**Topic: Registration Module: Part 14** 

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django: Automating Common Tasks



1. We need a URL pattern like http://127.0.0.1:8000/register/, so in the main project's URLS.PY.

```
×
     EXPLORER
                                                      AUTOMATINGCOMMONTASKS
Ф

    autocommontasks_main

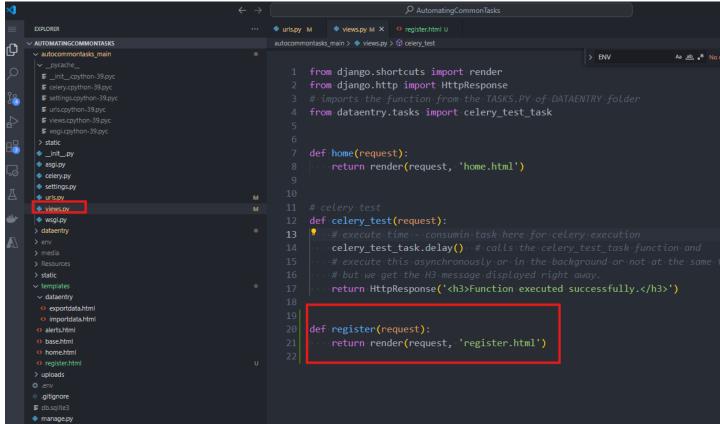
                                                                 1. Add an import: from my_app import views > ENV
                                                                                                                                       Aa ab * No res
                                                                 2. Add a URL to urlpatterns: path('', views.home, name='home')
                                                                 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
                                                             Including another URLconf
       ■ wsgi.cpython-39.py
                                                                 1. Import the include() function: from django.urls import include, path
      > static
H
      asgi.py
      celery.py
                                                             from django.contrib import admin
      settings.py
                                                             from django.urls import path, include
                                                             from django.conf.urls.static import static
     > dataentry
                                                             from django.conf import settings
                                                             urlpatterns = [

√ dataentry

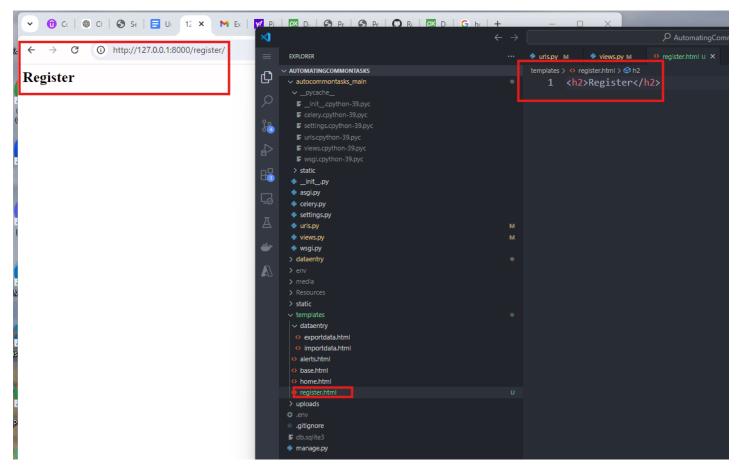
    exportdata.html

      o importdata.html
                                                                 path('celery-test/', views.celery_test),
      alerts.html
      base.html
      o home.html
                                                             ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
     manage.pv
```

2. Create the REGISTER function in VIEWS.PY:



3. Create the REGISTER.HTML in TEMPLATES.



4. We update our REGISTER.HTML to include our Django tags.

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

{% block content %}

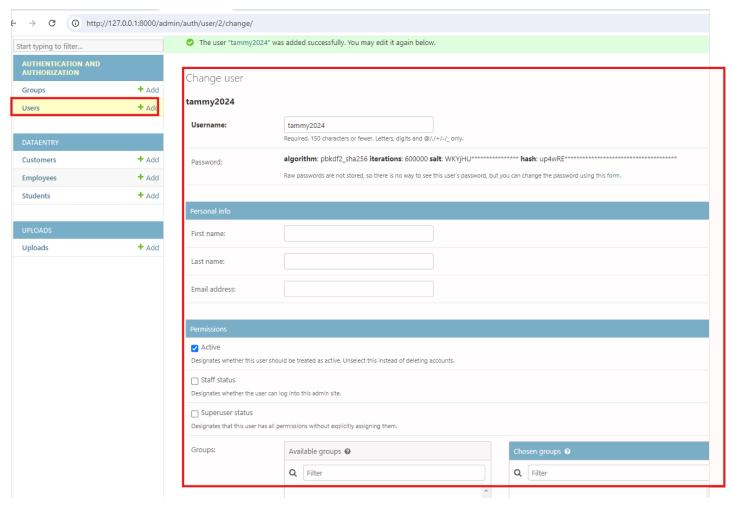
<we add our unique code here>
```

{% endblock %}

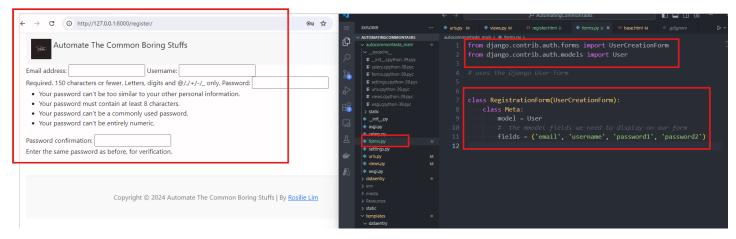
5. We update our BASE.HTML to link our webpages to REGISTER.HTML

```
| Variable | Variable
```

6. To create the registration form, we can manually create the UI form or automate this using DJANGO MODEL FORM (similar to our Admin Panel User Registration) WITH CRISPY TO FORMAT IT.



7. Create a new file, FORMS.PY, in your main project directory.



8. In our VIEWS.PY, we write:

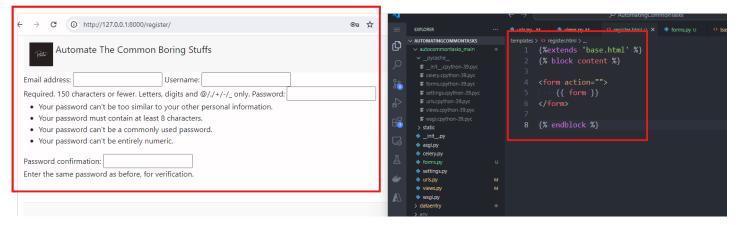
```
settings.py
                                   def register(request):
urls.py
                                        if request.method == 'POST':
views.py
wsgi.py
> dataentry
                                        else:
                                             form = RegistrationForm()
                                             context = {
                                                  'form': form,
> static

    templates

 dataentry
                                        return render(request, 'register.html', context)

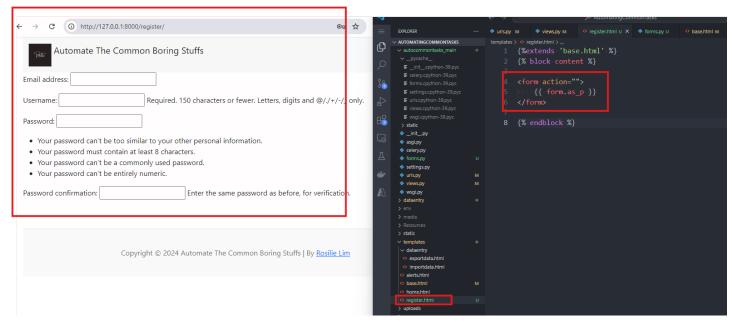
    exportdata.html
```

9. In our REGISTER.HTML, we write



10. If we use the conventional way to set up the form, it shall be like this:

```
{% extends 'base.html' %}
{% block content %}
<form action="">
    {{ form.as_p }}
</form>
{% endblock %}
```



11. To format this form, we can use the CRISPY FORM. See this for the CRISPY FORM DOCUMENTATION.

## Installing django-crispy-forms

Install latest stable version into your python environment using pip:

```
pip install django-crispy-forms
```

If you want to install development version (unstable), you can do so doing:

```
pip install git+git://github.com/django-crispy-forms/django-crispy-forms.git@main#egg
```

Or, if you'd like to install the development version as a git repository (so you can git pull updates), use the -e flag with pip install, like so:

```
pip install -e git+git://github.com/django-crispy-forms/django-crispy-forms.git@main#
```

Once installed add crispy\_forms to your INSTALLED\_APPS in settings.py:

```
INSTALLED_APPS = (
    ...
    'crispy_forms',
)
```

In production environments, always activate Django template cache loader. This is available since Django 1.2 and what it does is basically load templates once, then cache the result for every subsequent render. This leads to a significant performance improvement. To see how to set it up refer to the fabulous Django docs page.

## Template packs

uni-form Uni-form is a nice looking, well structured, highly customizable, accessible and usable forms.

In addition the following template packs are available through separately maintained projects.

- foundation Foundation In the creator's words, "The most advanced responsive front-end framework in the world." This template pack is available through crispy-forms-foundation
- tailwind Tailwind A utility first framework. This template pack is available through crispy-tailwind
- Support for newer versions of Bootstrap will be in separate template packs. This starts with version 5 and is available through crispy-bootstrap5
- Bulma Bulma: the modern CSS framework that just works. This template pack is available through crispy-bulma

If your form CSS framework is not supported and it's open source, you can create a <a href="mailto:crispy-forms-templatePackName">crispy-forms-templatePackName</a> project. Please let me know, so I can link to it. Documentation on How to create your own template packs is provided.

You can set your default template pack for your project using the CRISPY\_TEMPLATE\_PACK | Django settings variable:

CRISPY\_TEMPLATE\_PACK = 'uni\_form'

Please check the documentation of your template pack package for the correct value of the CRISPY\_TEMPLATE\_PACK setting (there are packages which provide more than one template pack).

Install Django Crispy Form: \$ pip install django-crispy-forms

In your SETTINGS.PY, register this in your INSTALLED\_APPS

```
INSTALLED_APPS = [

INSTAL
```



12. If you check your BASE.HTML the bootstrap we are using is version 5, that is why our CRISPY\_TEMPALTE\_PACK='bootstrap5' and so we need to install this as well

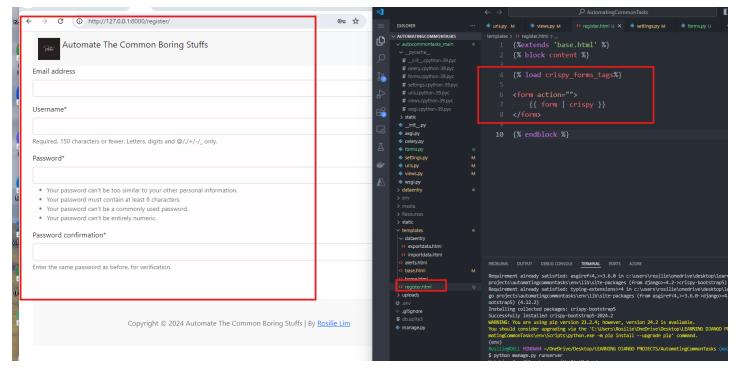
\$ pip install crispy-bootstrap5

Register this in your INSTALLED\_APPS:

```
# Application definition

INSTALLED_APPS = [

| 'django.contrib.admin',
| 'django.contrib.auth',
| 'django.contrib.contenttypes',
| 'django.contrib.sessions',
| 'django.contrib.messages',
| 'django.contrib.staticfiles',
| 'django.contrib.staticfiles',
| 'crispy_forms',
| 'crispy_forms',
| 'crispy_bootstrap5',
| 'dsagarana | 'crispy_bootstrap5',
| 'dsagarana | 'dsagar
```



14. Further updating our REGISTER.HTML.

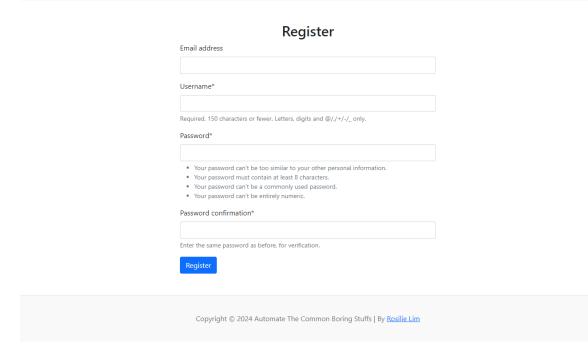
```
×
                                                                                                                        EXPLORER
                           ··· • urls.ov M • views.ov M ○ register.html U X • settings.py M
                                                                                                        O base,html M
                                 templates > ♦ register.html > ♦ div.container.mt-5

∨ AUTOMATINGCOMMONTASKS

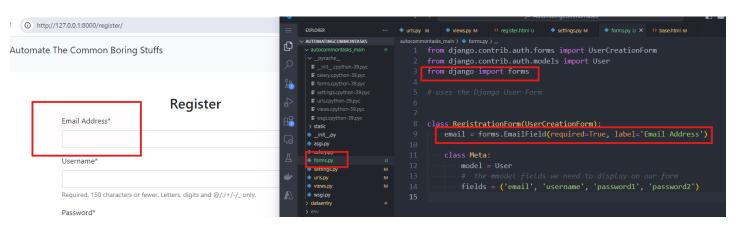
Ð
                                   1 {%extends 'base.html' %}

    autocommontasks_main

                                    2 {% block content %}
       ■ celery.cpython-39.pyc
                                       {% load crispy_forms_tags%}
                                        <div class="container mt-5" style="max-width: 600px;">
                                             <h2 class="text-center">Register</h2>
      > static
      __init__.py
      asgi.py
      celery.py
                                                  <input type="submit" value="Register" class="btn btn-primary">
      forms.pv
                                             </form>
      settings.py
                                       </div>
      views.py
      wsgi.py
      > dataentry
```



15. In the default User Form, the email is an optional field. We can make this as a required field by updating our FORMS.PY



## 16. Our VIEWS.PY as:

```
tocommontasks_main > ♠ views.py > ۞ register

from django.shortcuts import render, redirect

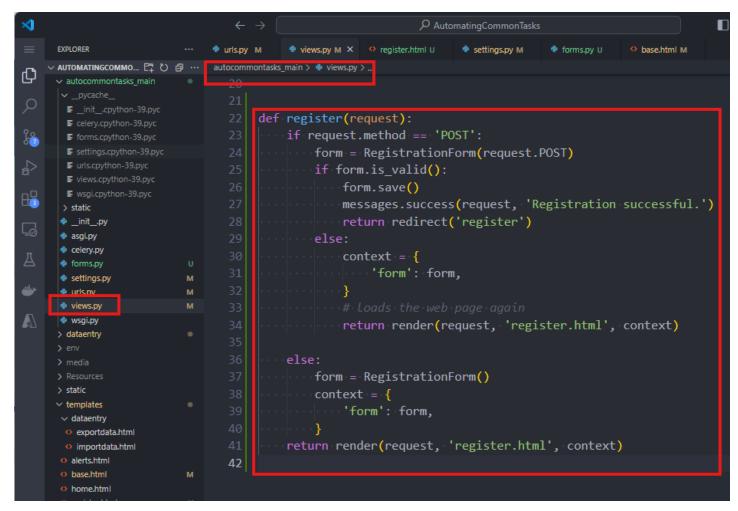
from django.http import HttpResponse

#*imports the function from the TASKS.PY of DATA

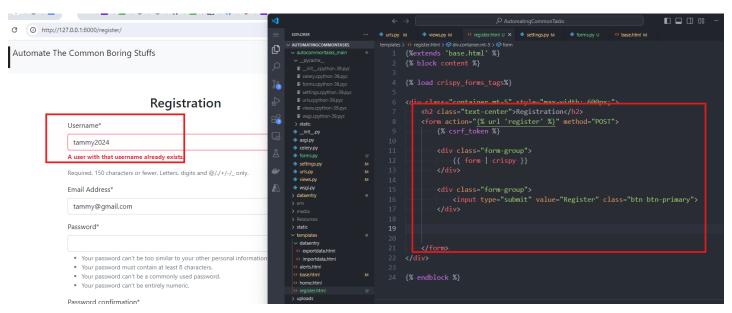
from dataentry.tasks import celery_test_task

from forms import RegistrationForm

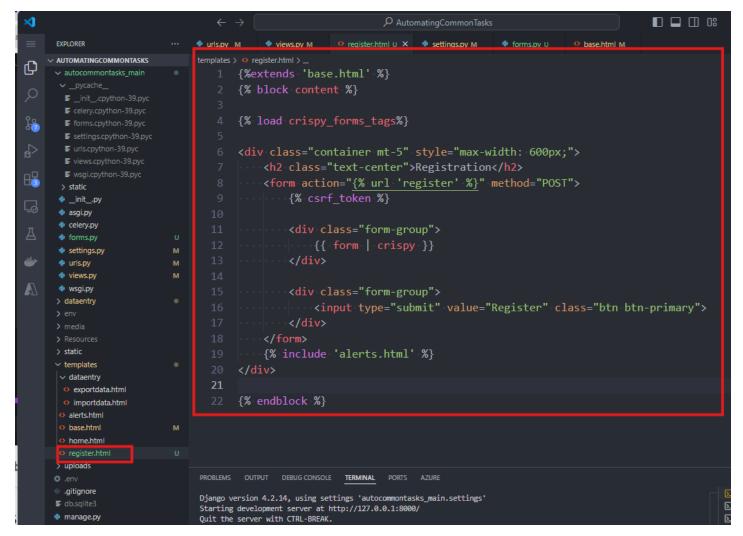
from django.contrib import messages
```



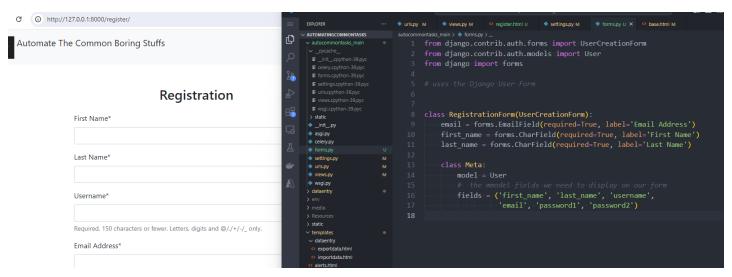
17. Testing the registration form with an existing user:



17. To include our message alert for successful registration.



18. I updated the form to include the firstname and the lastname in the registration.



19.