Web Apps w/ Django

ANDREW AND SKYLAR

What are web apps

Software accessible by any web browser that has internet connection

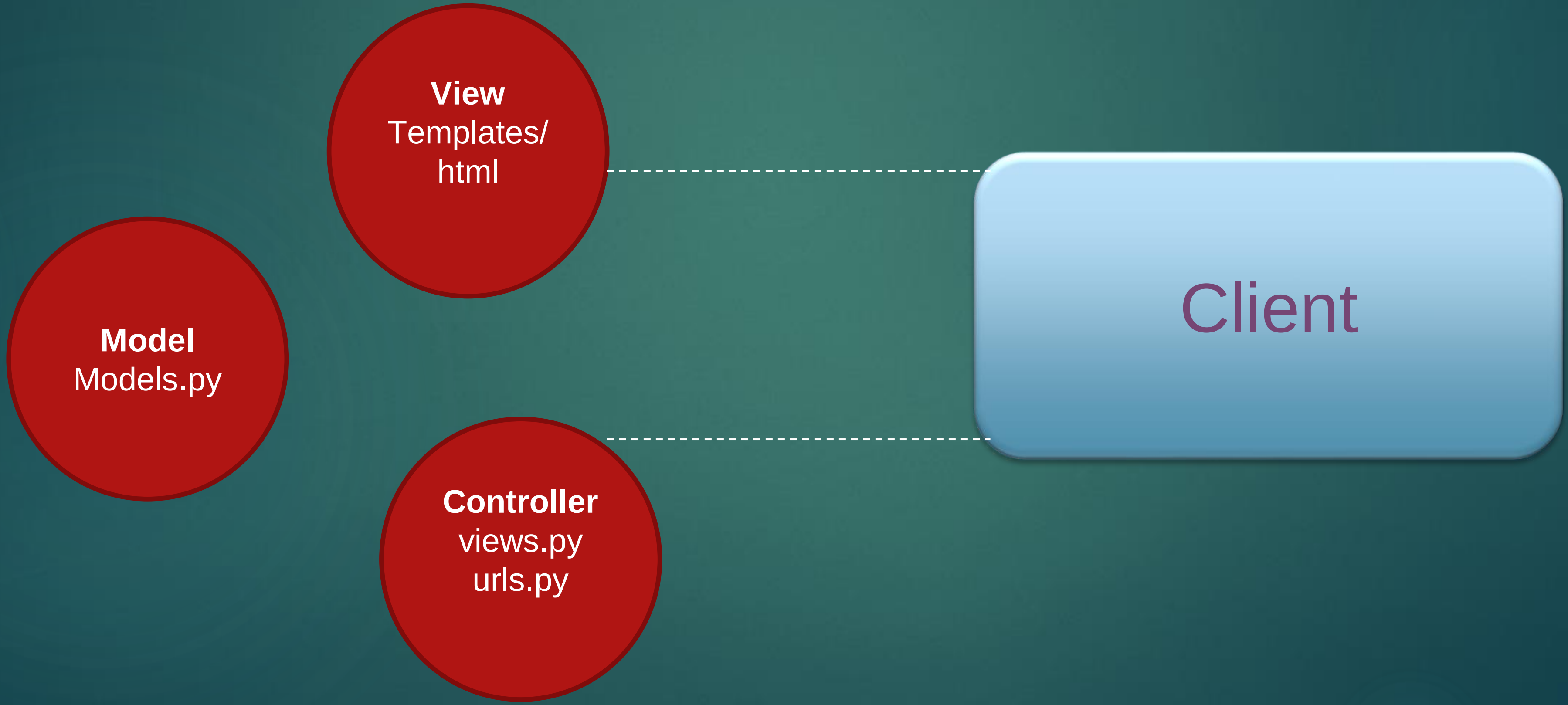
- All websites are web apps!

Django is a python framework

Module vs Framework

- Modules provide you functions that you call
- You provide functions for the framework to call

Model View Controller Framework



View
Templates/
html

Model
Models.py

Controller
views.py
urls.py

Client

Creating a Django project

Installation: type in your command prompt:

```
pip install django==3.0.2
```

Create project directory (don’t forget the dot at the end!)

```
django-admin startproject webapps .
```

Create the app

```
python manage.py startapp appName
```

Creating a Django project (p2)

Create the database

- python manage.py migrate

Go into webapps/settings.py

- scroll to the list INSTALLED_APP and add ‘appName’

Run the app:

- python manage.py runserver

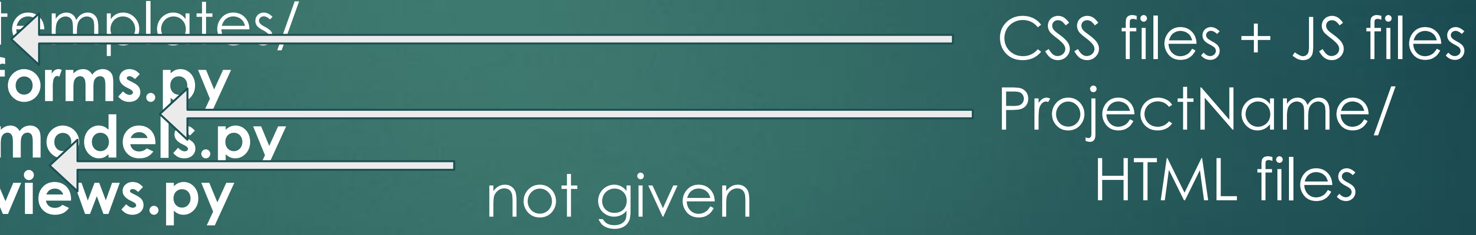
Type in chrome:

- localhost:8000

Structure

- projectFolder
  |-- README.md
- |-- webapps/
  |-- **settings.py**
  |-- **urls.py**
- |-- projectName/
  |-- static/
  |-- templates/
  |-- **forms.py**
  |-- **models.py**
  |-- **views.py**

|-- **manage.py**



CSS files + JS files

ProjectName/
HTML files

not given

HTML and CSS

- HTML - HyperText Markup Language
  - Structure of the website
  - Buttons, Searchbars, any website element

- CSS - Cascading Style Sheets
  - Style of the website
  - decoration for different HTML elements

- Bootstrap - framework for website design
  - gives you HTML and CSS so you don’t have to build it yourself

Controller

- All the controller functions exist inside “views.py” (bad name!)

- request -> data the client gives you (the server)
- context -> the data that you (server) want to give the client
  - context and request can be treated as python dictionaries

```
# Example:
def login(request):
        context = {‘data’: ‘hello’}
        return render(request, “appName/login.HTML”, context)
```

HTTP

HTTP - Hypertext Transfer Protocol - enables communication between server and client

POST request

- Send information to the server (when you login or register)

GET request

- Request data from server, but doesn’t modify server data

More information here:

https://www.w3schools.com/tags/ref_httpmethods.asp

Django Templating Language

- When you pass information to the client, how do you use it in HTML of the page dynamically?

- For variables use 2 curly braces {{ variable_name }}

- For tags, use {% tag %}

{% If condition %}
Some HTML
{% endif %}
	{% for item in list %}
	Some HTML, possibly involving {{item}}
	{% endfor %}

Forms

- HTML element that sends information to the server
- Specify the method type (post or get)
- HTML element <button> will send information

```
<button name = "button" value = "0" > 0 </button>
```

- Hidden fields - information you want to send to the client but not show

```
<input type="hidden" name="storedInt" value="{{storedInt}}">
```

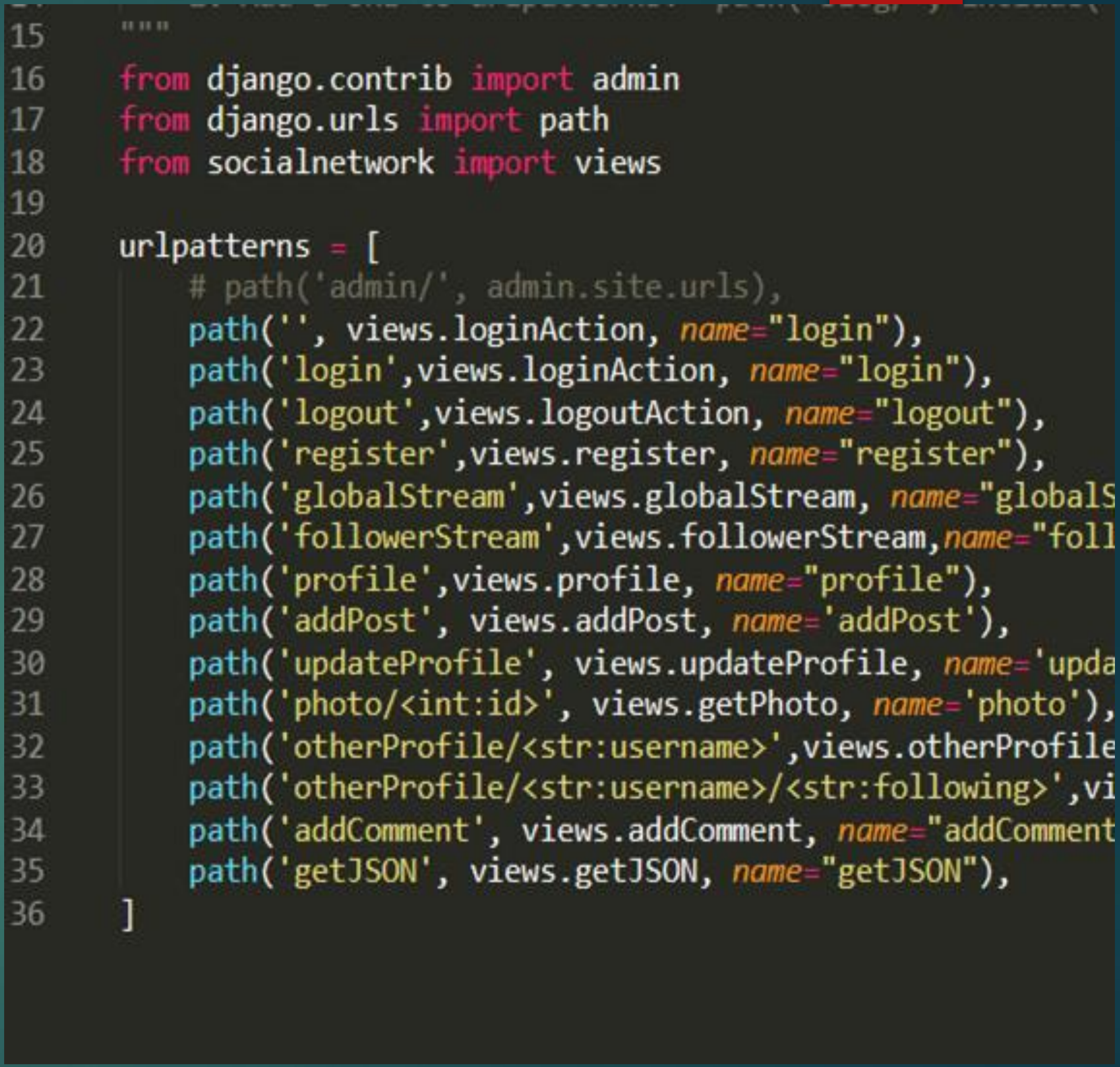
Cross Site Request Forgery

- What are CSRF attacks?
- Django will check to make sure that the CSRF token is valid
- You must pass in a CSRF token in your form (otherwise you'll get an error)
- Only prevents against POST request. (why not GET requests?)

```
{% csrf_token %}
```

URL Routes

- urls.py file keeps track of all the pages in your app (list of mappings)
- When URL has nothing in it (localhost:8000) it will go to views.loginAction (line 22)



```
15      """
16      from django.contrib import admin
17      from django.urls import path
18      from socialnetwork import views
19
20      urlpatterns = [
21          # path('admin/', admin.site.urls),
22          path('', views.loginAction, name="login"),
23          path('login',views.loginAction, name="login"),
24          path('logout',views.logoutAction, name="logout"),
25          path('register',views.register, name="register"),
26          path('globalStream',views.globalStream, name="globalS
27          path('followerStream',views.followerStream,name="foll
28          path('profile',views.profile, name="profile"),
29          path('addPost', views.addPost, name='addPost'),
30          path('updateProfile', views.updateProfile, name='upda
31          path('photo/<int:id>', views.getPhoto, name='photo'),
32          path('otherProfile/<str:username>',views.otherProfile
33          path('otherProfile/<str:username>/<str:following>',vi
34          path('addComment', views.addComment, name="addComment
35          path('getJSON', views.getJSON, name="getJSON"),
36      ]
```

Calculator App

https://github.com/anddhong/DjangoCalculator

Conclusion

- If you’re interested in Web Apps, consider taking

  17-437: Web Application Development

- Attendance