
Create ‘show.js’ & ‘add.js’ in the folder ‘js’



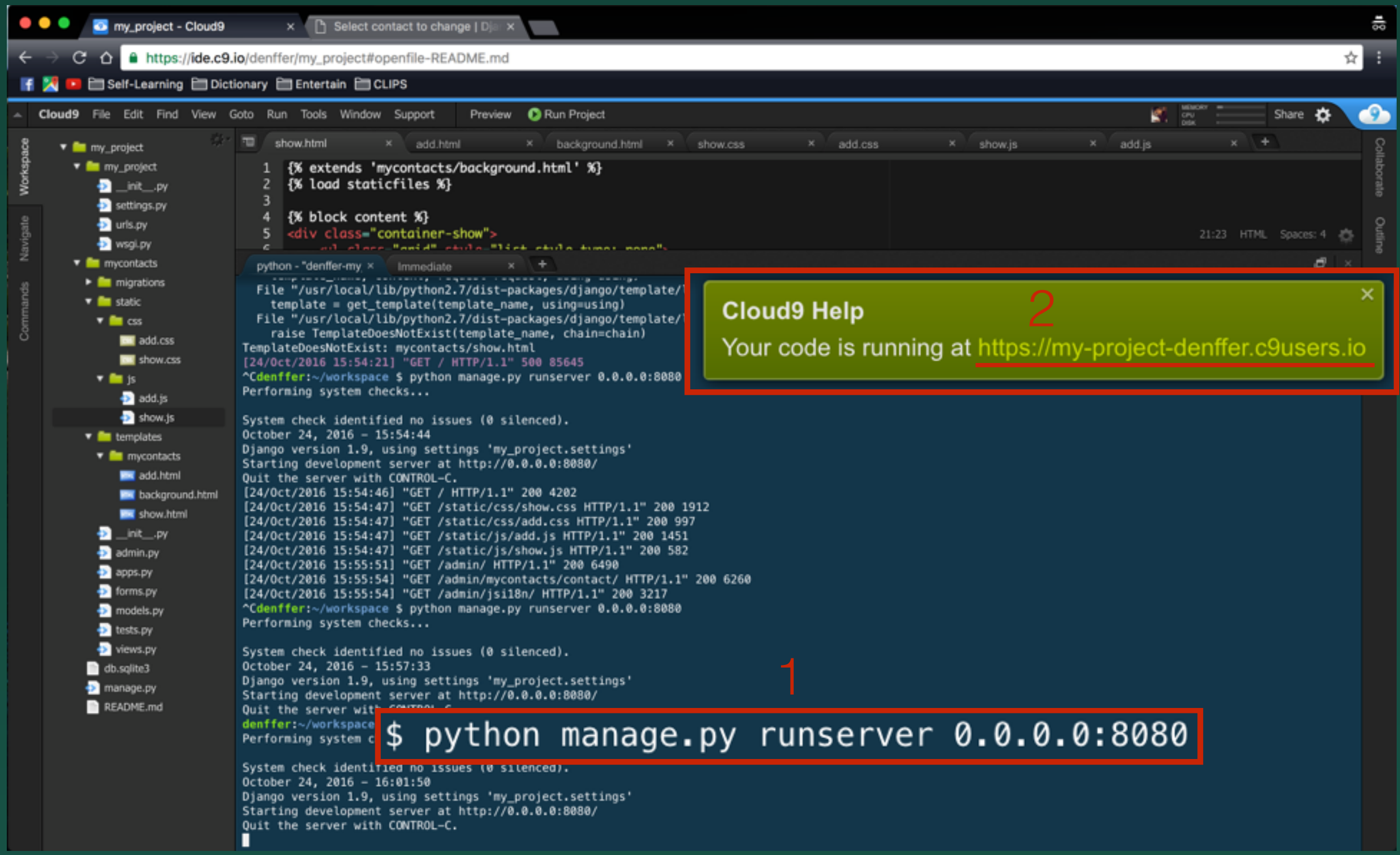
Respectively copy and paste the codes in the **JS** files from the repository you cloned from Github



Go to command line. Hit ‘control’ + ‘c’ to kill the server

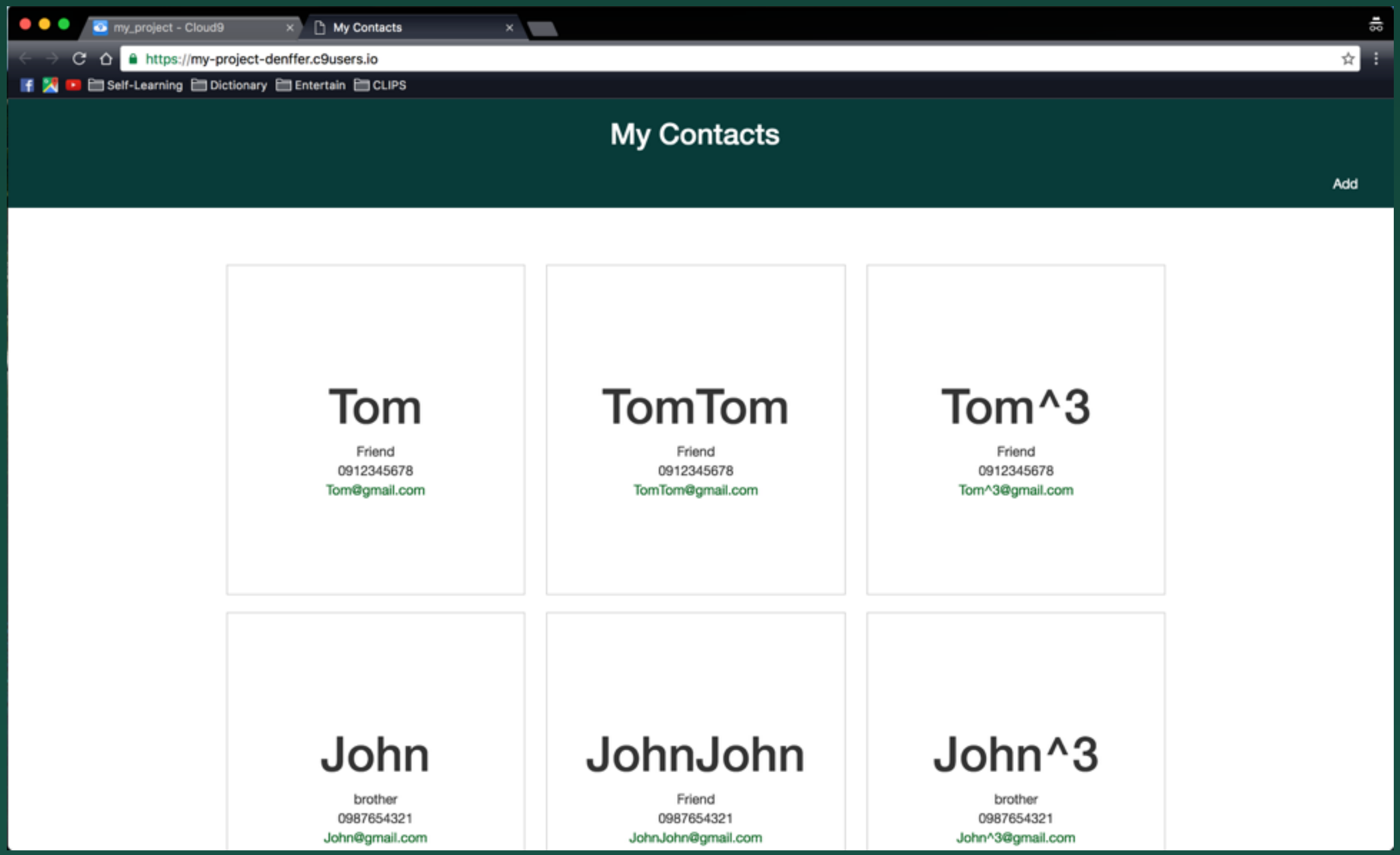


Start up the server again!



Ladies and gentlemen …

It worked! Feel free to play around!



Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. **Tutorial**
5. Conclusion

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. Tutorial
5. **Conclusion**

Conclusion

Alignment & Consistency

Two basic rules for a successful design

Conclusion

Error is your friend

Learn to embrace it

Conclusion

Simple is good

Try not to squeeze everything in a single page

Conclusion

Don’t play hero

Learn to cooperate with one another

Outline

1. Introduction
2. Basic Knowledge
3. Technical Details
4. Tutorial
5. **Conclusion**

Thank you

Presented by Denffer

Department of Computer Science, University of Taipei

Django

Build an organized web application

Presented by Denffer

2017@Department of Computer Science, University of Taipei