Sistemas de Negocios

Clase 2B: More programming with django

September 21, 2016

David Olivieri
Uvigo, E.S. Informatica

outline

- Another example with templates.
- Improve the interface
- Use Bootstrap

Improve the template

- We want to improve the template that we did previously.

Static Files

- Before implementing Bootstrap as a template, we first need to server “static” files correctly.
- Static files:
  - What will be rendered by browser and not used directly by django code
  - CSS, JavaScript, Images

Change “settings.py”

If the DEBUG option, then we will define
certain directories and global variables:

```
If DEBUG:
        MEDIA_URL    =  ‘/media/’
        STATIC_ROOT = os.path.join(os.path.dirname(BASE_DIR), “static”, “static-only”)
        MEDIA_ROOT = os.path.join(os.path.dirname(BASE_DIR), “static”, “media”)
        STATICFILES_DIRS = (
                os.path.join(os.path.dirname(BASE_DIR), “static”, “static”),
        )
```

We will see how to configure “live” static files,
but these definitions are for the local computer
configurations

Serve static files (Local only)

- Static files are CSS, JavaScript, and Images

```
# Static files (CSS, JavaScript, Images)

STATIC_URL = ‘/static/’

# Template location
TEMPLATE_DIRS = (
   os.path.join(os.path.dirname(BASE_DIR), “static”, “templates”),
)

If DEBUG:
       MEDIA_URL    =  ‘/media/’
       STATIC_ROOT = os.path.join(os.path.dirname(BASE_DIR), “static”, “static-only”)
       MEDIA_ROOT = os.path.join(os.path.dirname(BASE_DIR), “static”, “media”)
       STATICFILES_DIRS = (
                     os.path.join(os.path.dirname(BASE_DIR), “static”, “static”),
       )
```

Create the static dirs

- Now that they are defined previously in settings, you need to create the static directories.

Modification of the “url.py”

- Need to be able to use these variables.

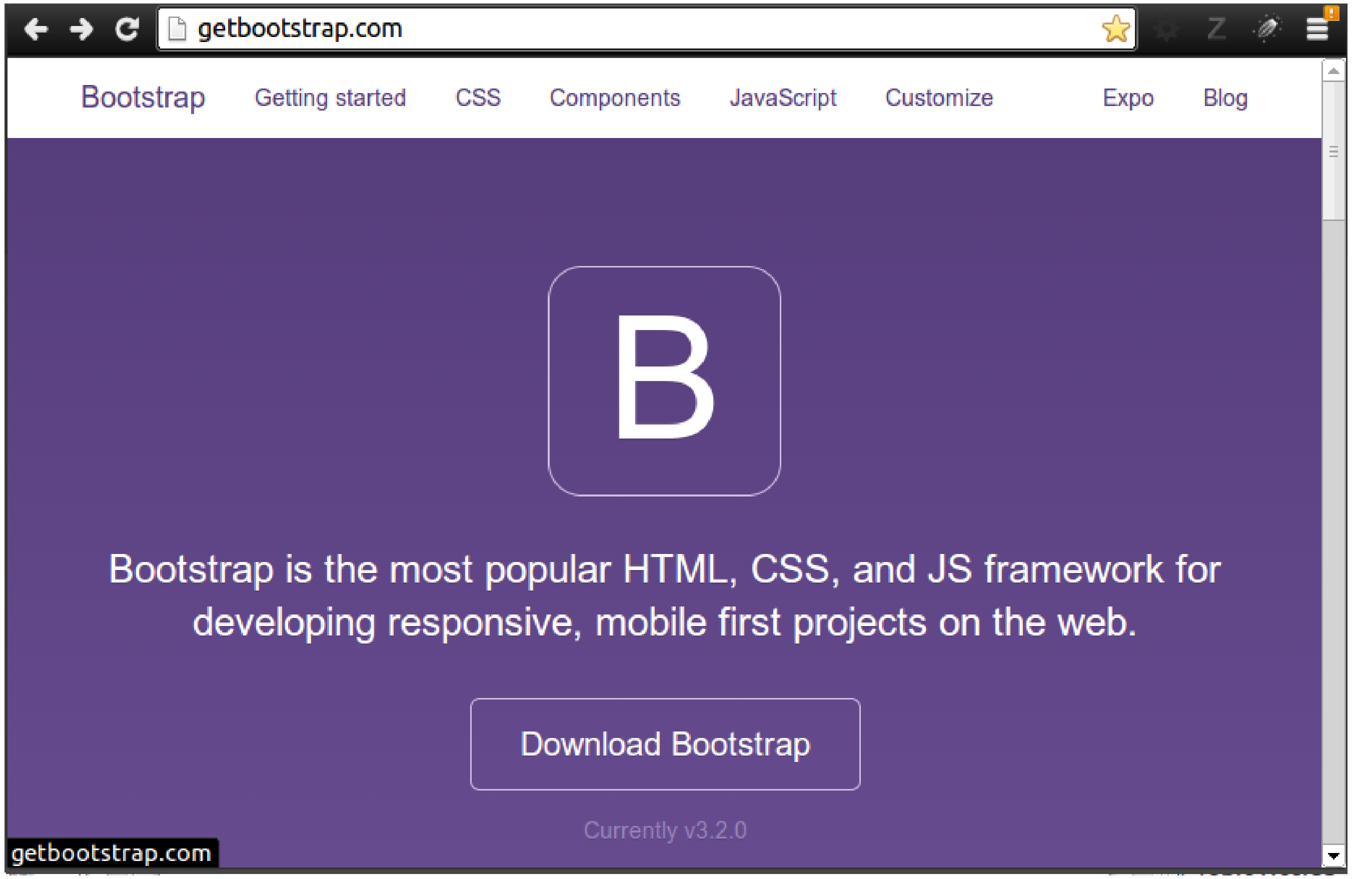
```
from django.conf.urls import patterns, include, url
from django.conf import settings
from django.conf.urls.static import static

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    url(r'^$', 'signups.views.home', name='home'),
    # url(r'^blog/', include('blog.urls')),

    url(r'^admin/', include(admin.site.urls)),
)

if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL,
                          document_root=settings.STATIC_ROOT)

    urlpatterns += static(settings.MEDIA_URL,
                          document_root=settings.MEDIA_ROOT)
```

Now manage the static files

At the terminal prompt, type the following:

```
$ python  manage.py  collectstatic
```

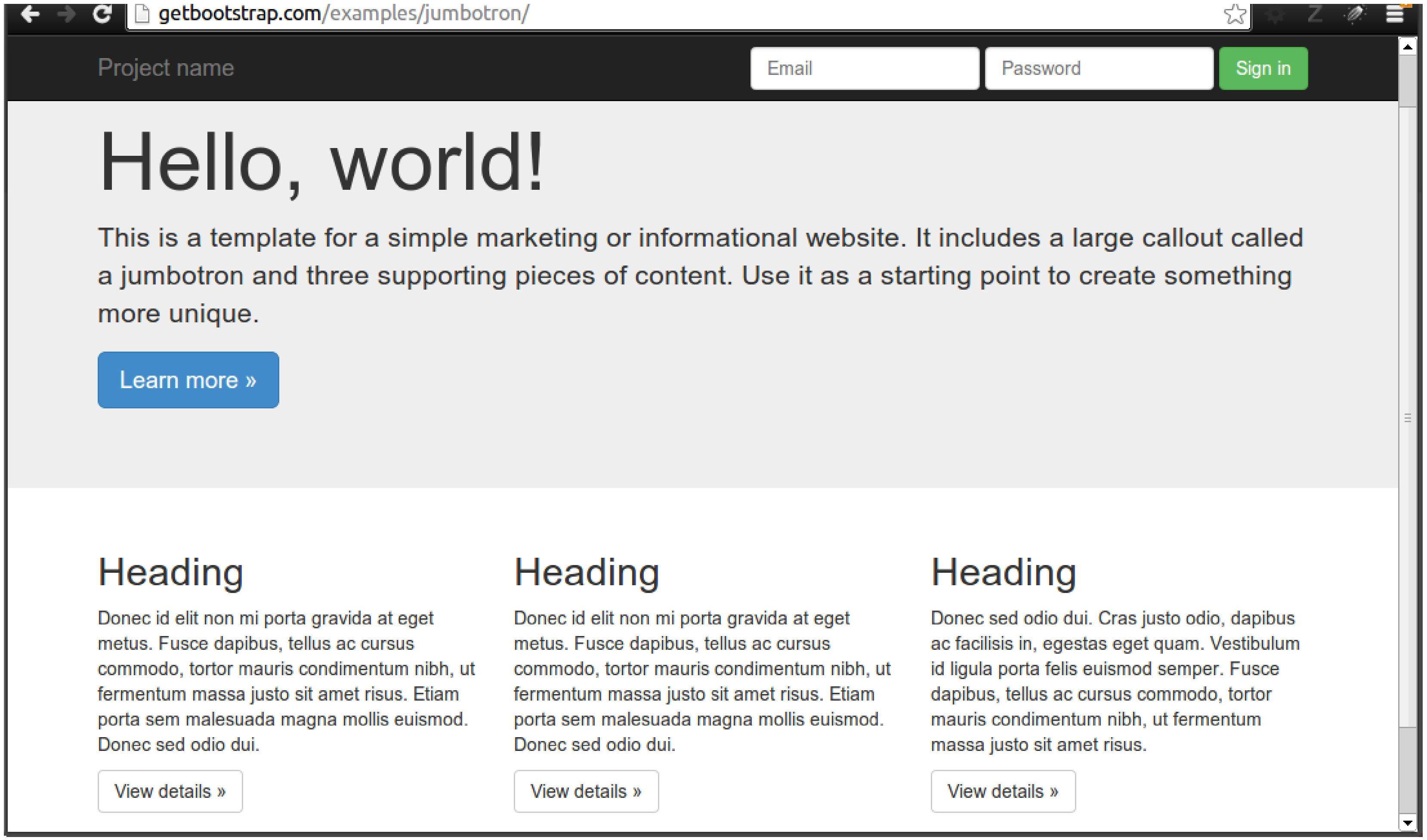
Bootstrap



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Jumbletron

- http://getbootstrap.com/examples/jumbotron/


getbootstrap.com/examples/jumbotron/

Project name

Email

Password

Sign in

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Learn more »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

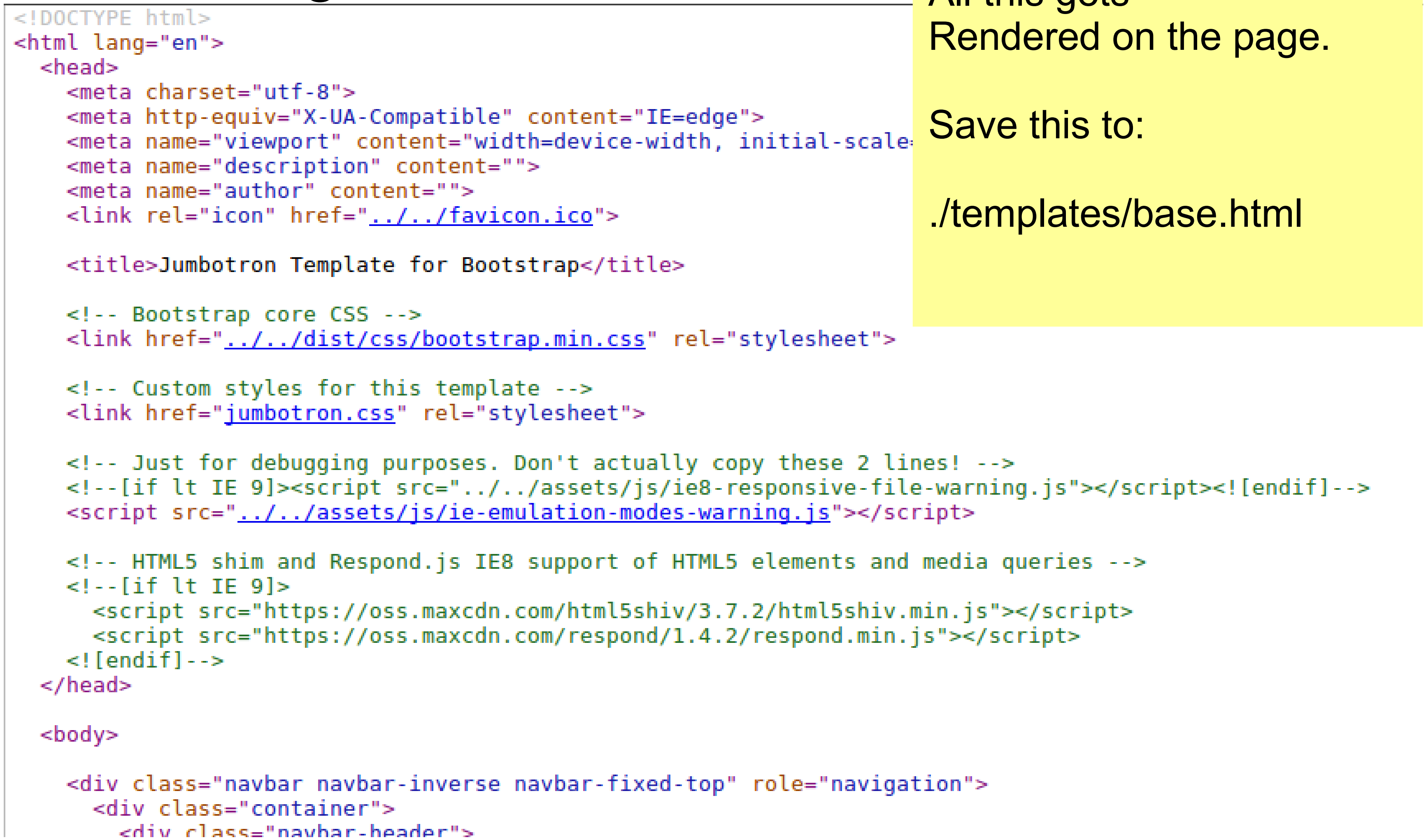
Using the example directly.

- Ver codigo fuente:

All this gets
Rendered on the page.

Save this to:

./templates/base.html



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="../../favicon.ico">

    <title>Jumbotron Template for Bootstrap</title>

    <!-- Bootstrap core CSS -->
    <link href="../../dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="jumbotron.css" rel="stylesheet">

    <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
    <!--[if lt IE 9]><script src="../../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->
    <script src="../../assets/js/ie-emulation-modes-warning.js"></script>

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="navbar navbar-inverse navbar-fixed-top" role="navigation">
      <div class="container">
        <div class="navbar-header">
```

Changing “signup.html”

We want to change our previous web page to now use this base.html page.

Previous: signup.html

```
<!DOCTYPE html>
   <html>
  <head>
    </head>
    <body>
      <h1>Join Now<h1>
       <form method=‘POST’ action=‘’>  {%
csrf_token %}
             {{  form.as_p  }}
       </form>
    </body>
   </html>
```

updated: signup.html

```
{% extends ‘base.html’ %}


{% block content  %}
<h1>Join Now<h1>
       <form method=‘POST’ action=‘’>  {%
csrf_token %}
             {{  form.as_p  }}
       </form>
{% endblock  %}
```

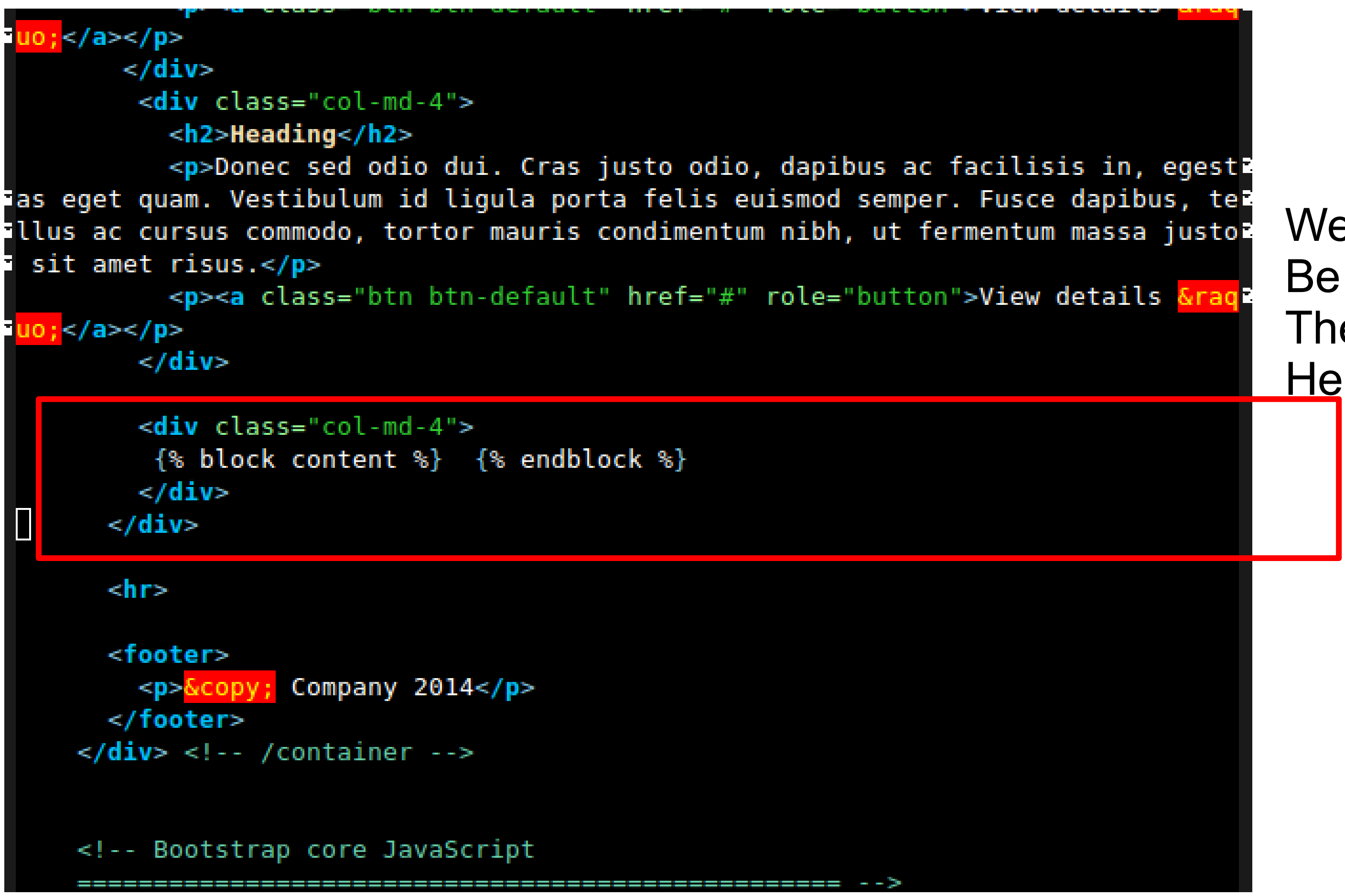
Modify base.html

- We need to put in the signup information.

```
....

<div class="col-md-4">
   {% block content %}
   {% endblock  %}
  </div>
</div>
<hr>

<footer>
....
```

Just insert this anywhere (sensible)
Within the bootstrap html that
you downloaded

Here is the modification in base.html



```
uo;</a></p>
       </div>
        <div class="col-md-4">
          <h2>Heading</h2>
          <p>Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egest
as eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, te
llus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo
 sit amet risus.</p>
          <p><a class="btn btn-default" href="#" role="button">View details &raq
uo;</a></p>
        </div>

        <div class="col-md-4">
         {% block content %}  {% endblock %}
        </div>
      </div>

      <hr>

      <footer>
        <p>&copy; Company 2014</p>
      </footer>
    </div> <!-- /container -->


    <!-- Bootstrap core JavaScript
    ================================================== -->
```

We add the code to
Be able to add
The block content
Here.

Put the CSS styles in place

- One way to do it is to see the source and just

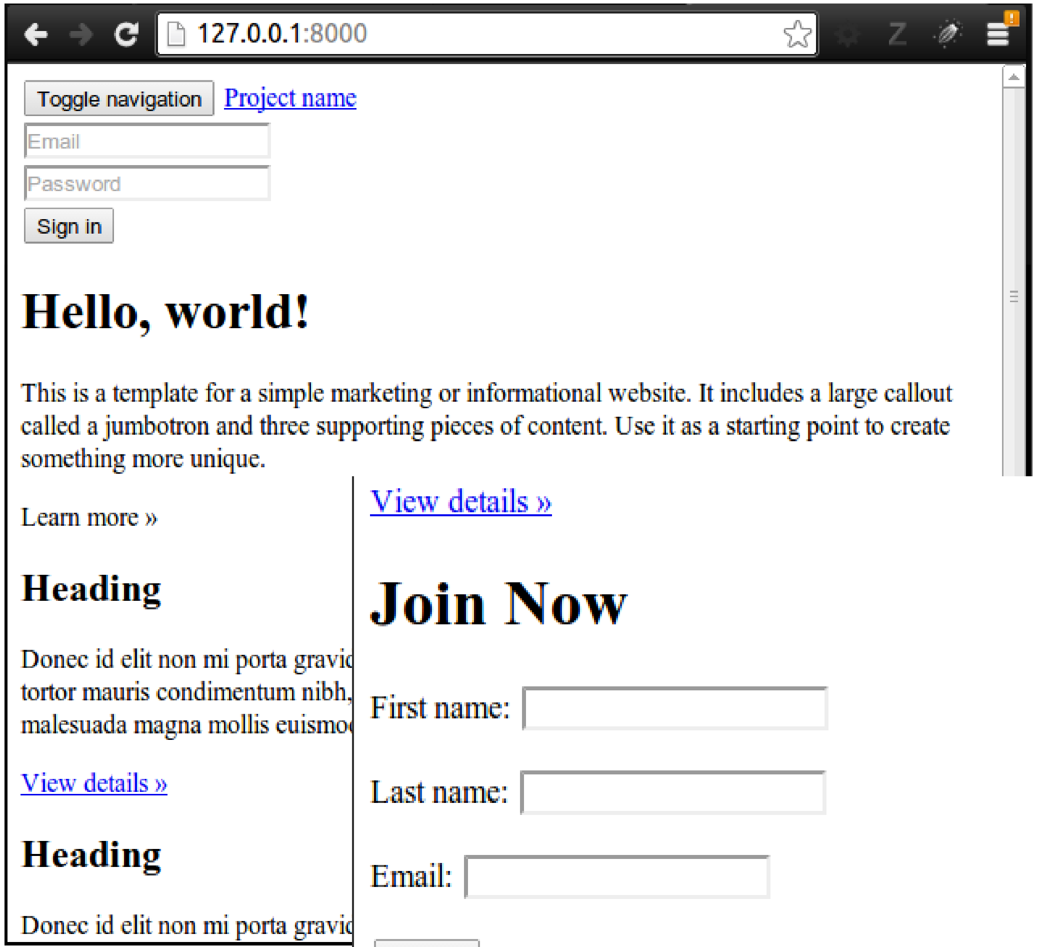
- Save bootstrap.css into the local directories
- For css.
- Save js files in same place ….(show this)

Change the paths in the base.html so that it can find the css path.

```
$  python manage.py collectstatic
```

Make sure that you do this.

Test out the web page



127.0.0.1:8000

Toggle navigation Project name

Email

Password

Sign in

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Learn more »

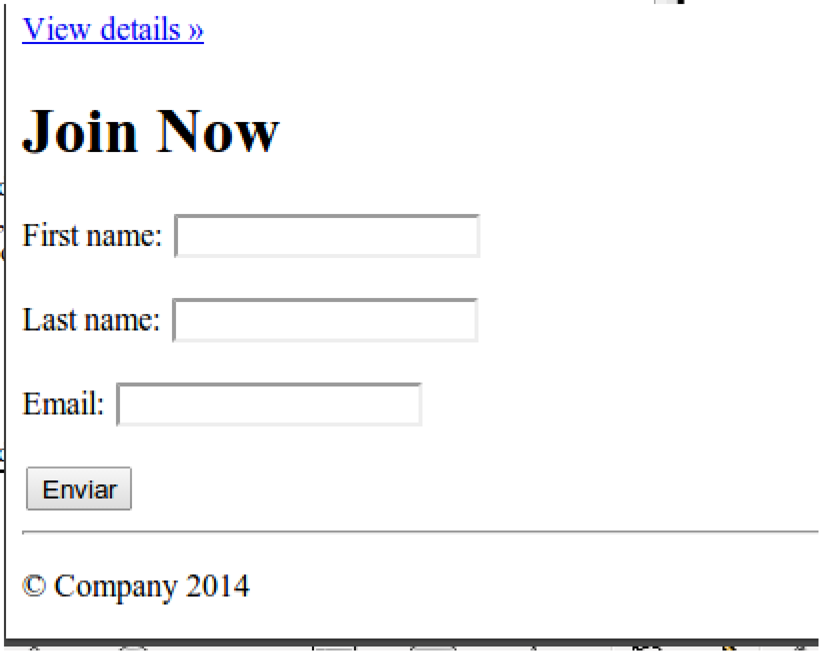
Heading

Donec id elit non mi porta gravic
tortor mauris condimentum nibh,
malesuada magna mollis euismod

View details »

Heading

Donec id elit non mi porta gravic



View details »

Join Now

First name:

Last name:

Email:

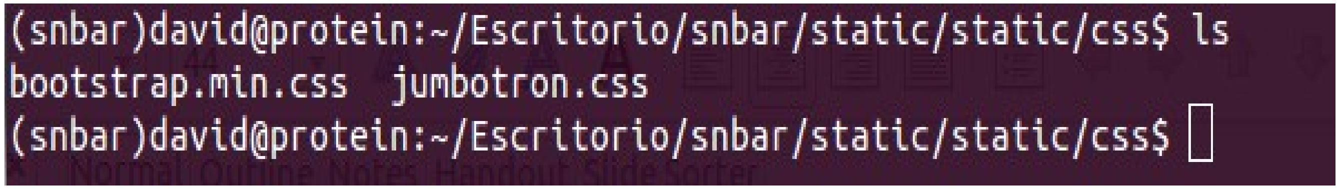
Enviar

© Company 2014

It works, but we have no CSS that we expected.

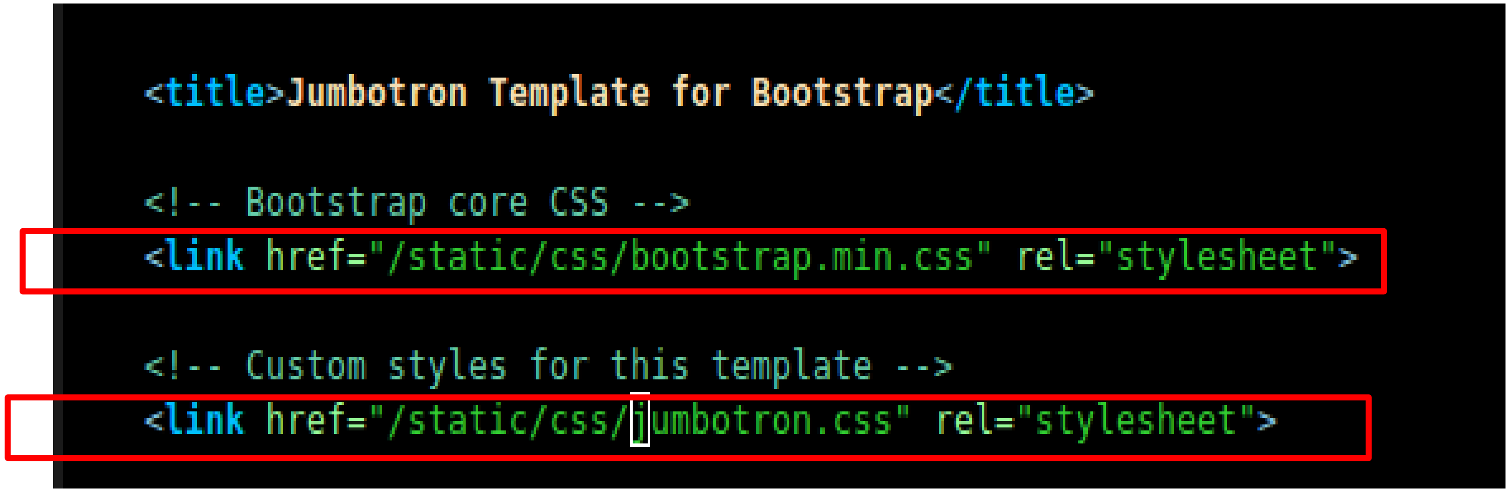
This can be fixed by Downloading the CSS And placing it in the Appropriate place.

Save the css to our static files

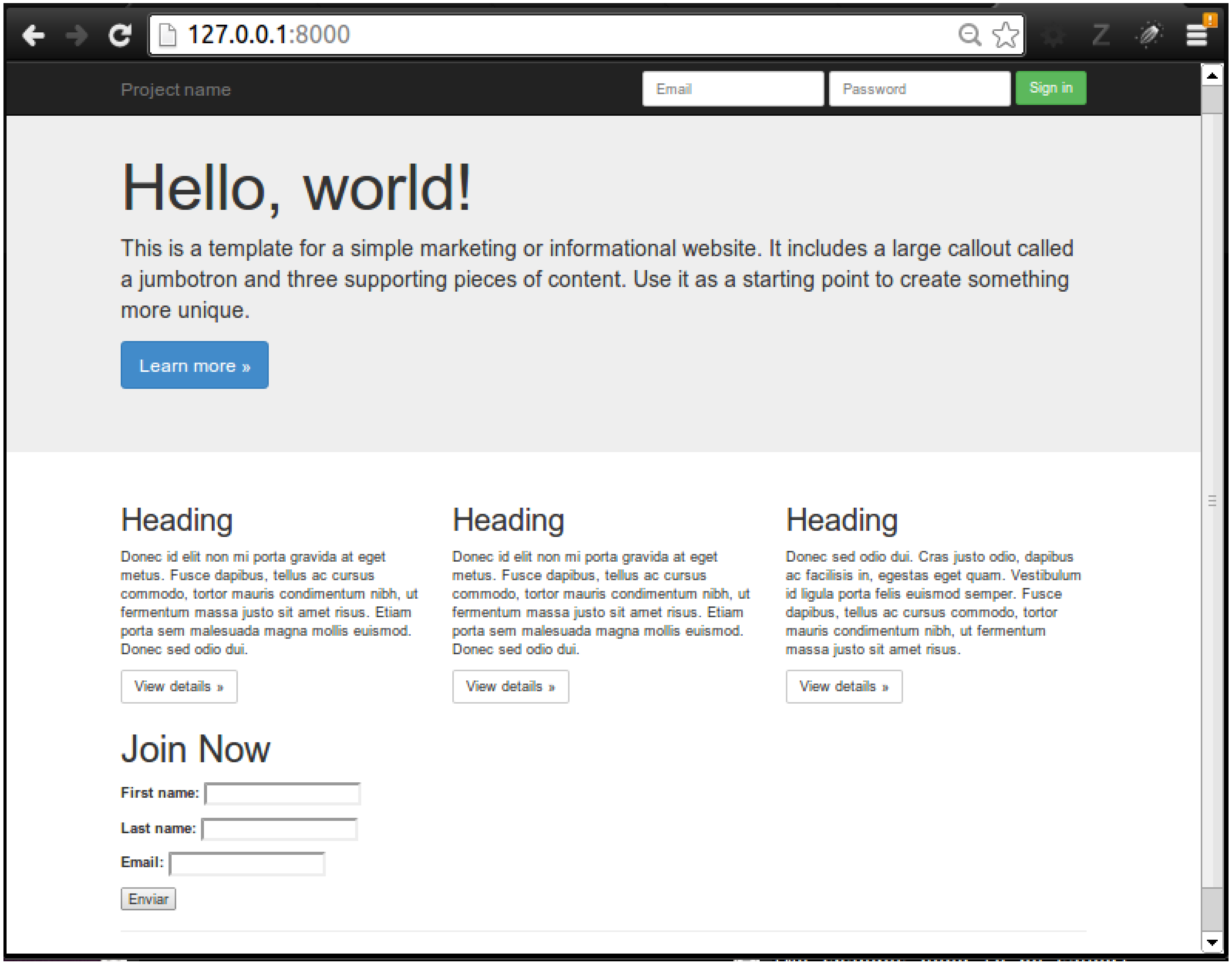


```
(snbar)david@protein:~/Escritorio/snbar/static/static/css$ ls
bootstrap.min.css  jumbotron.css
(snbar)david@protein:~/Escritorio/snbar/static/static/css$ 
```

Modify the “base.html” file to point to the correct css



```
<title>Jumbotron Template for Bootstrap</title>

<!-- Bootstrap core CSS -->
<link href="/static/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom styles for this template -->
<link href="/static/css/jumbotron.css" rel="stylesheet">
```



Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

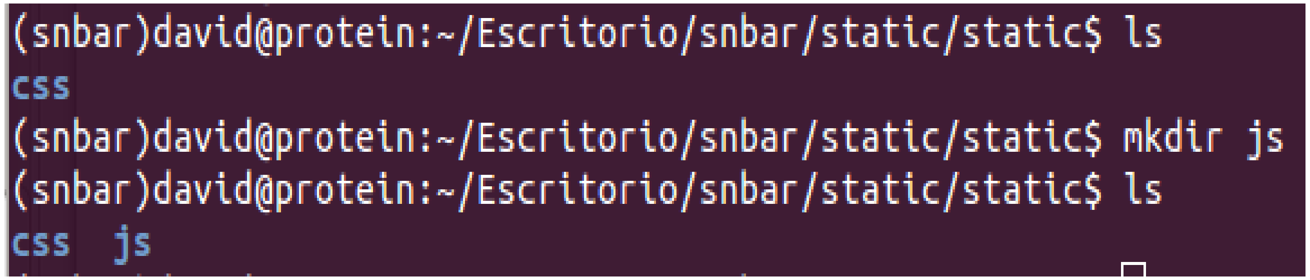
Join Now

First name:

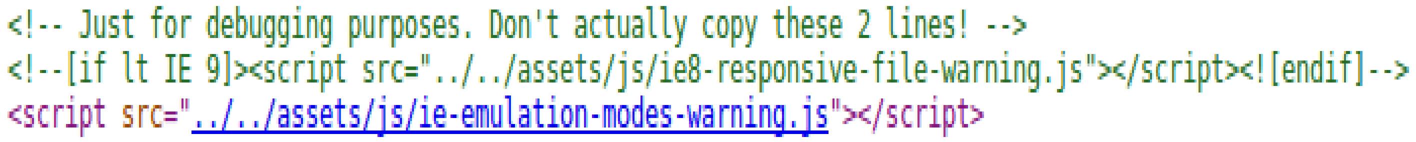
Last name:

Email:

Now save javascript

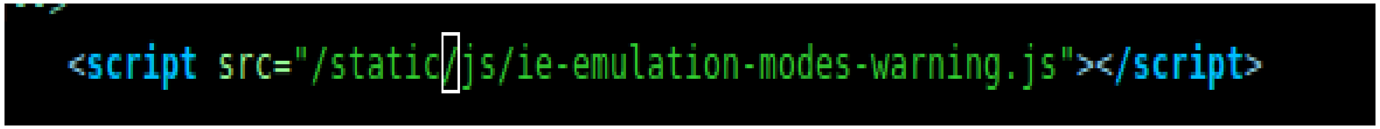


```
(snbar)david@protein:~/Escritorio/snbar/static/static$ ls
css
(snbar)david@protein:~/Escritorio/snbar/static/static$ mkdir js
(snbar)david@protein:~/Escritorio/snbar/static/static$ ls
css  js
```



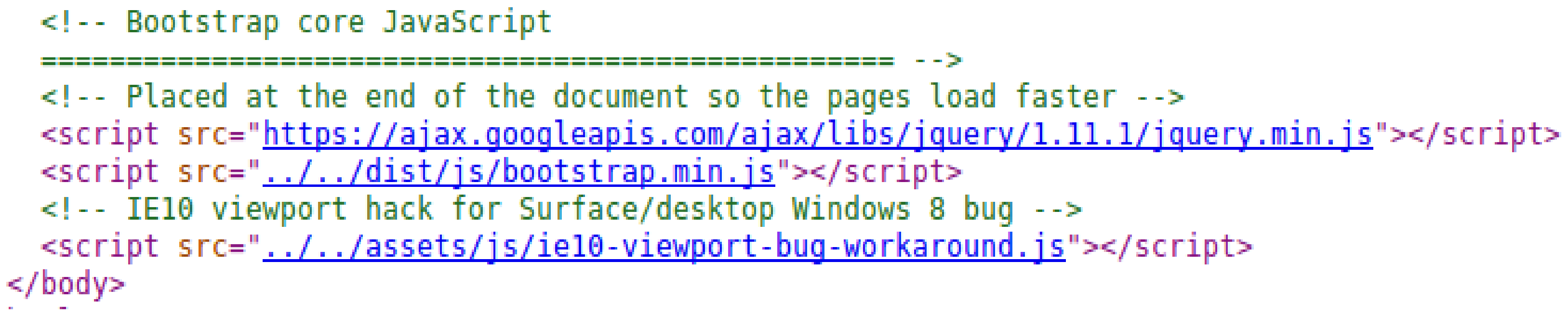
```
<!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
<!--[if lt IE 9]><script src="../../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->
<script src="../../assets/js/ie-emulation-modes-warning.js"></script>
```

Save this file to /static/js   and then change “base.html”



```
  <script src="/static/js/ie-emulation-modes-warning.js"></script>
```

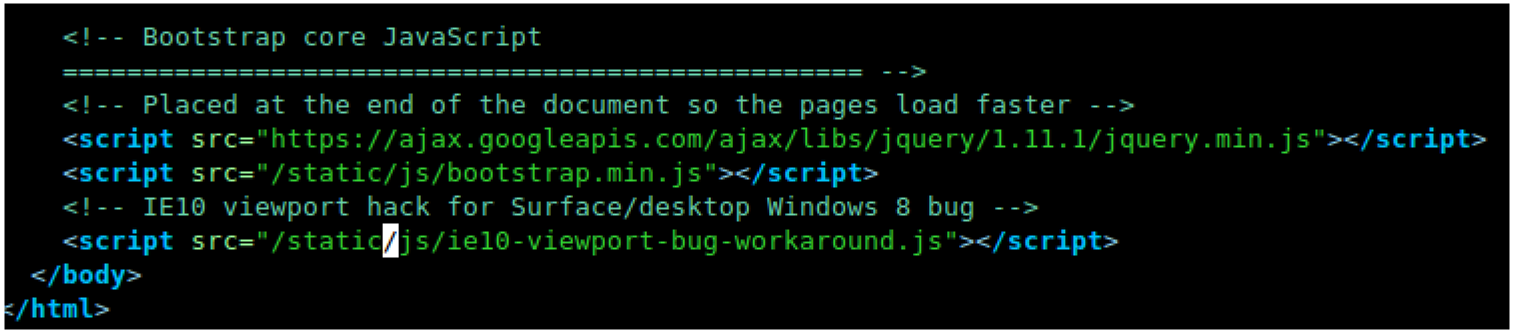
More javascript to save



```
  <!-- Bootstrap core JavaScript
  ================================================== -->
  <!-- Placed at the end of the document so the pages load faster -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
  <script src="../../dist/js/bootstrap.min.js"></script>
  <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
  <script src="../../assets/js/ie10-viewport-bug-workaround.js"></script>
</body>
```

http://getbootstrap.com/dist/js/bootstrap.min.js

Save this also to the js directory



```
    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <script src="/static/js/bootstrap.min.js"></script>
    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <script src="/static/js/ie10-viewport-bug-workaround.js"></script>
  </body>
</html>
```

How Bootstrap works

- Explore with chrome’s “Inspect Element”
- We can customize the styles and CSS

Create a new file “custom.css”

```
.jumbotron {
    background-color:
#17D1FF
}
```

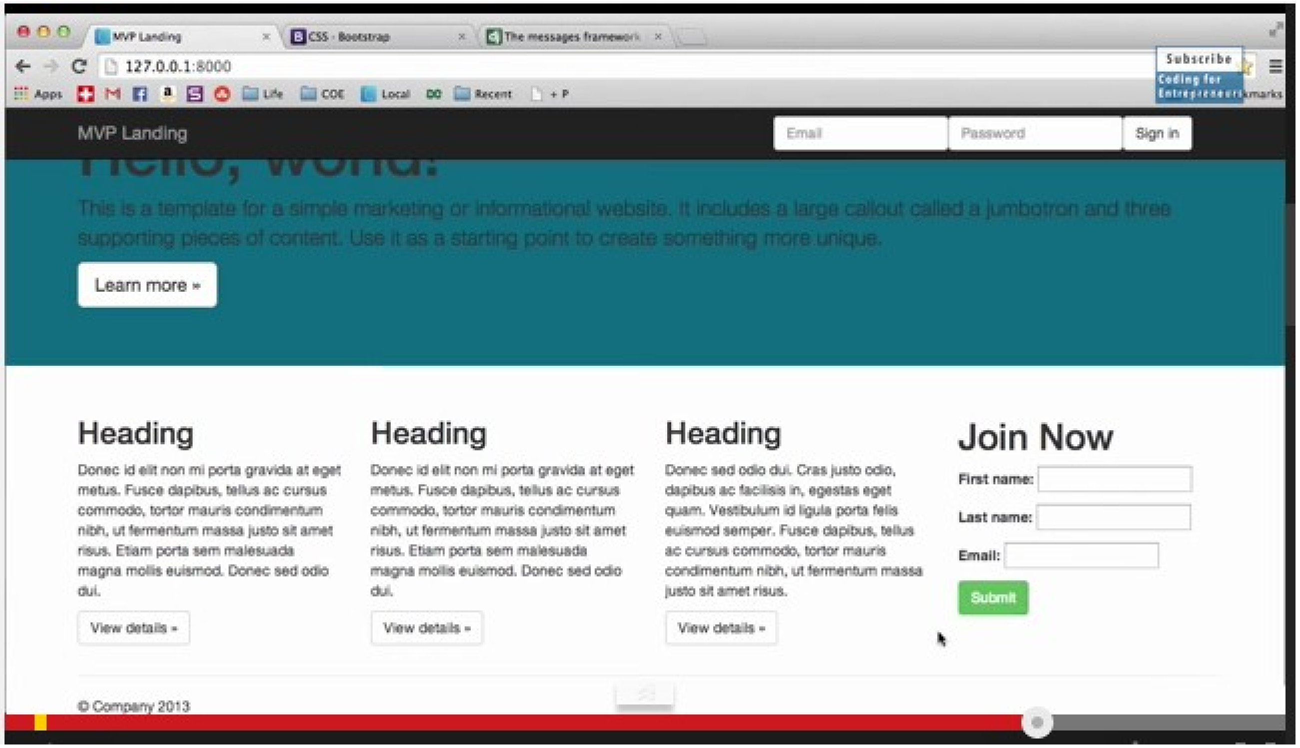
Bootstrap details

Using the Bootstrap to customize

- Info at: http://getbootstrap.com/css/

Result of Message

- Try Django Tutorial 10 of 21 - Learn to Customize Twitter Bootstrap 3



MVP Landing

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

Join Now

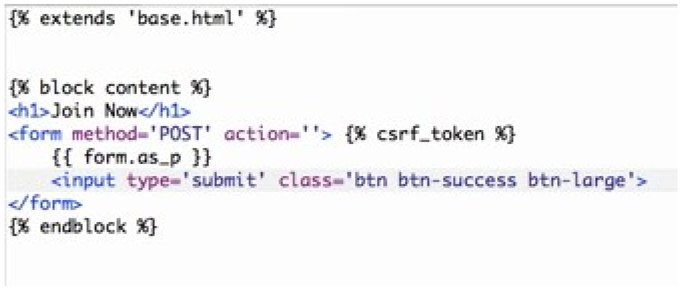
First name:

Last name:

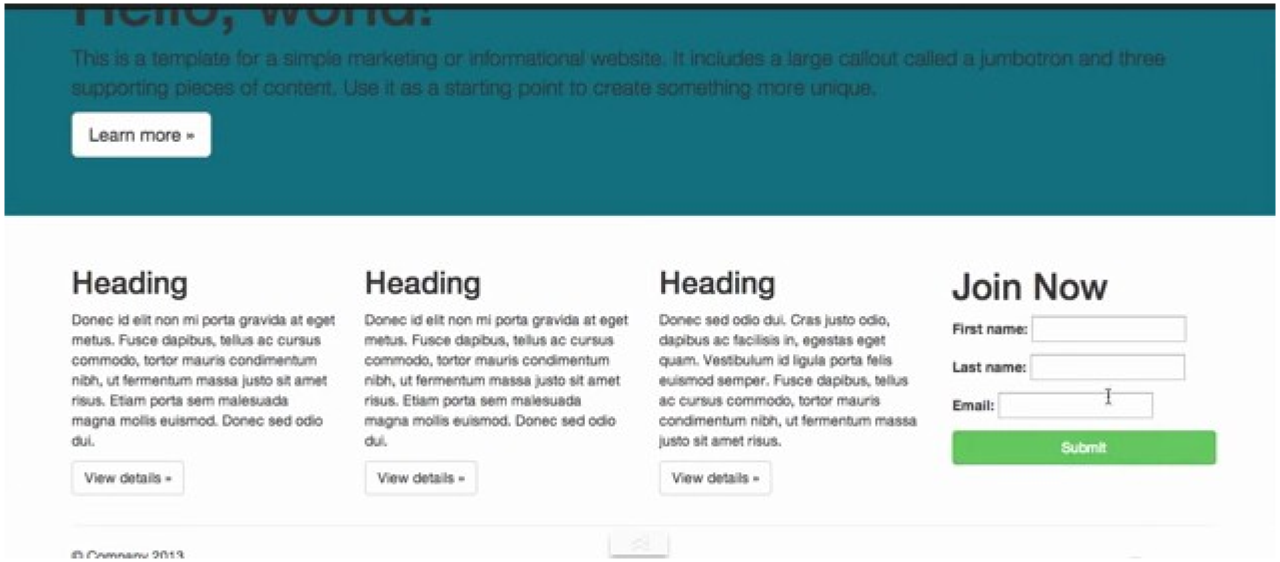
Email:

© Company 2013

Using bootstrp to add a success button



```
{% extends 'base.html' %}


{% block content %}
<h1>Join Now</h1>
<form method='POST' action=''> {% csrf_token %}
    {{ form.as_p }}
    <input type='submit' class='btn btn-success btn-large'>
</form>
{% endblock %}
```



This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Learn more »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

Join Now

First name:

Last name:

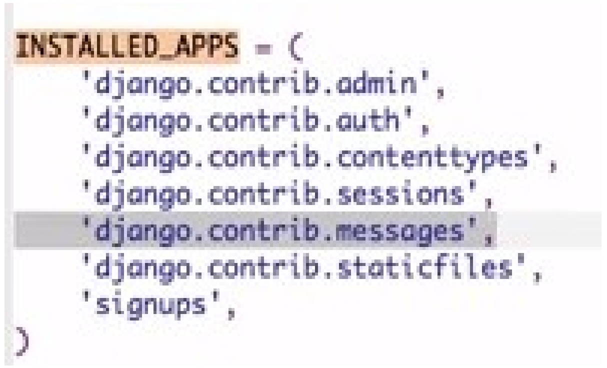
Email:

Submit

© Company 2013

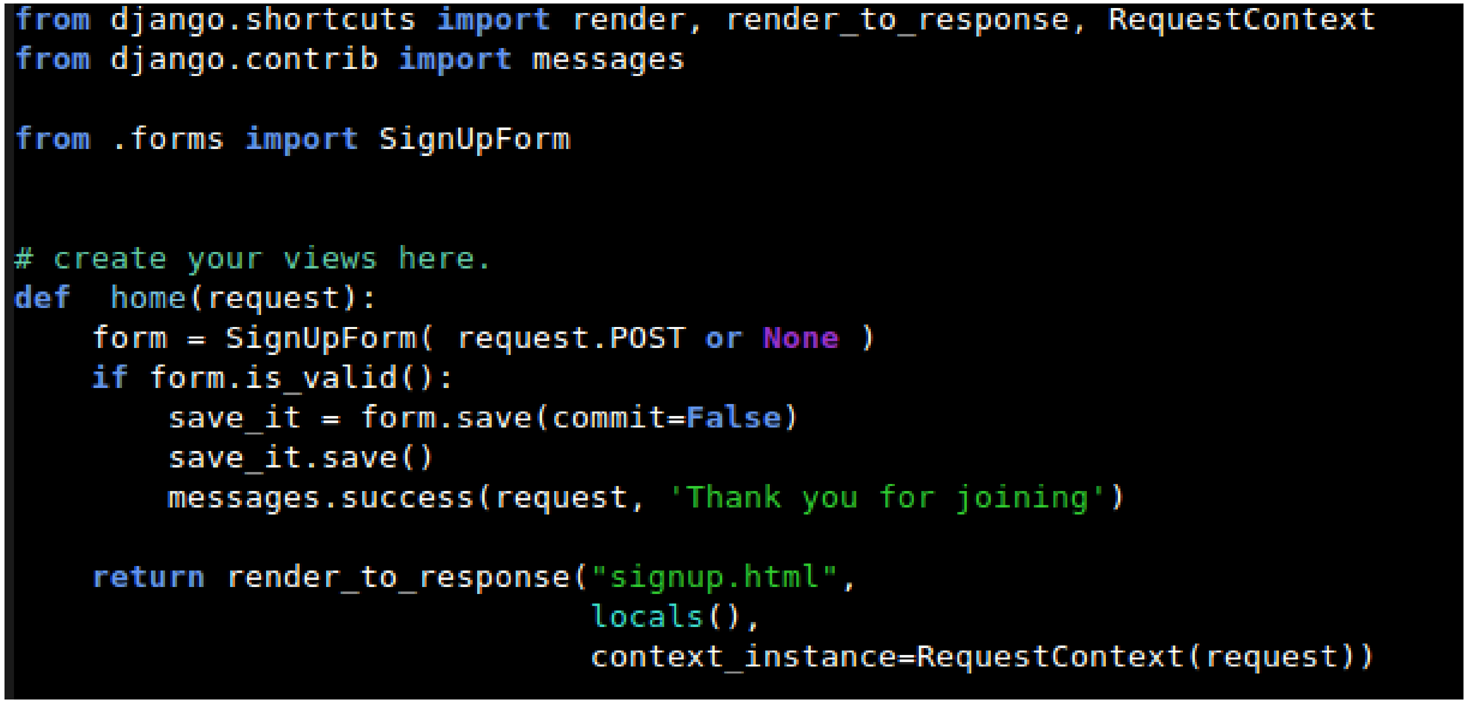
Messaging and Redirect

- After someone joins, we want to give a page that says thanks
- We use message of Django.

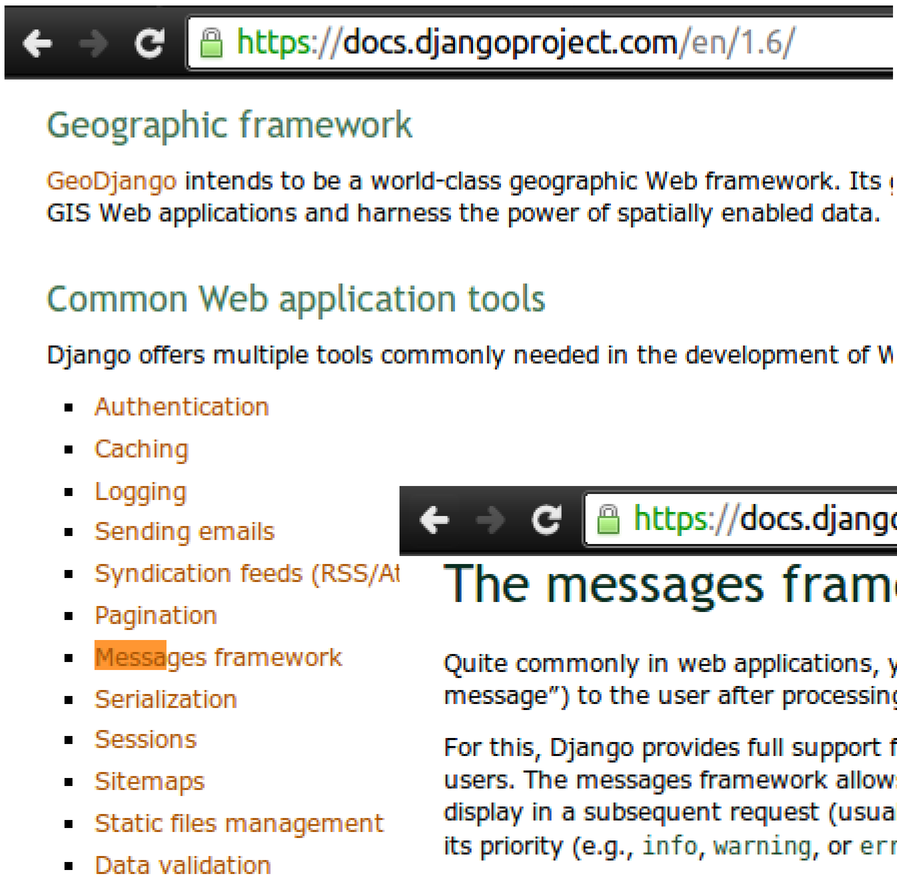


```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'signups',
)
```

In views, we want to import this into
The file “views



```
from django.shortcuts import render, render_to_response, RequestContext
from django.contrib import messages

from .forms import SignUpForm


# create your views here.
def  home(request):
    form = SignUpForm( request.POST or None )
    if form.is_valid():
        save_it = form.save(commit=False)
        save_it.save()
        messages.success(request, 'Thank you for joining')

    return render_to_response("signup.html",
                              locals(),
                              context_instance=RequestContext(request))
```


https://docs.djangoproject.com/en/1.6/

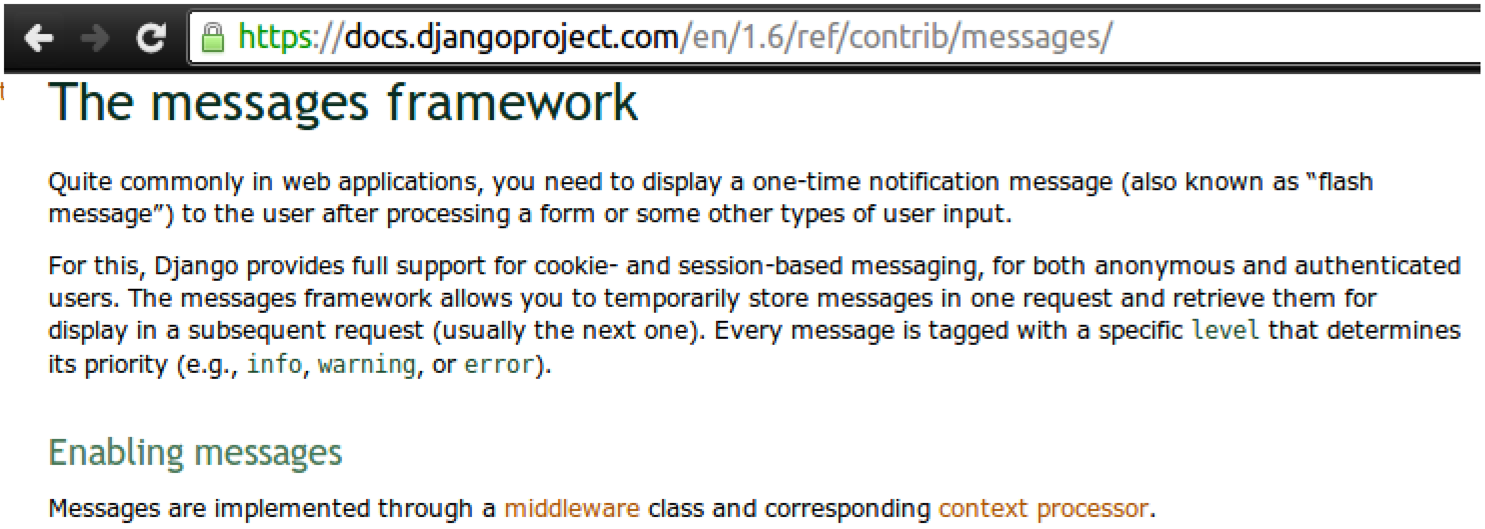
Geographic framework

GeoDjango intends to be a world-class geographic Web framework. Its
GIS Web applications and harness the power of spatially enabled data.

Common Web application tools

Django offers multiple tools commonly needed in the development of W

- Authentication
- Caching
- Logging
- Sending emails
- Syndication feeds (RSS/At
- Pagination
- Messages framework
- Serialization
- Sessions
- Sitemaps
- Static files management
- Data validation


https://docs.djangoproject.com/en/1.6/ref/contrib/messages/

The messages framework

Quite commonly in web applications, you need to display a one-time notification message (also known as "flash message") to the user after processing a form or some other types of user input.

For this, Django provides full support for cookie- and session-based messaging, for both anonymous and authenticated users. The messages framework allows you to temporarily store messages in one request and retrieve them for display in a subsequent request (usually the next one). Every message is tagged with a specific `level` that determines its priority (e.g., `info`, `warning`, or `error`).

Enabling messages

Messages are implemented through a middleware class and corresponding context processor.

Adding a message

To add a message, call:

```
from django.contrib import messages
messages.add_message(request, messages.INFO, 'Hello world.')
```

Some shortcut methods provide a standard way to add messages with commonly used tags (which are usually represented as HTML classes for the message):

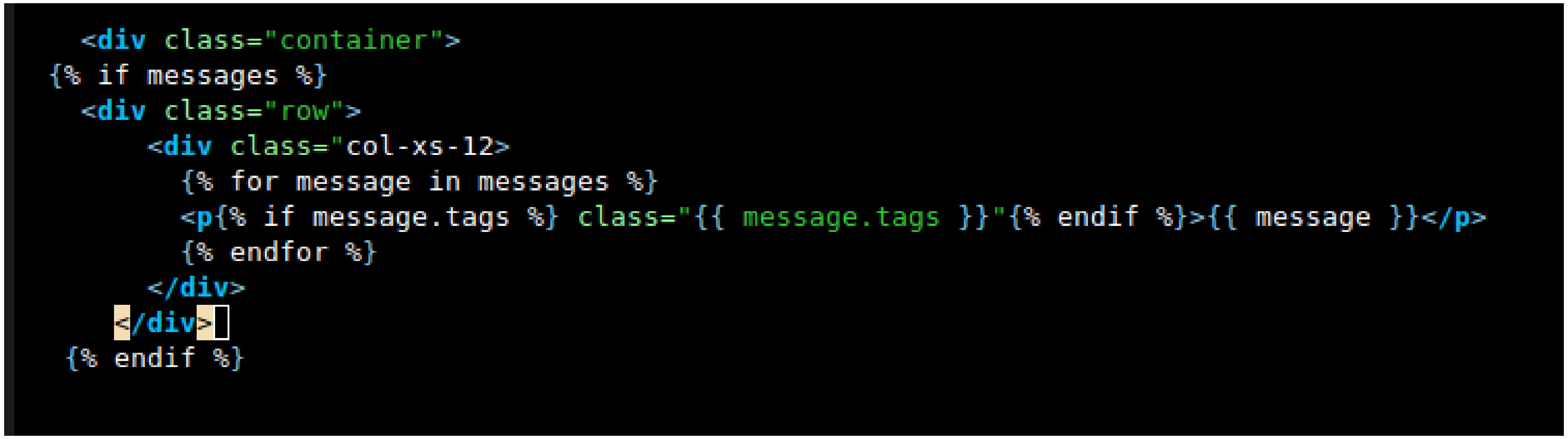
```
messages.debug(request, '%s SQL statements were executed.' % count)
messages.info(request, 'Three credits remain in your account.')
messages.success(request, 'Profile details updated.')
messages.warning(request, 'Your account expires in three days.')
messages.error(request, 'Document deleted.')
```

Displaying messages

In your template, use something like:

```
{% if messages %}
<ul class="messages">
    {% for message in messages %}
    <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>
    {% endfor %}
</ul>
{% endif %}
```

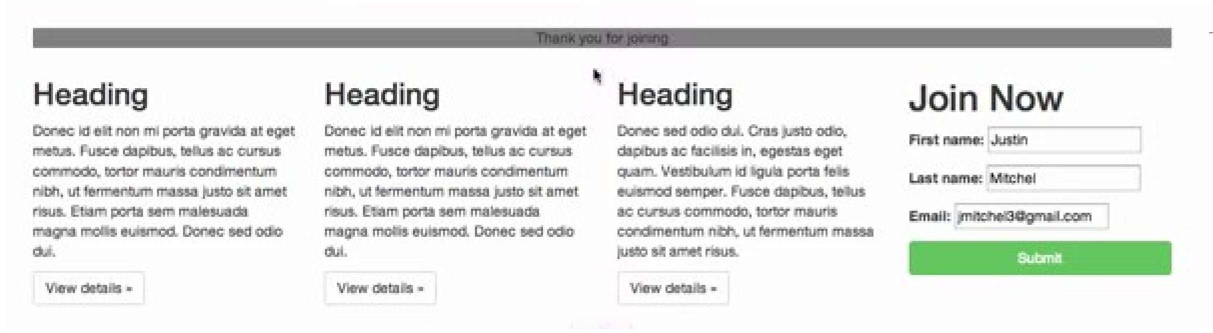
So we need to render the message



```
   <div class="container">
{% if messages %}
  <div class="row">
      <div class="col-xs-12>
        {% for message in messages %}
        <p{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</p>
        {% endfor %}
      </div>
    </div>
 {% endif %}
```

result in page

Here is the Result...



Thank you for joining

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

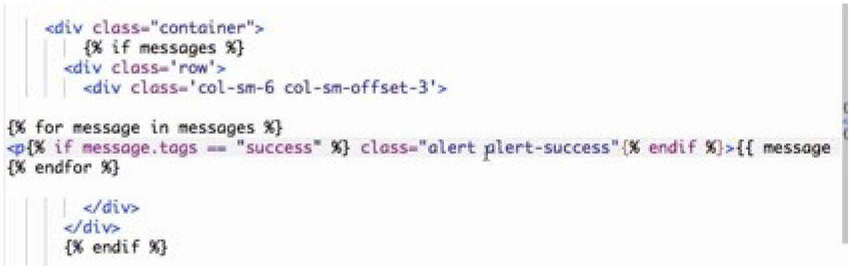
Join Now

First name: Justin

Last name: Mitchel

Email: jmitchel3@gmail.com

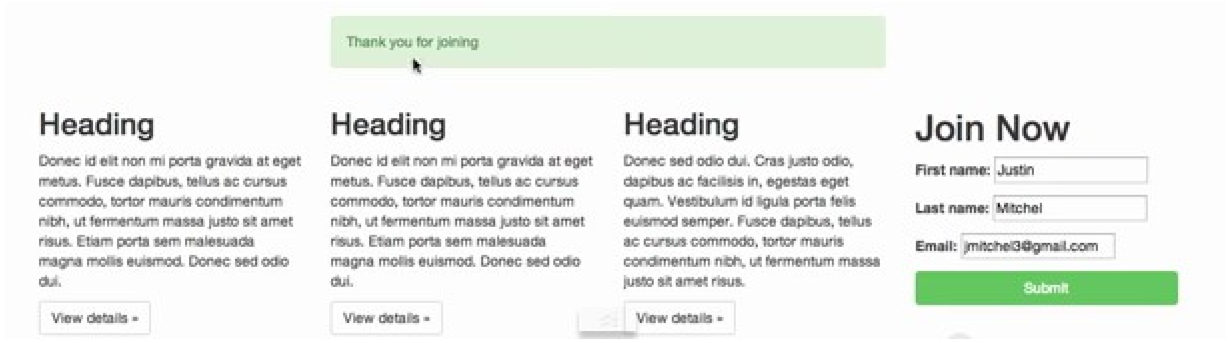
Submit



```
  <div class="container">
      {% if messages %}
    <div class='row'>
      <div class='col-sm-6 col-sm-offset-3'>

{% for message in messages %}
<p{% if message.tags == "success" %} class="alert alert-success"{% endif %}>{{ message
{% endfor %}

      </div>
    </div>
    {% endif %}
```

Modify the “base.html” to use bootstrap's “alert-success” class

See Django Tutorial 11 (5:26)



Thank you for joining

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

Join Now

First name: Justin

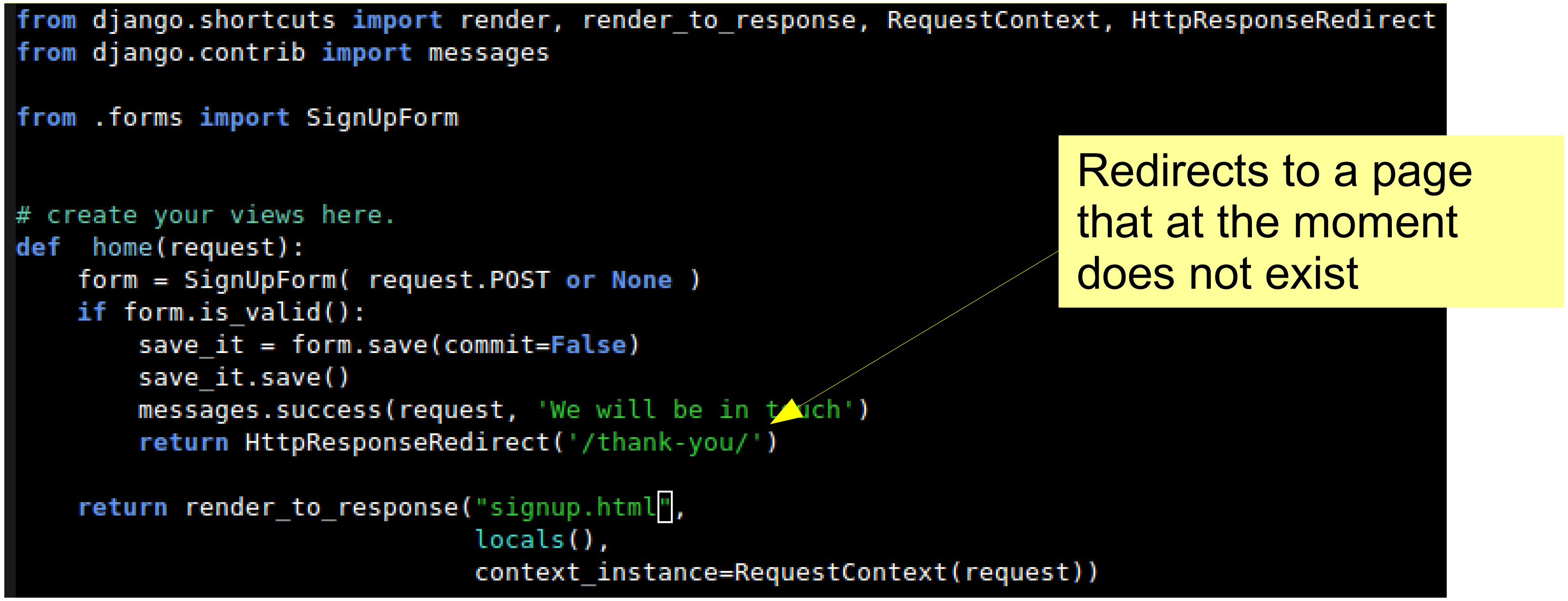
Last name: Mitchel

Email: jmitchel3@gmail.com

Submit

Redirecting

- Imagine, once we press the button, we want to send the user to a different page.
- For this, in the “views.py”, we add a class called:
  - HttpResponseRedirect

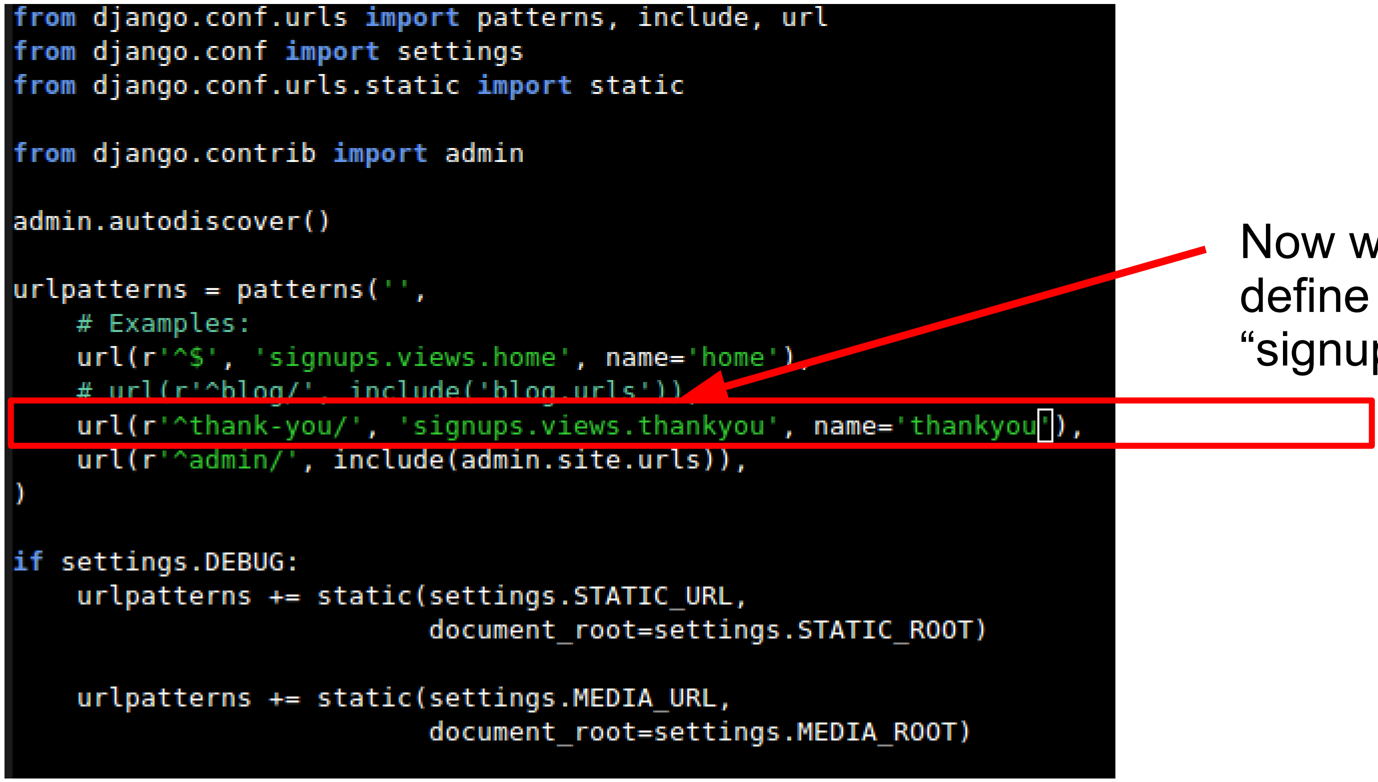


```python
from django.shortcuts import render, render_to_response, RequestContext, HttpResponseRedirect
from django.contrib import messages

from .forms import SignUpForm


# create your views here.
def  home(request):
    form = SignUpForm( request.POST or None )
    if form.is_valid():
        save_it = form.save(commit=False)
        save_it.save()
        messages.success(request, 'We will be in touch')
        return HttpResponseRedirect('/thank-you/')

    return render_to_response("signup.html",
                              locals(),
                              context_instance=RequestContext(request))
```

Redirects to a page that at the moment does not exist

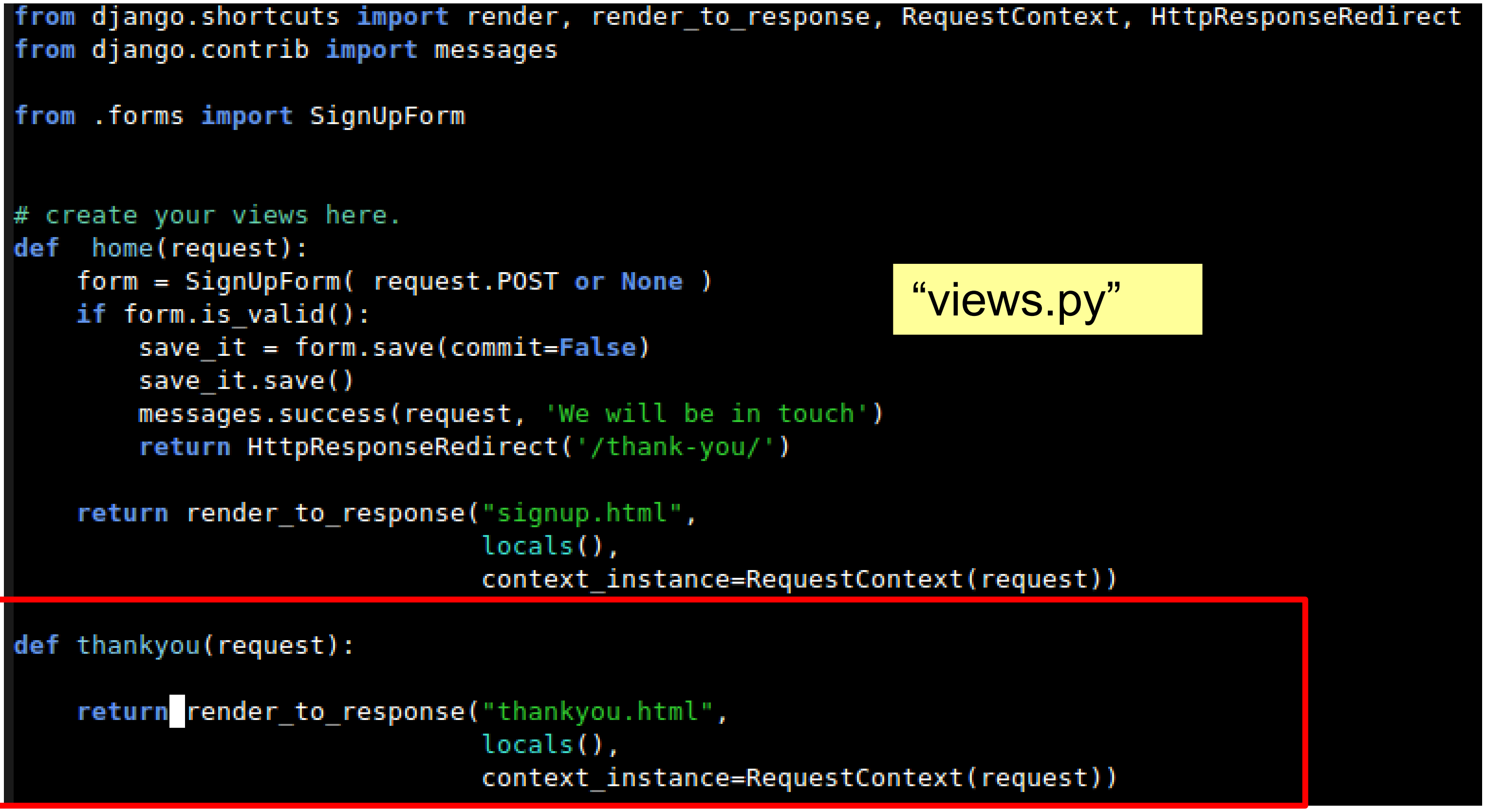
Adding the redirect page

- Modify “urls.py”



```
from django.conf.urls import patterns, include, url
from django.conf import settings
from django.conf.urls.static import static

from django.contrib import admin

admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    url(r'^$', 'signups.views.home', name='home'),
    # url(r'^blog/', include('blog.urls')),
    url(r'^thank-you/', 'signups.views.thankyou', name='thankyou'),
    url(r'^admin/', include(admin.site.urls)),
)

if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL,
                          document_root=settings.STATIC_ROOT)

    urlpatterns += static(settings.MEDIA_URL,
                          document_root=settings.MEDIA_ROOT)
```

Now we need to define this in the “signups.views”



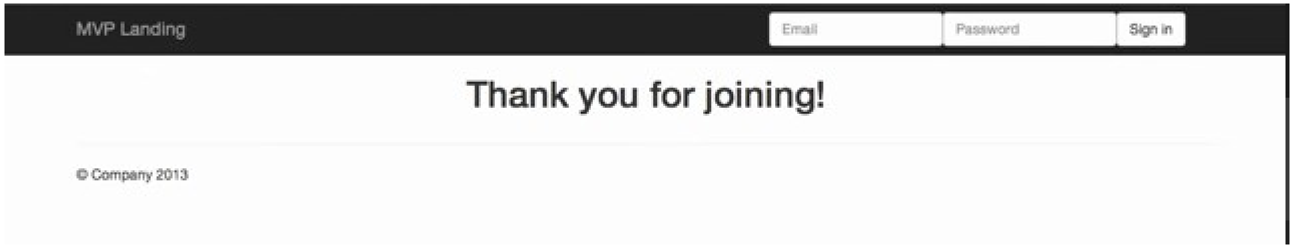
```
from django.shortcuts import render, render_to_response, RequestContext, HttpResponseRedirect
from django.contrib import messages

from .forms import SignUpForm


# create your views here.
def  home(request):
    form = SignUpForm( request.POST or None )
    if form.is_valid():
        save_it = form.save(commit=False)
        save_it.save()
        messages.success(request, 'We will be in touch')
        return HttpResponseRedirect('/thank-you/')

    return render_to_response("signup.html",
                              locals(),
                              context_instance=RequestContext(request))

def thankyou(request):

    return render_to_response("thankyou.html",
                              locals(),
                              context_instance=RequestContext(request))
```

"views.py"

We need to create an empty
Html document in the directory:
/static/templates/

Now the power of templates

- For this, see the tutorial:
  - Django tutorial 12: (2:40)
  - https://www.youtube.com/watch?v=HuBeNrCa3cc&list=PLEsfXFp6DpzRgedo9IzmcpXYoSeDg29Tx&index=12

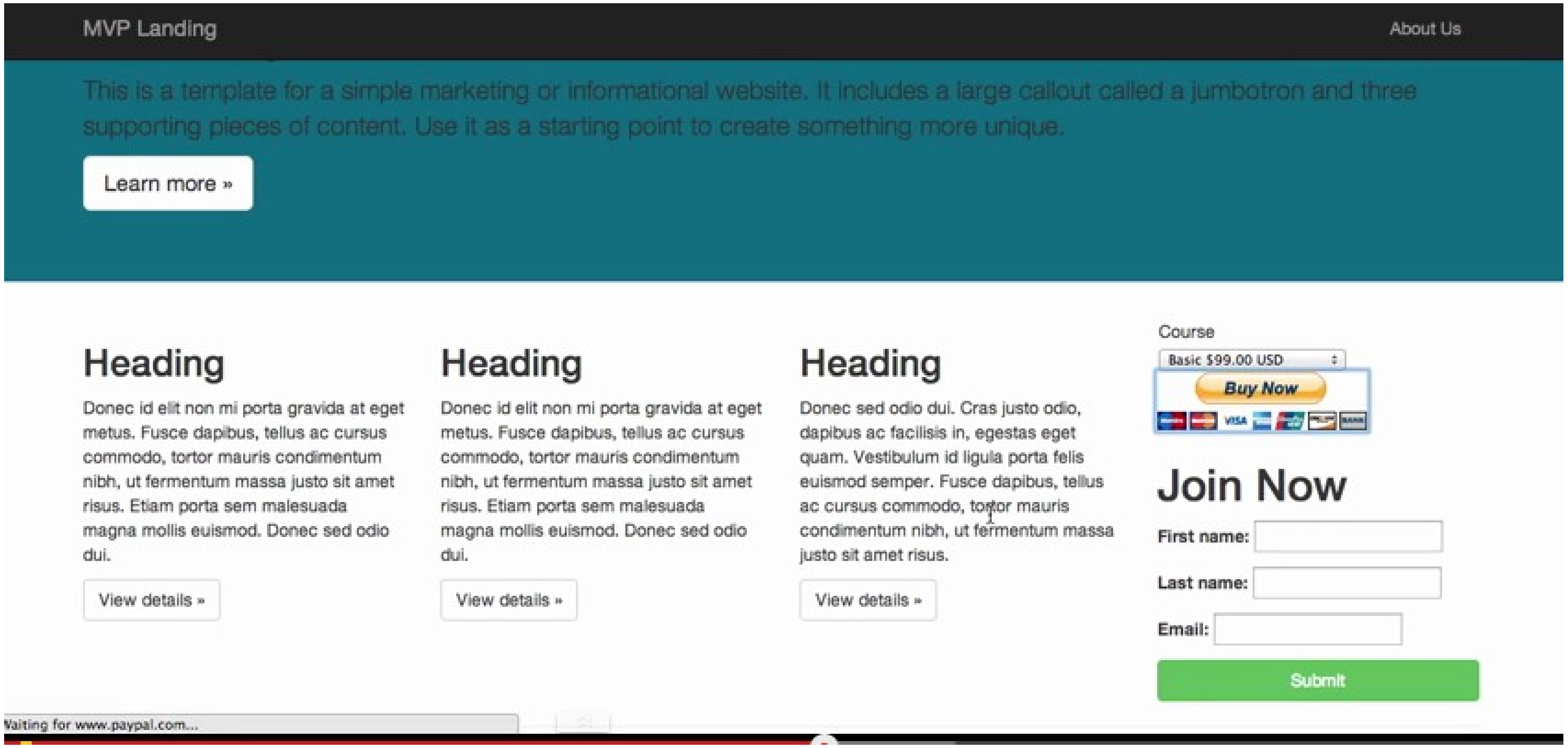


Thank you for joining!

© Company 2013

Adding paypal button

- Try Django Tutorial 14 of 21 - Add Basic Paypal Button to Django Project Template
  - https://www.youtube.com/watch?v=qKS87S0Imsk&index=14&list=PLEsfXFp6DpzRgedo9IzmcpXYoSeDg29Tx



MVP Landing

About Us

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Learn more »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

Course

Basic $99.00 USD

Buy Now

Join Now

First name:

Last name:

Email:

Submit

Waiting for www.paypal.com...

Summary

- We have now connected Django to Bootstrap; a powerful HTML, CSS, JS Framework
- The data is connected and we can view it in the admin
- We can redirect to more pages
- We can start to design more powerful apps.
- Ecommerce with paypal