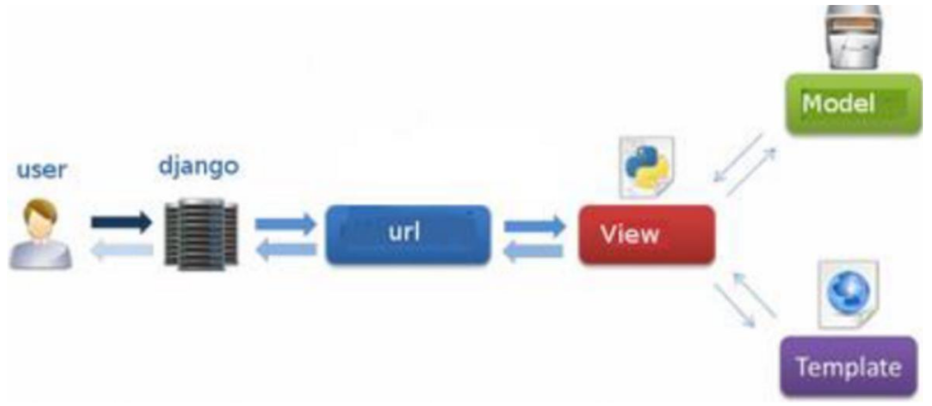
OS Lab 9 - Django

IULIANA MARIN

About Django

- Python based web framework.
- Uses the model-template-view architectural pattern.
  - Django manages the controller part for the interaction between the Model and the View.
  - The template is a HTML file with Django Template Language.
- Has search engine optimization by maintaining the website through URLs rather than IP addresses on the server.
- Is characterized by high scalability and security.



Installation

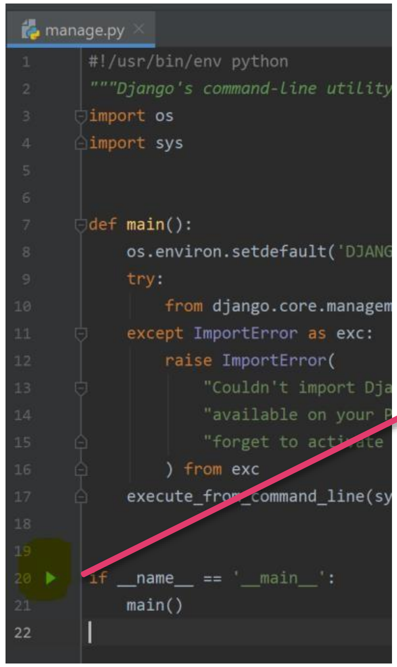
- You need to have Python already installed.
- Install pip:
  - On Windows:
    - Download get-pip.py (https://bootstrap.pypa.io/get-pip.py) to a folder on your computer.
    - Open a command prompt and navigate to the folder containing get-pip.py.
    - Run the following command: python get-pip.py
    - Pip is now installed!
  - On Ubuntu:
    - $ sudo apt-get update
    - $ sudo apt-get install python3-pip for version 3, $sudo apt-get install python-pip for version 2
    - Check the pip version using: $ pip3 –version (for version 3) OR $ pip –version (for version 2)
- Install Django using: pip install Django==3.0.5
- Install Pycharm integrated development environtment (IDE)
- Create a new project called auth inside the terminal of Pycharm: django-admin startproject auth

User Model

- Django has a model for managing users. The model is used for authentication.
- A user has as attributes:
  - username – required, has maximum 150 characters.
  - password – required. Django does not store raw password.
  - email – optional for email address which can be used for password resetting.
  - first_name – optional, has maximum 30 characters
  - last_name – optional, has maximum150 characters

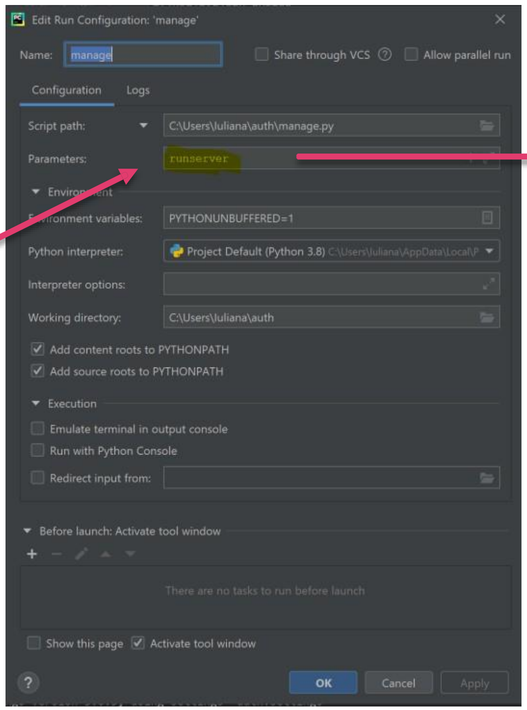
Test the application

- Open the terminal in PyCharm and run the command: python manage.py runserver
- An alternative is by opening the manage.py file, click on the green symbol, choose the Edit 'manage' option, add the runserver parameter and click on Apply, after which you can click on the green symbol.



```
#!/usr/bin/env python
"""Django's command-line utility
import os
import sys


def main():
    os.environ.setdefault('DJANG
    try:
        from django.core.managem
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Dja
            "available on your P
            "forget to activate
        ) from exc
    execute_from_command_line(sy


if __name__ == '__main__':
    main()

```

Choose Edit 'manage'



Name: manage
☐ Share through VCS ☐ Allow parallel run

Script path: C:\Users\Iuliana\auth\manage.py
Parameters: runserver

Environment
Environment variables: PYTHONUNBUFFERED=1
Python interpreter: Project Default (Python 3.8) C:\Users\Iuliana\AppData\Local\P
Interpreter options:
Working directory: C:\Users\Iuliana\auth
☑ Add content roots to PYTHONPATH
☑ Add source roots to PYTHONPATH

Execution
☐ Emulate terminal in output console
☐ Run with Python Console
☐ Redirect input from:

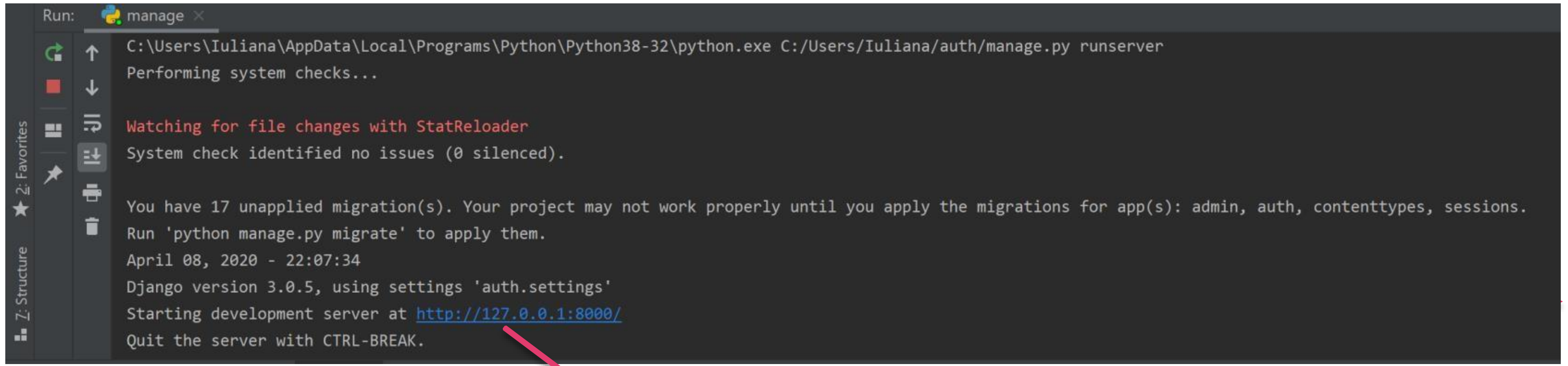
Before launch: Activate tool window
There are no tasks to run before launch
☐ Show this page ☑ Activate tool window

Add 'runserver' as parameter
Click on Apply

After this, click on the green symbol which appears in the image on the left.

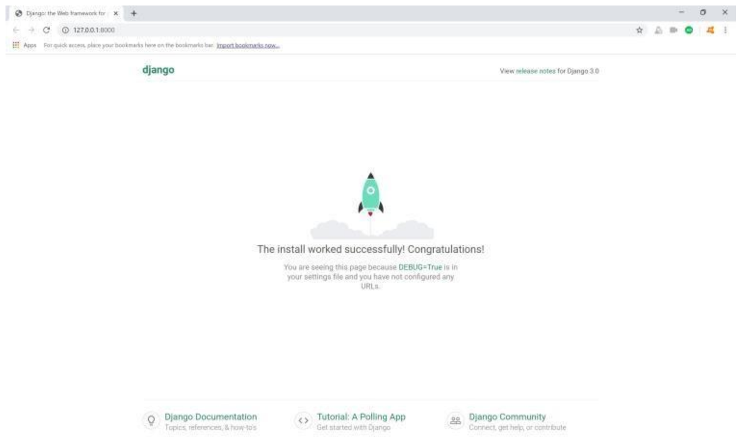
Test the application

- The output inside the terminal will be:



```
C:\Users\Iuliana\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Iuliana/auth/manage.py runserver
Performing system checks...

Watching for file changes with StatReloader
System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
April 08, 2020 - 22:07:34
Django version 3.0.5, using settings 'auth.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Open the link


Django: the Web framework for
127.0.0.1:8000
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
django
View release notes for Django 3.0

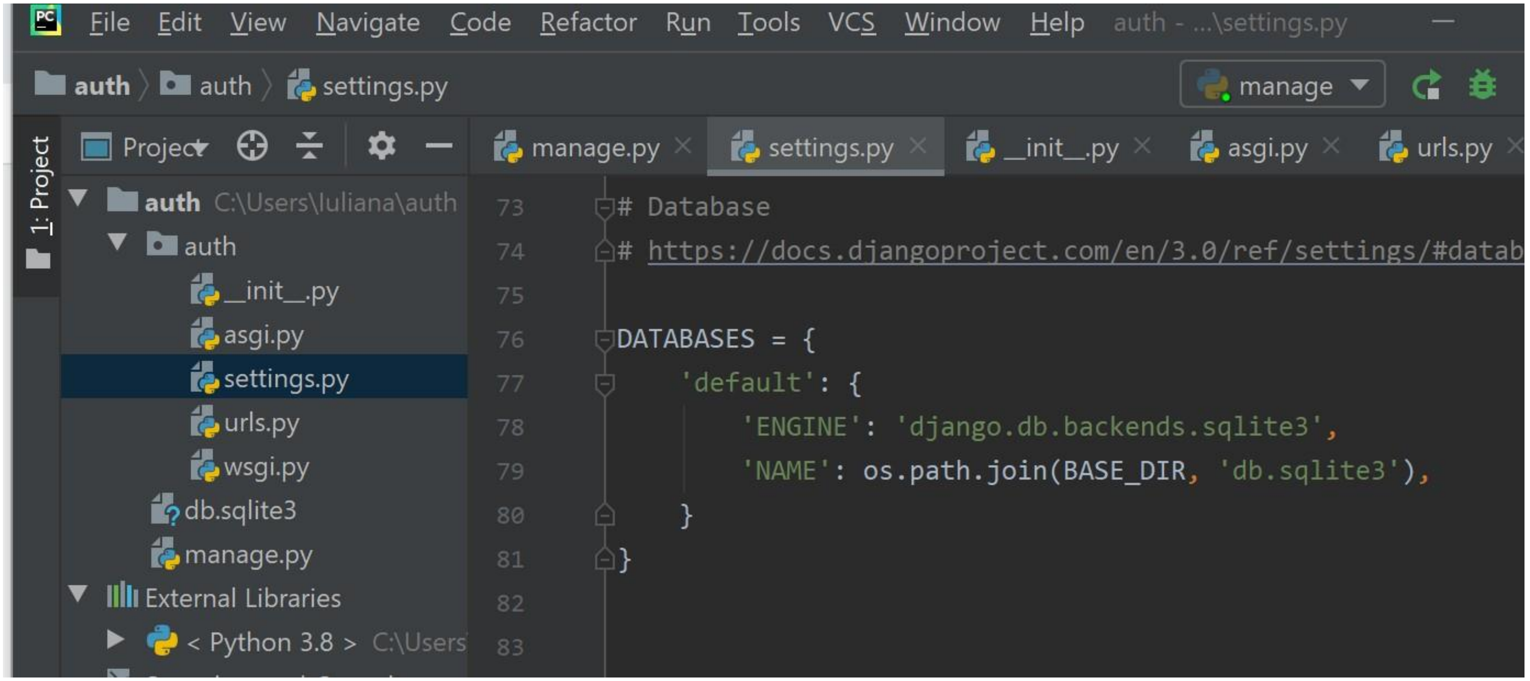
The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

Database

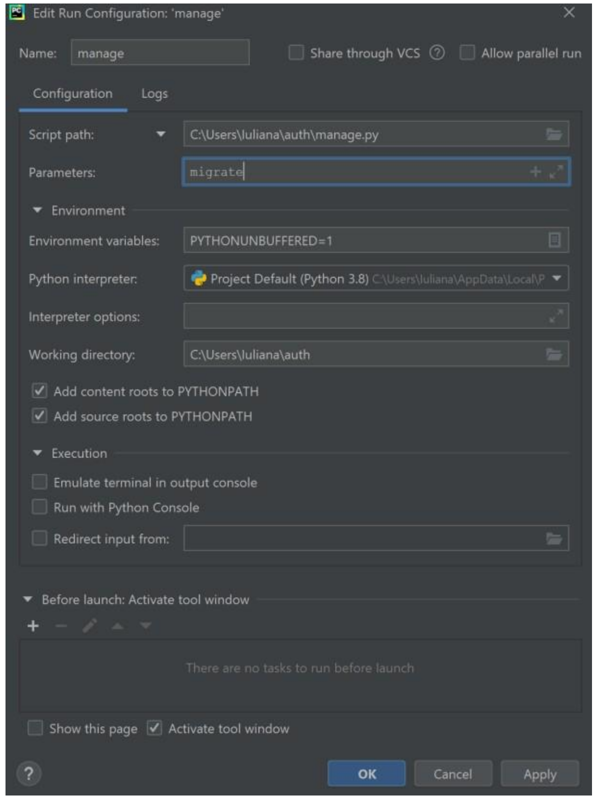
- The database setup is placed inside auth/settings.py. By default, Django uses SQLite.
- SQLite is already included in python and you don't need to install it.


File Edit View Navigate Code Refactor Run Tools VCS Window Help auth - ...\settings.py
auth auth settings.py
manage
Project
manage.py settings.py __init__.py asgi.py urls.py
auth C:\Users\Iuliana\auth
auth
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
External Libraries
< Python 3.8 > C:\Users
1: Project

```
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#datab

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Migration

- Propagates changes done on the models into the database schema. This is done by running the command: $ python manage.py migrate   OR Edit ‘manage’ and click run.



Edit Run Configuration: 'manage'

Name: manage
☐ Share through VCS
☐ Allow parallel run

Configuration | Logs

Script path: C:\Users\Iuliana\auth\manage.py
Parameters: migrate

Environment
Environment variables: PYTHONUNBUFFERED=1
Python interpreter: Project Default (Python 3.8) C:\Users\Iuliana\AppData\Local\P
Interpreter options:
Working directory: C:\Users\Iuliana\auth
☑ Add content roots to PYTHONPATH
☑ Add source roots to PYTHONPATH

Execution
☐ Emulate terminal in output console
☐ Run with Python Console
☐ Redirect input from:

Before launch: Activate tool window
There are no tasks to run before launch

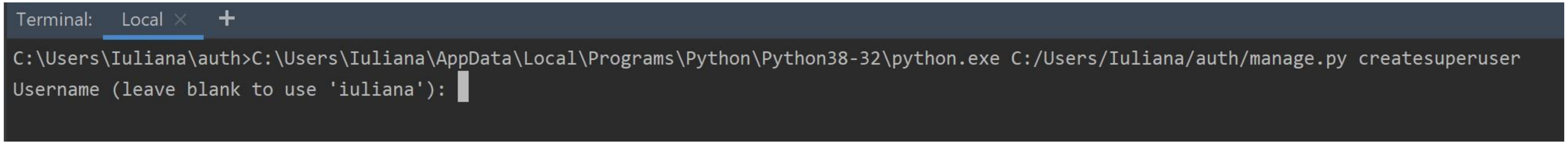
☐ Show this page ☑ Activate tool window

Create Django Admin

- Create admin website to add, edit and delete content:

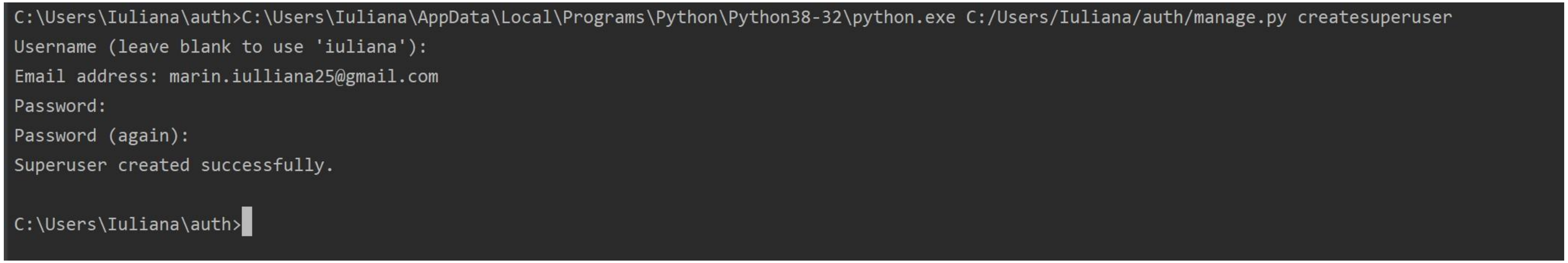
$ python manage.py createsuperuser OR By editing'manage'

- An alternative is by running in the terminal the retrieved command after editing manage.



```
C:\Users\Iuliana\auth>C:\Users\Iuliana\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Iuliana/auth/manage.py createsuperuser
Username (leave blank to use 'iuliana'):
```

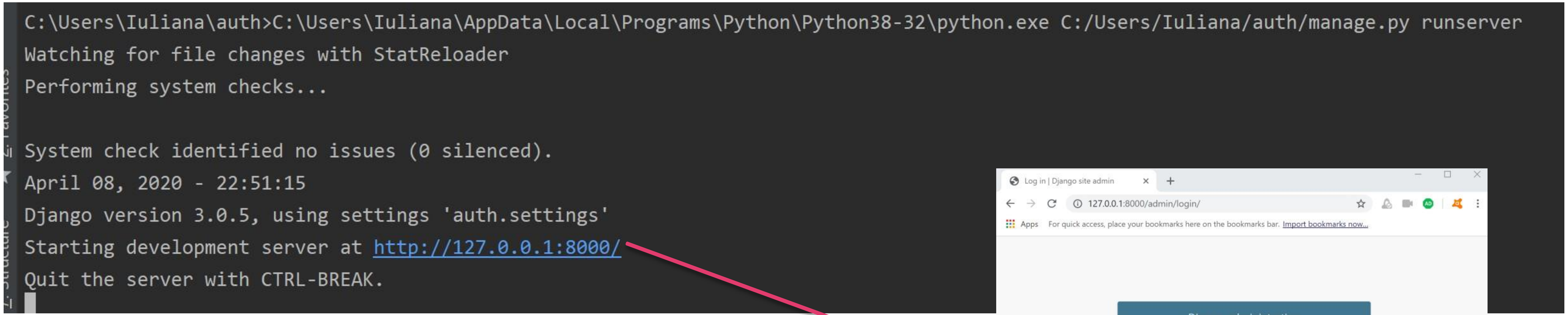
- Enter the desired username, email address and password for the application's superuser.



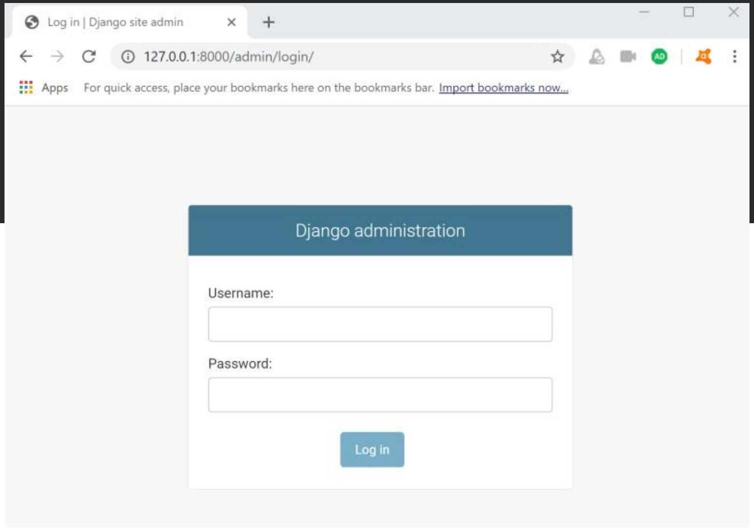
```
C:\Users\Iuliana\auth>C:\Users\Iuliana\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Iuliana/auth/manage.py createsuperuser
Username (leave blank to use 'iuliana'):
Email address: marin.iulliana25@gmail.com
Password:
Password (again):
Superuser created successfully.

C:\Users\Iuliana\auth>
```

Test the application

- Run the application by having runserver as an argument for the command:



```
C:\Users\Iuliana\auth>C:\Users\Iuliana\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Iuliana/auth/manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 08, 2020 - 22:51:15
Django version 3.0.5, using settings 'auth.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



Log in | Django site admin

127.0.0.1:8000/admin/login/

Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Django administration

Username:

Password:

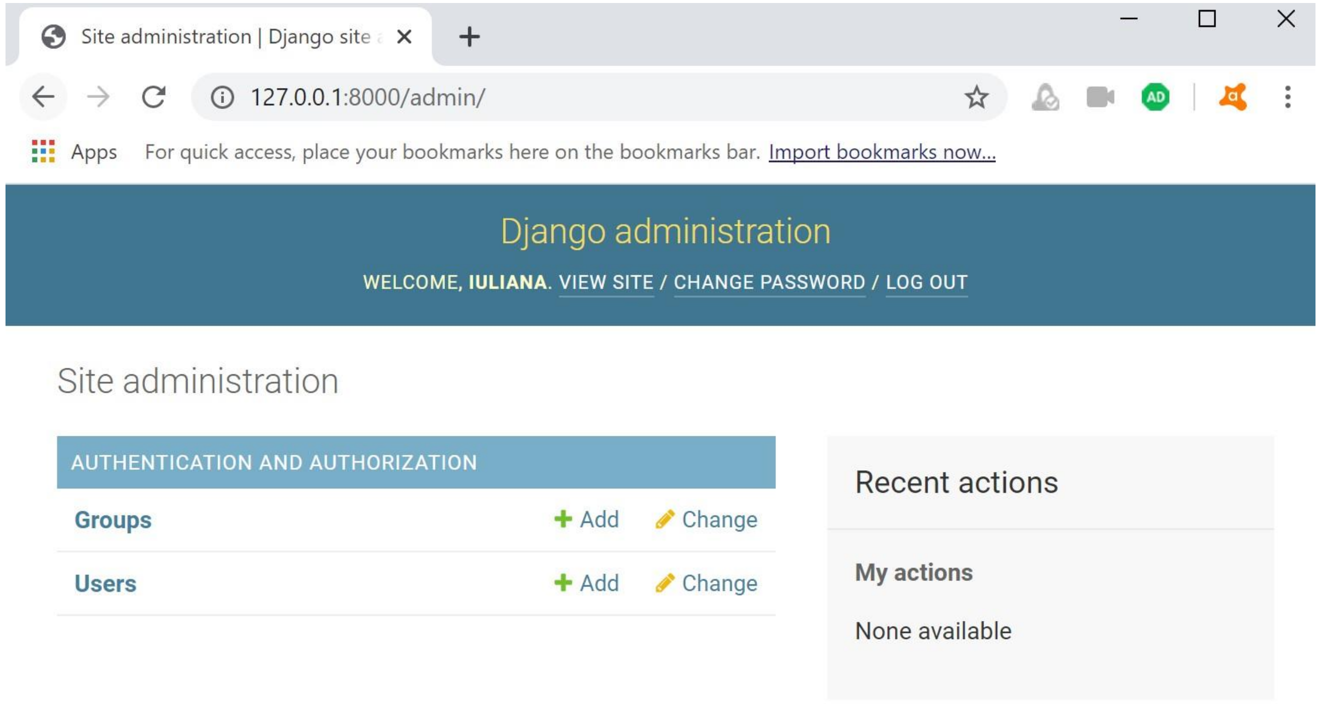
Log in

Access the admin page
http://127.0.0.1:8000/admin

Enter your just created credentials

Test the application

- After entering your credentials, you will have this view:


Site administration | Django site
127.0.0.1:8000/admin/
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Django administration

WELCOME, **IULIANA**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

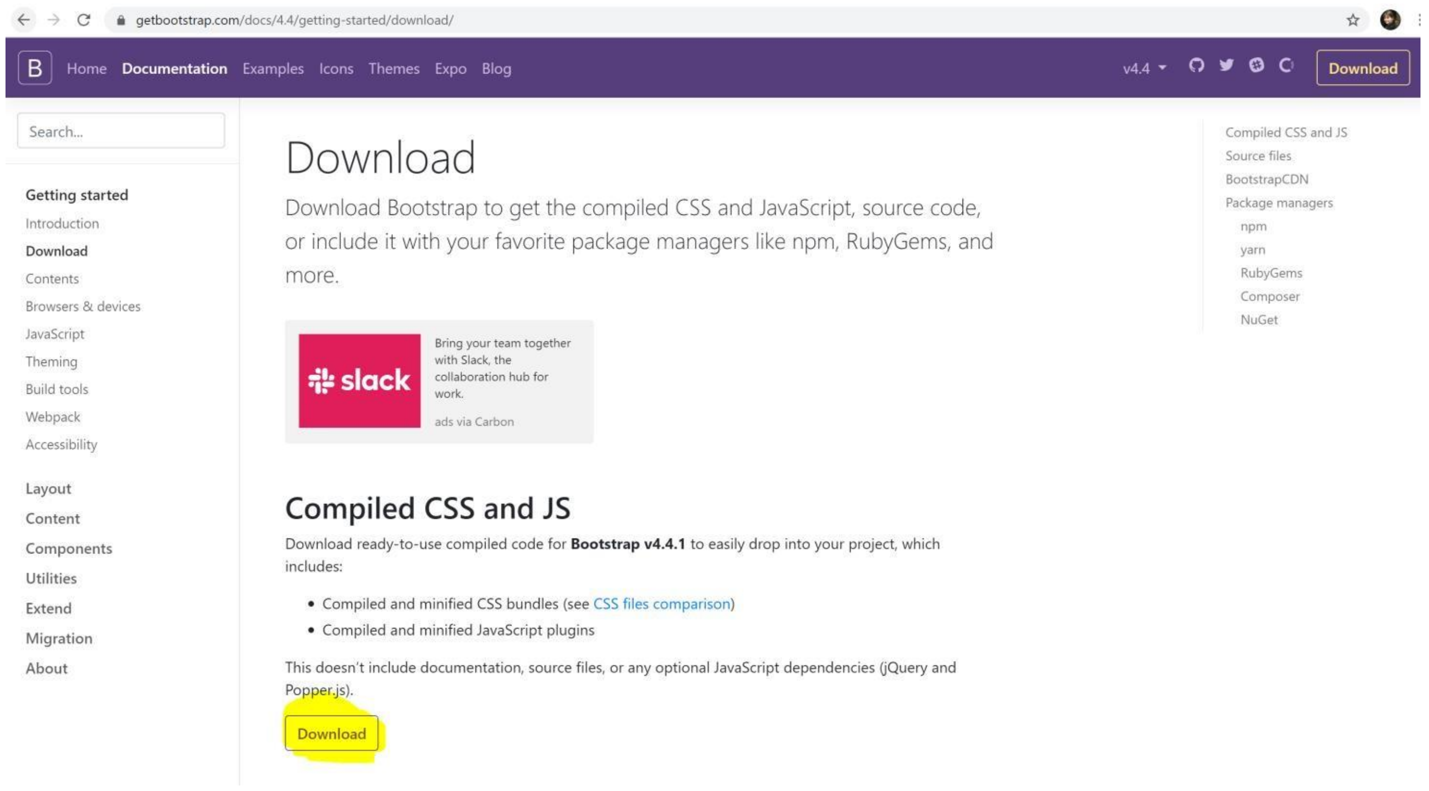
Recent actions

My actions

None available

Integrate Bootstrap in your application

- Download Bootstrap from the official website (https://getbootstrap.com/docs/4.4/getting-started/download/).



Download

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.4.1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see CSS files comparison)
- Compiled and minified JavaScript plugins

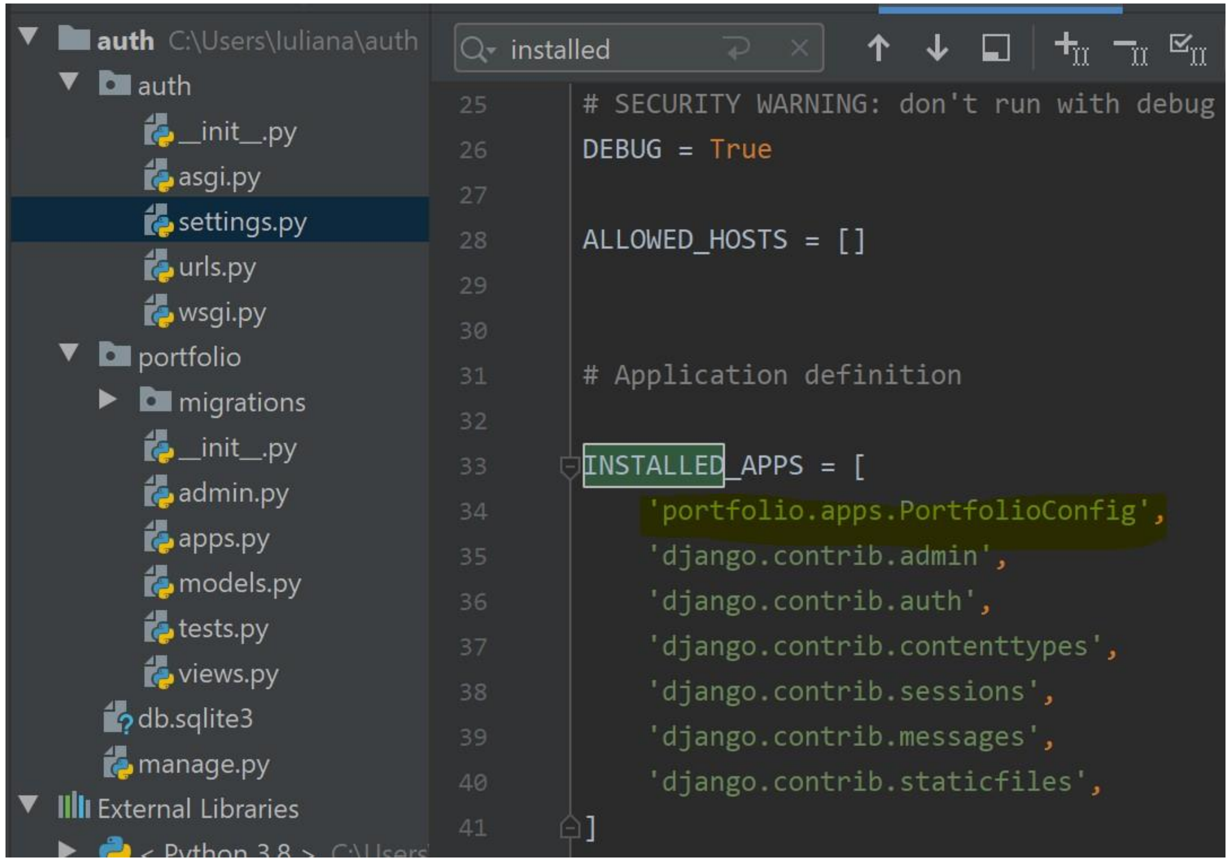
This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper.js).

Create a new application

- Create a new application called portfolio by running the command:

$ python manage.py startapp portfolio

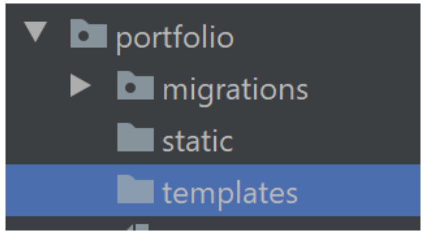
- PortfolioConfig is placed in the portfolio/apps.py file and the corresponding path is portfolio.apps.PortfolioConfig.
- Include the app in the project by adding a reference to the INSTALLED_APPS configuration in the auth/settings.py file. In this way, Django will include the portfolio app.



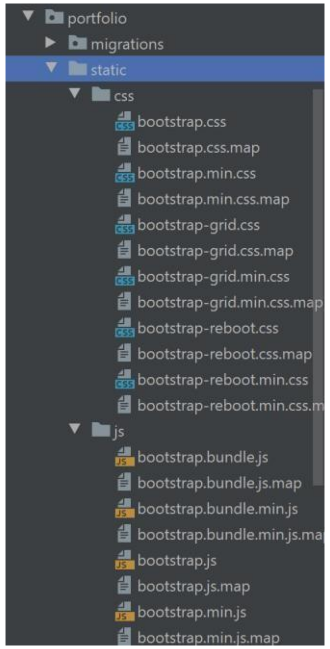
```
# SECURITY WARNING: don't run with debug
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'portfolio.apps.PortfolioConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Templates and files

- Django looks into the folder templates, as well as the static one for files.
- Create the static and templates folders under the portfolio app.
- Place the downloaded Boostrap files in the static folder like here:



```
▼ portfolio
  ▶ migrations
    static
    templates
```



```
▼ portfolio
  ▶ migrations
  ▼ static
    ▼ css
        bootstrap.css
        bootstrap.css.map
        bootstrap.min.css
        bootstrap.min.css.map
        bootstrap-grid.css
        bootstrap-grid.css.map
        bootstrap-grid.min.css
        bootstrap-grid.min.css.map
        bootstrap-reboot.css
        bootstrap-reboot.css.map
        bootstrap-reboot.min.css
        bootstrap-reboot.min.css.m
    ▼ js
        bootstrap.bundle.js
        bootstrap.bundle.js.map
        bootstrap.bundle.min.js
        bootstrap.bundle.min.js.ma
        bootstrap.js
        bootstrap.js.map
        bootstrap.min.js
        bootstrap.min.js.map
```

Templates and files

- Create a template called base.html inside the templates folder, with the following code:

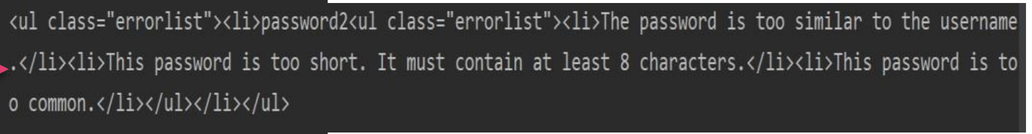
```
<!doctype html>
{% load static %}
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}" crossorigin="anonymous">
    <title>Django Authentication System - HLAB</title>
  </head>
  <body>
    {% block content %}
    {% endblock %}
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
    <script src="{% static 'js/bootstrap.bundle.min.js' %}"></script>
  </body>
</html>
```

User creation form

- Create a user creation form which inherits the ModelForm class. Edit the portfolio/views.py file and add the following lines of code:

```python
from django.shortcuts import render
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import login, authenticate
from django.http import HttpResponse
def sign_up(request):
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        print(form.errors)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password1')
            user = authenticate(username=username, password=password)
            login(request, user)
            return HttpResponse('A new user has been successfully registered!')
    else:
        form = UserCreationForm()
    return render(request, 'register.html', {'form': form})
```

Can lead to erros like:



```
<ul class="errorlist"><li>password2<ul class="errorlist"><li>The password is too similar to the username
.</li><li>This password is too short. It must contain at least 8 characters.</li><li>This password is to
o common.</li></ul></li></ul>
```

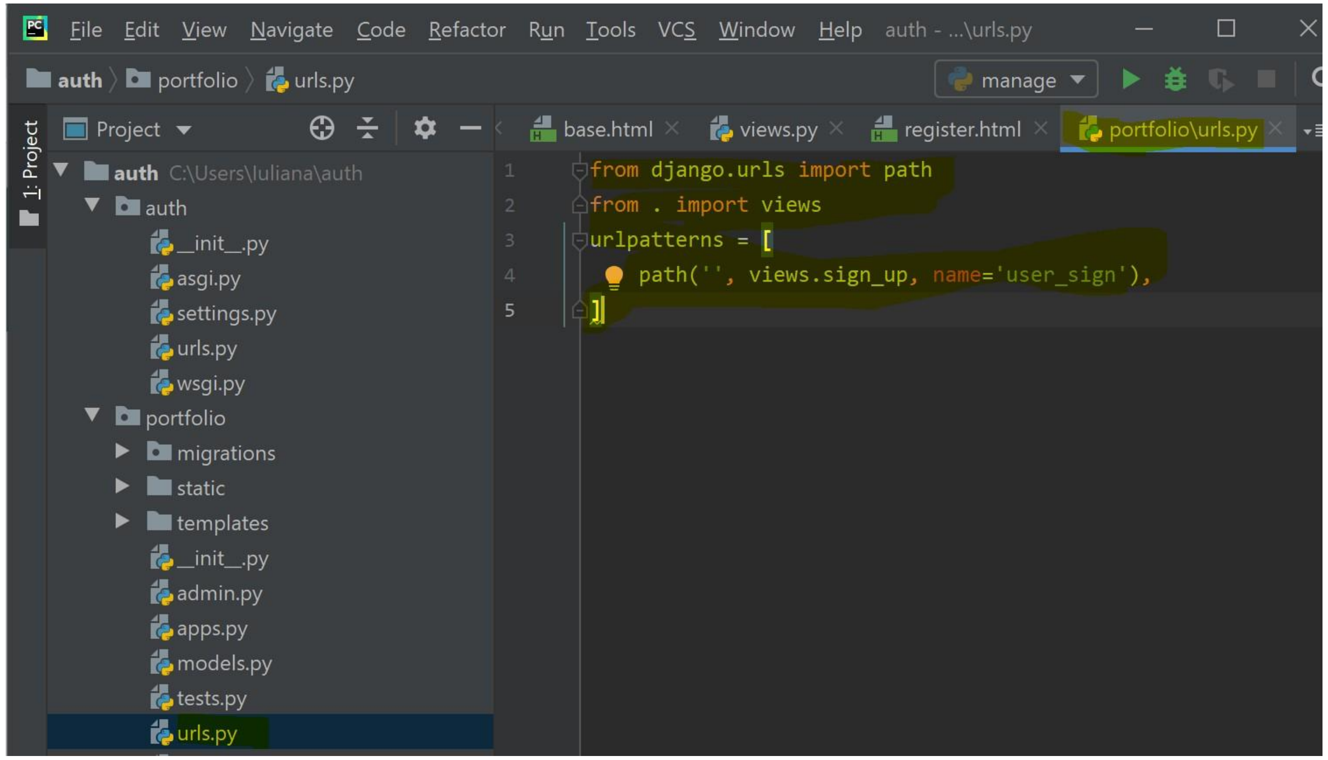
Creation of register.html

- Create the register.html file inside the template folder:

```
{% extends 'base.html' %}
{% block content %}
    <div class="container">
    <div class="row">
        <br>
        <br>
    </div>
    <div class="row">
        <div class="col-md-6 offset-3">
            <form method="post">
                {% csrf_token %}
              <div class="form-group">
                <label for="exampleInputEmail1">Username</label>
                <input type="text" class="form-control" name="username" id="exampleInputEmail1" placeholder="Enter username" required>
              </div>
              <div class="form-group">
                <label for="exampleInputPassword1">Password</label>
                <input type="password" class="form-control" name="password1" id="exampleInputPassword1" placeholder="Password" required>
              </div>
                 <div class="form-group">
                <label for="exampleInputPassword2">Confirm Password</label>
                <input type="password" class="form-control" name="password2" id="exampleInputPassword2" placeholder="confirm password"
                required>
              </div>
              <button type="submit" class="btn btn-primary">Submit</button>
            </form>
        </div>
    </div>
    </div>
{% endblock %}
```

Creation of Uniform Resource Locators (URL)s

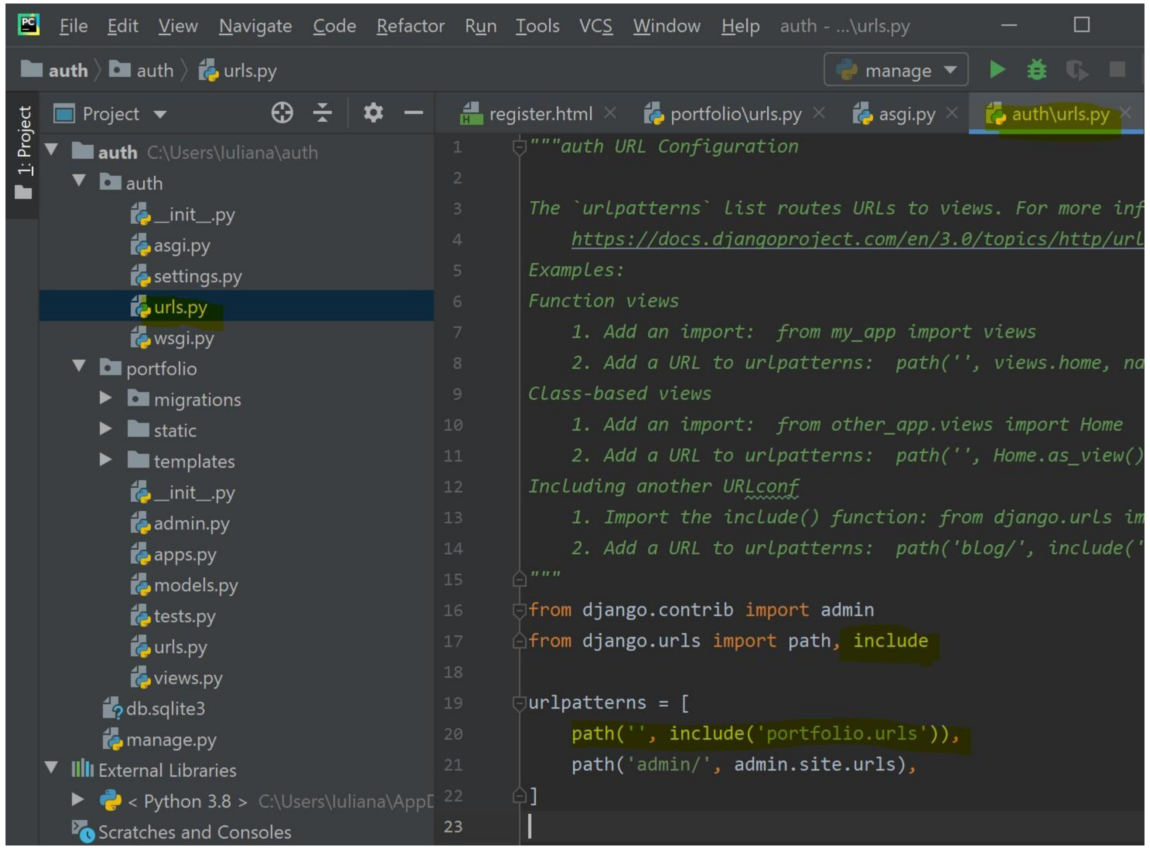
- Create the urls.py file in the portfolio app and add the following lines of code which import the views in the portfolio app and map the view to a url:



```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.sign_up, name='user_sign'),
]
```

Creation of Uniform Resource Locators (URL)s

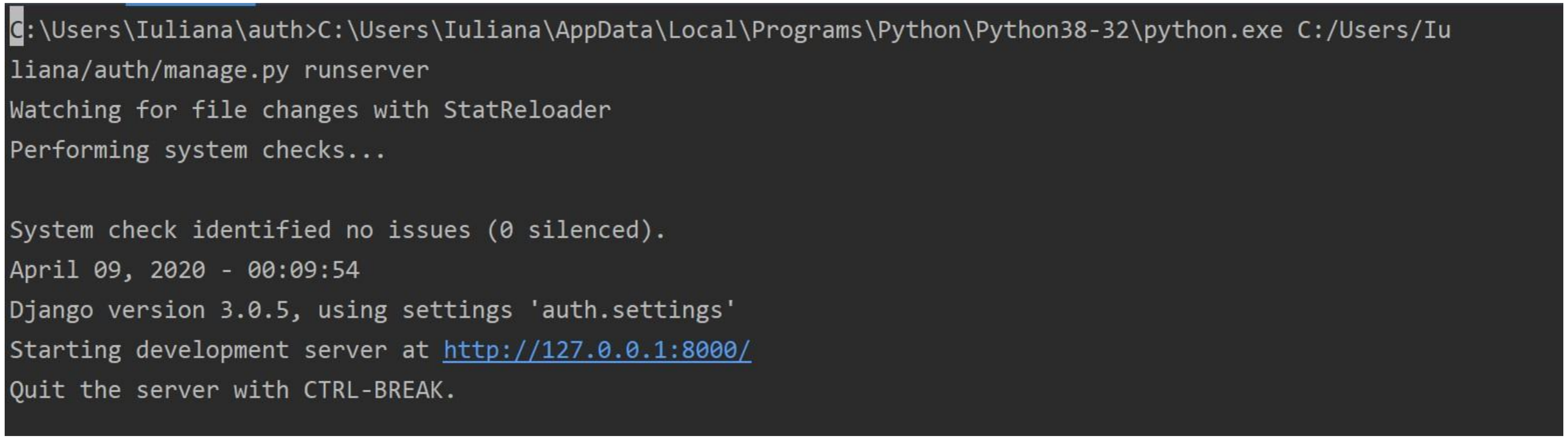
- Edit the auth/urls.py file and include the URL for portfolio.



```
"""auth URL Configuration

The `urlpatterns` list routes URLs to views. For more inf
    https://docs.djangoproject.com/en/3.0/topics/http/url
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, na
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view()
Including another URLconf
    1. Import the include() function: from django.urls im
    2. Add a URL to urlpatterns:  path('blog/', include('
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('', include('portfolio.urls')),
    path('admin/', admin.site.urls),
]

```

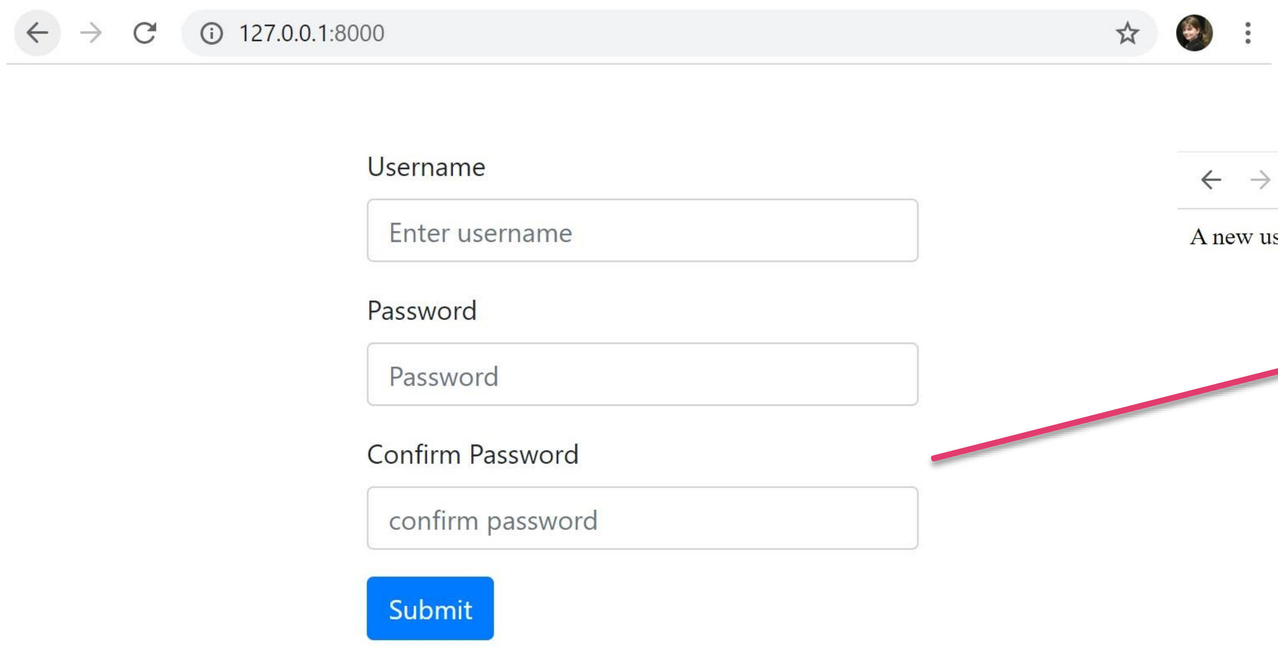
Test the application

- Run your application by having runserver as an argument of the manage.py file.



```
C:\Users\Iuliana\auth>C:\Users\Iuliana\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Iu
liana/auth/manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 09, 2020 - 00:09:54
Django version 3.0.5, using settings 'auth.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Test the application


127.0.0.1:8000

Username

Enter username

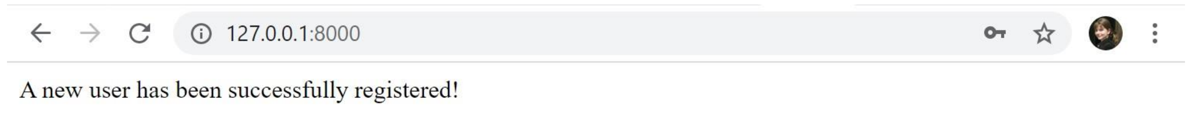
Password

Password

Confirm Password

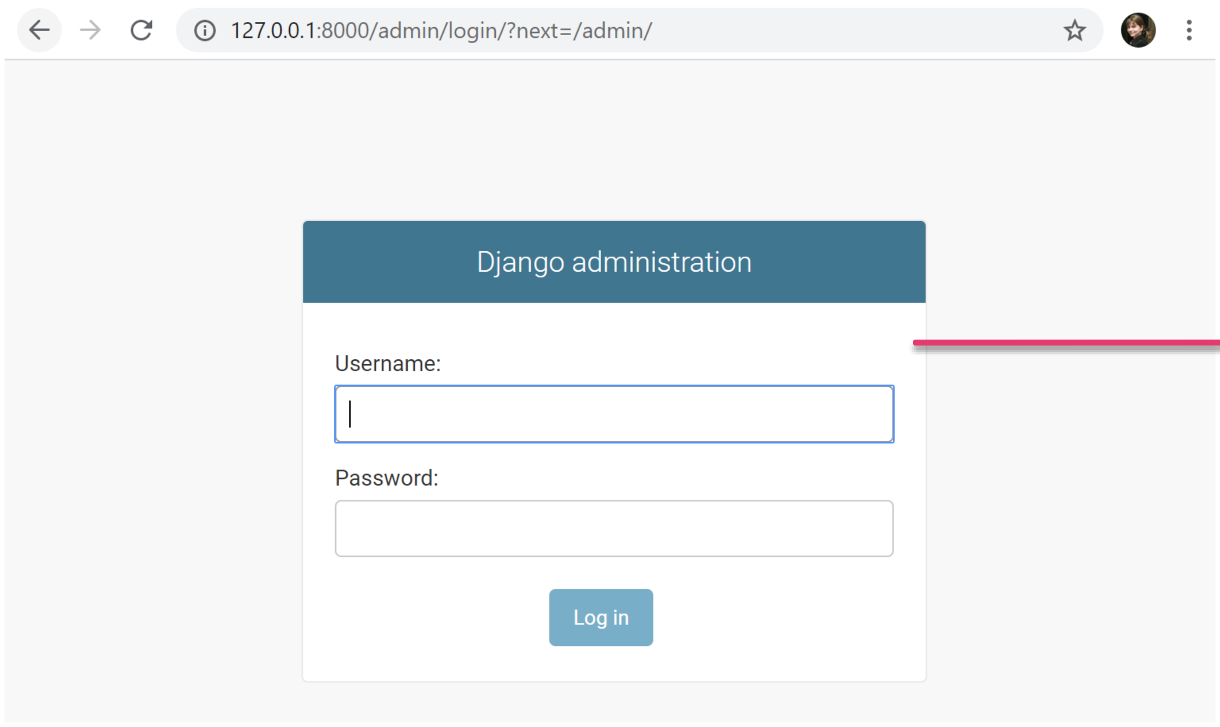
confirm password

Submit


127.0.0.1:8000

A new user has been successfully registered!

Test the application
Enter the admin page and see the new created user

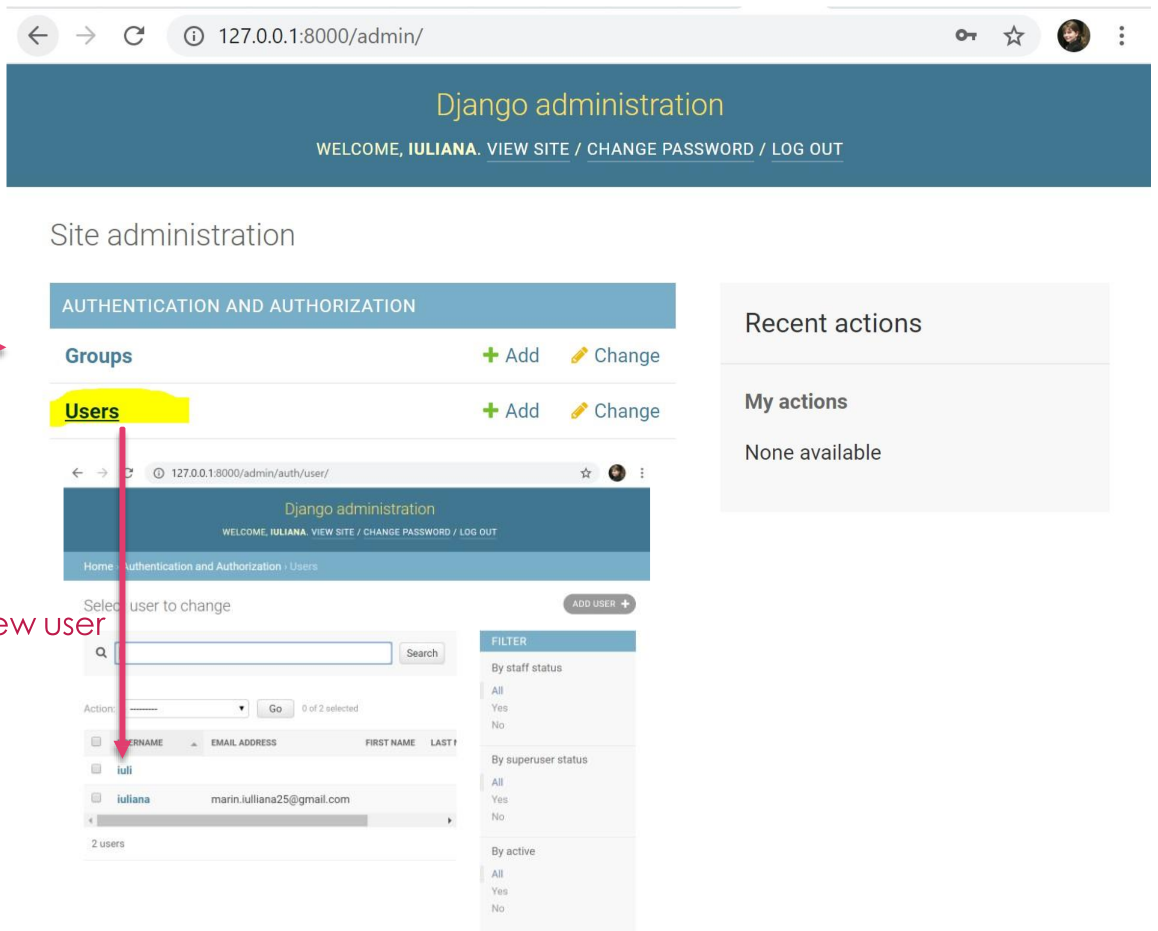


127.0.0.1:8000/admin/login/?next=/admin/

Django administration

Username:

Password:



127.0.0.1:8000/admin/

Django administration

WELCOME, **IULIANA**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

Recent actions

My actions

None available

127.0.0.1:8000/admin/auth/user/

Django administration

WELCOME, **IULIANA**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Users

Selec user to change

0 of 2 selected

ERNAME	EMAIL ADDRESS	FIRST NAME	LAST
iuli			
iuliana	marin.iulliana25@gmail.com		

2 users

FILTER

By staff status
- All
- Yes
- No

By superuser status
- All
- Yes
- No

By active
- All
- Yes
- No

New user

Congratulations!
You have finished your first Django app!