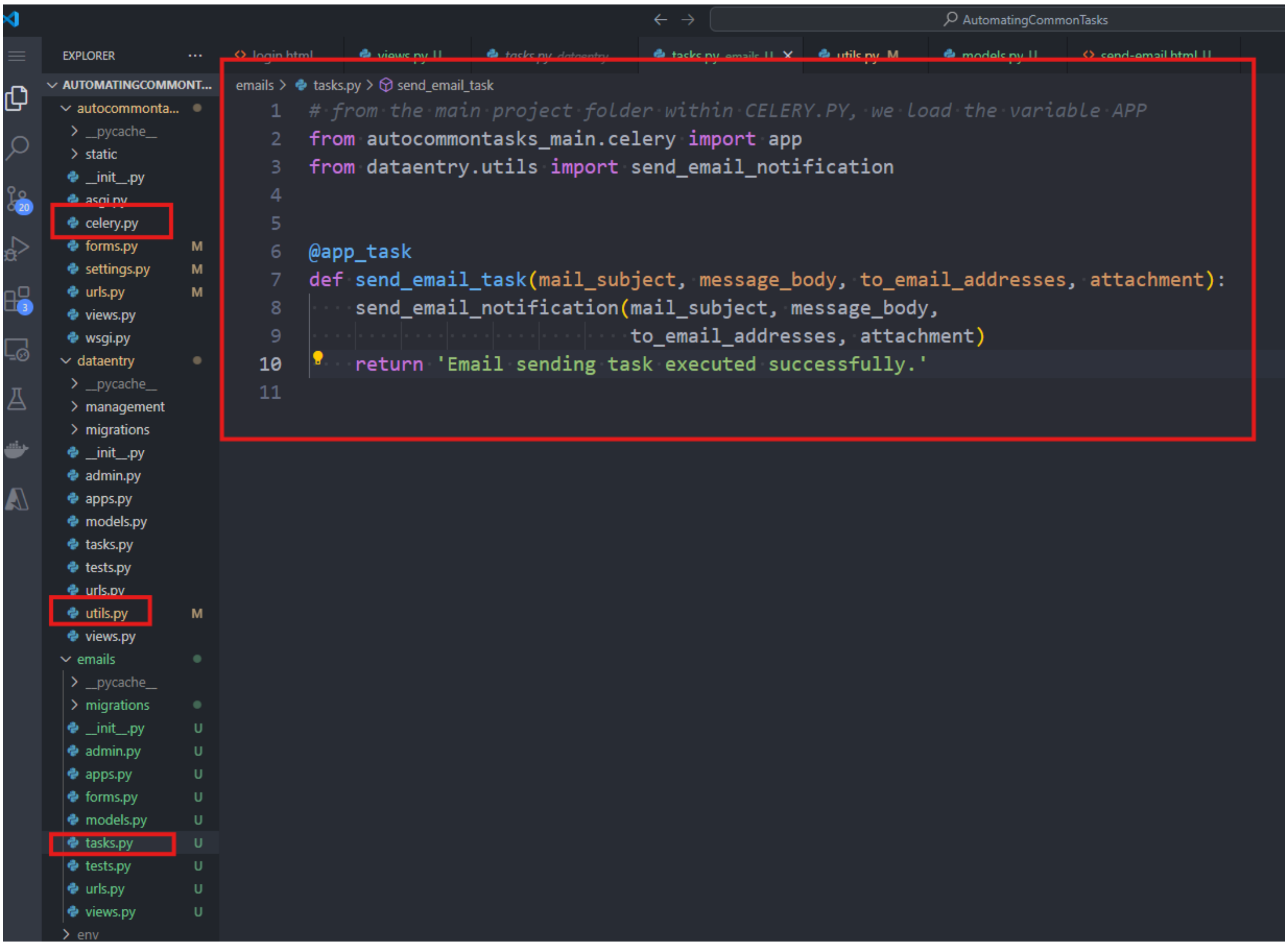
Topic: Bulk Email Tool 18: Handing the Task To Celery

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

---

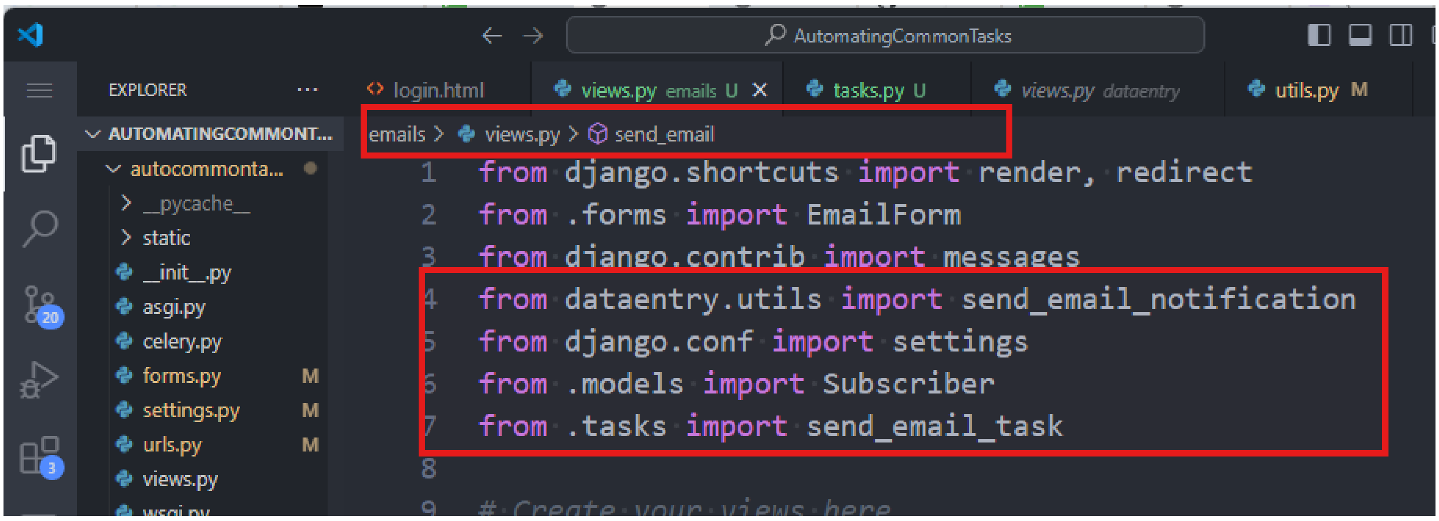
Sending bulk emails will take awhile especially if its thousands or millions of email addresses. To resolve this, we use CELERY again just like how we used it in DATAENTRY to import or export large data.

1. In EMAILS folder, create a new file, TASKS.PY

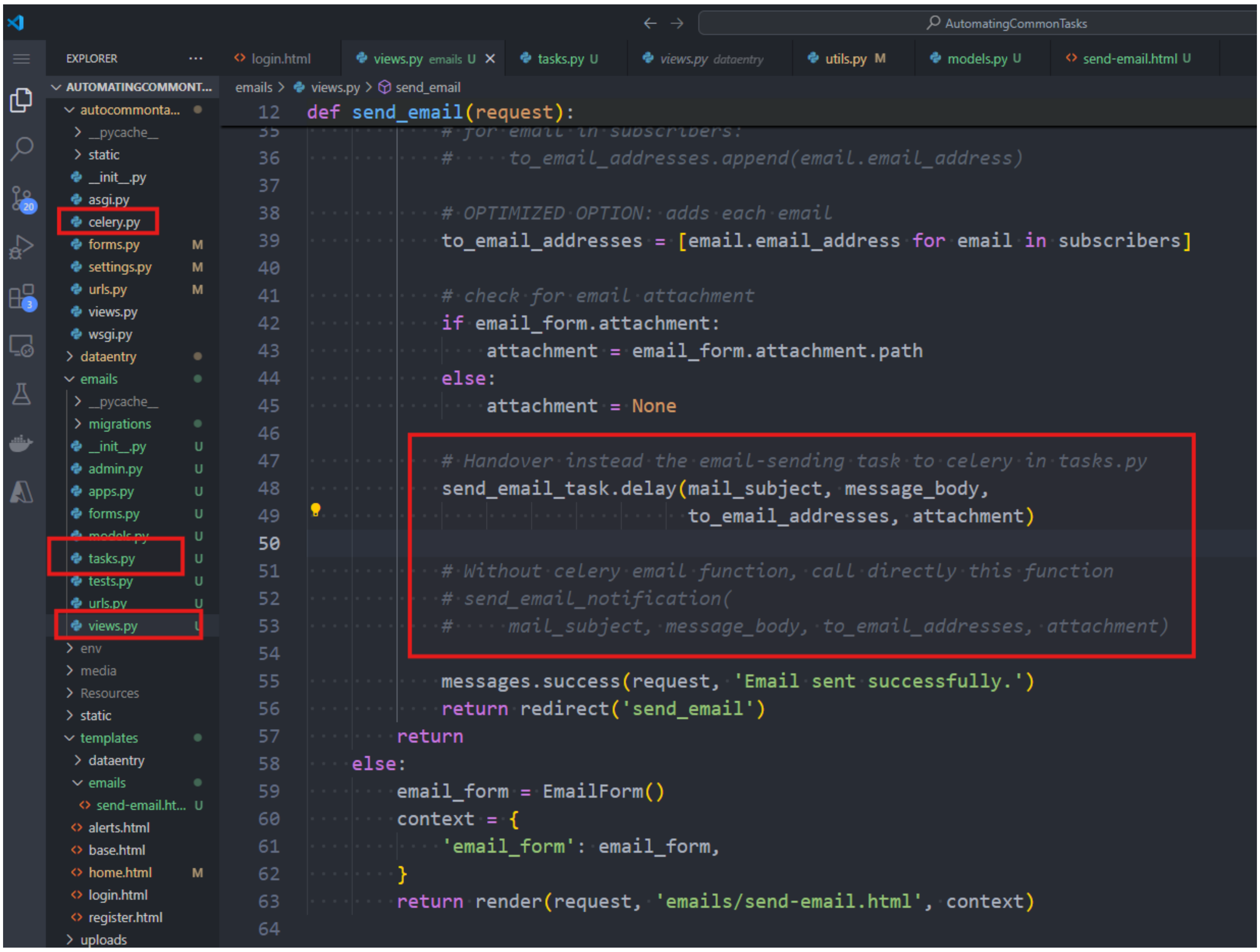


```python
# from the main project folder within CELERY.PY, we load the variable APP
from autocommontasks_main.celery import app
from dataentry.utils import send_email_notification


@app_task
def send_email_task(mail_subject, message_body, to_email_addresses, attachment):
    send_email_notification(mail_subject, message_body,
                            to_email_addresses, attachment)
    return 'Email sending task executed successfully.'
```

2. Update our EMAILS\VIEWS.PY to call the celery function in EMAILS\VIEWS.PY

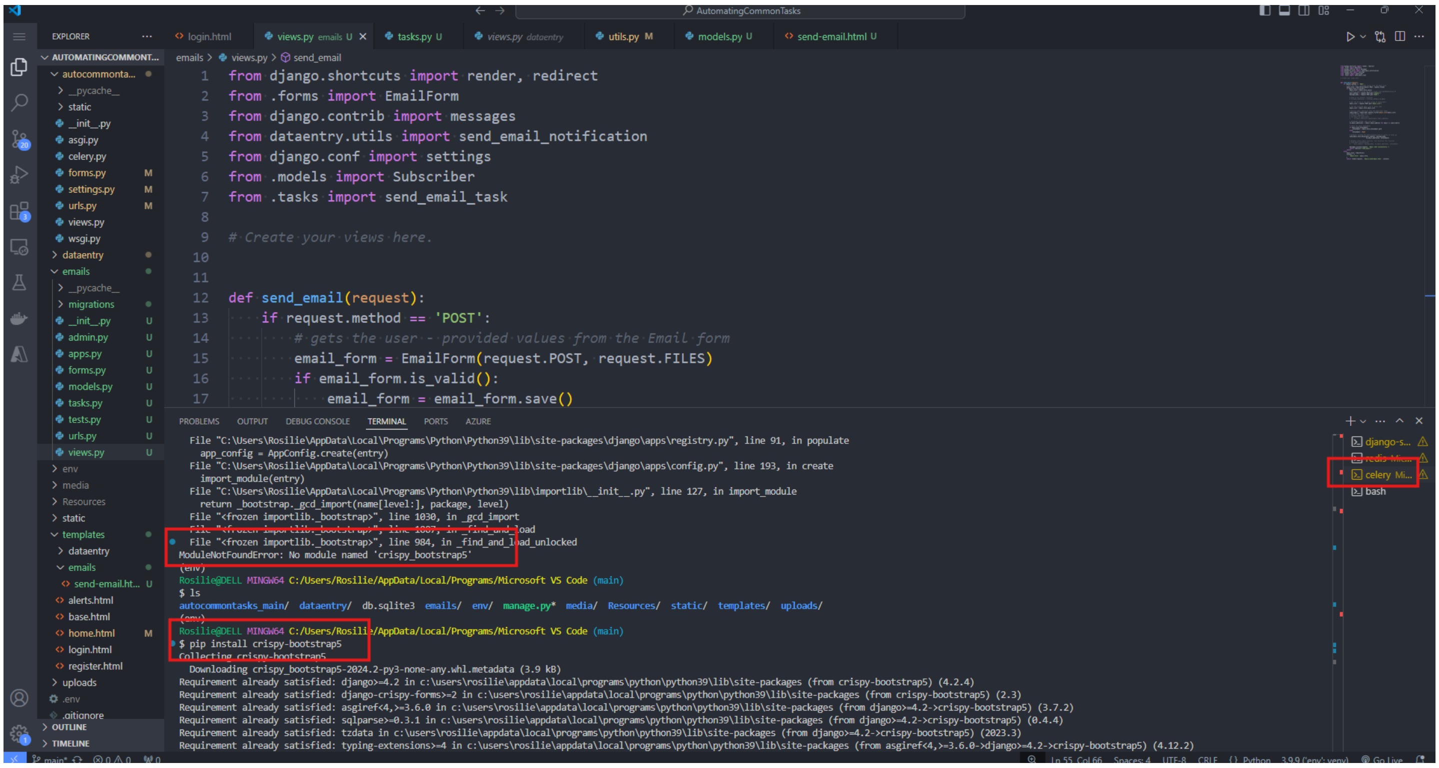


```
emails > views.py > send_email
from django.shortcuts import render, redirect
from .forms import EmailForm
from django.contrib import messages
from dataentry.utils import send_email_notification
from django.conf import settings
from .models import Subscriber
from .tasks import send_email_task

# Create your views here
```



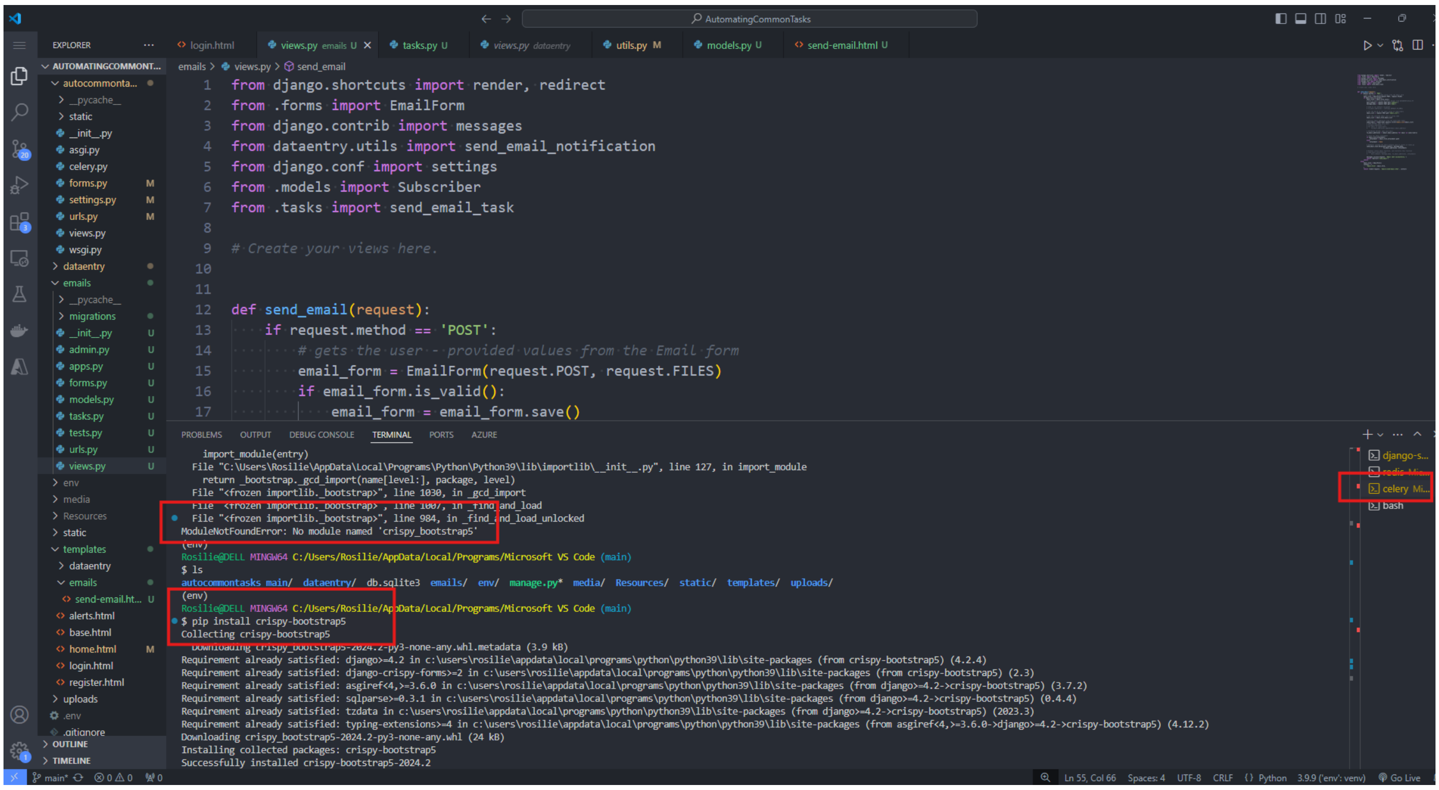
```
emails > views.py > send_email
def send_email(request):
        # for email in subscribers:
        #     to_email_addresses.append(email.email_address)

        # OPTIMIZED OPTION: adds each email
        to_email_addresses = [email.email_address for email in subscribers]

        # check for email attachment
        if email_form.attachment:
            attachment = email_form.attachment.path
        else:
            attachment = None

        # Handover instead the email-sending task to celery in tasks.py
        send_email_task.delay(mail_subject, message_body,
                                  to_email_addresses, attachment)

        # Without celery email function, call directly this function
        # send_email_notification(
        #     mail_subject, message_body, to_email_addresses, attachment)

        messages.success(request, 'Email sent successfully.')
        return redirect('send_email')
        return
    else:
        email_form = EmailForm()
        context = {
            'email_form': email_form,
        }
        return render(request, 'emails/send-email.html', context)
```

3. Go to your CELERY BASH TERMINAL, press CTRL + C to stop it from running. Now execute the command to reload CeleryIMPORTANT REMINDER: When you run your celery terminal, it has error notifications for installed packages, CRISPY FORM AND CRISPY BOOTSTRAP5, so we installed it again..



```python
from django.shortcuts import render, redirect
from .forms import EmailForm
from django.contrib import messages
from dataentry.utils import send_email_notification
from django.conf import settings
from .models import Subscriber
from .tasks import send_email_task

# Create your views here.


def send_email(request):
    if request.method == 'POST':
        # gets the user - provided values from the Email form
        email_form = EmailForm(request.POST, request.FILES)
        if email_form.is_valid():
            email_form = email_form.save()
```

```
  File "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\site-packages\django\apps\registry.py", line 91, in populate
    app_config = AppConfig.create(entry)
  File "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\site-packages\django\apps\config.py", line 193, in create
    import_module(entry)
  File "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\importlib\__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
  File "<frozen importlib._bootstrap>", line 984, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'crispy_bootstrap5'
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)
$ ls
autocommontasks_main/  dataentry/  db.sqlite3  emails/  env/  manage.py*  media/  Resources/  static/  templates/  uploads/
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)
$ pip install crispy-bootstrap5
Collecting crispy-bootstrap5
  Downloading crispy_bootstrap5-2024.2-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: django>=4.2 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from crispy-bootstrap5) (4.2.4)
Requirement already satisfied: django-crispy-forms>=2 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from crispy-bootstrap5) (2.3)
Requirement already satisfied: asgiref<4,>=3.6.0 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (3.7.2)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (0.4.4)
Requirement already satisfied: tzdata in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (2023.3)
Requirement already satisfied: typing-extensions>=4 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from asgiref<4,>=3.6.0->django>=4.2->crispy-bootstrap5) (4.12.2)
```

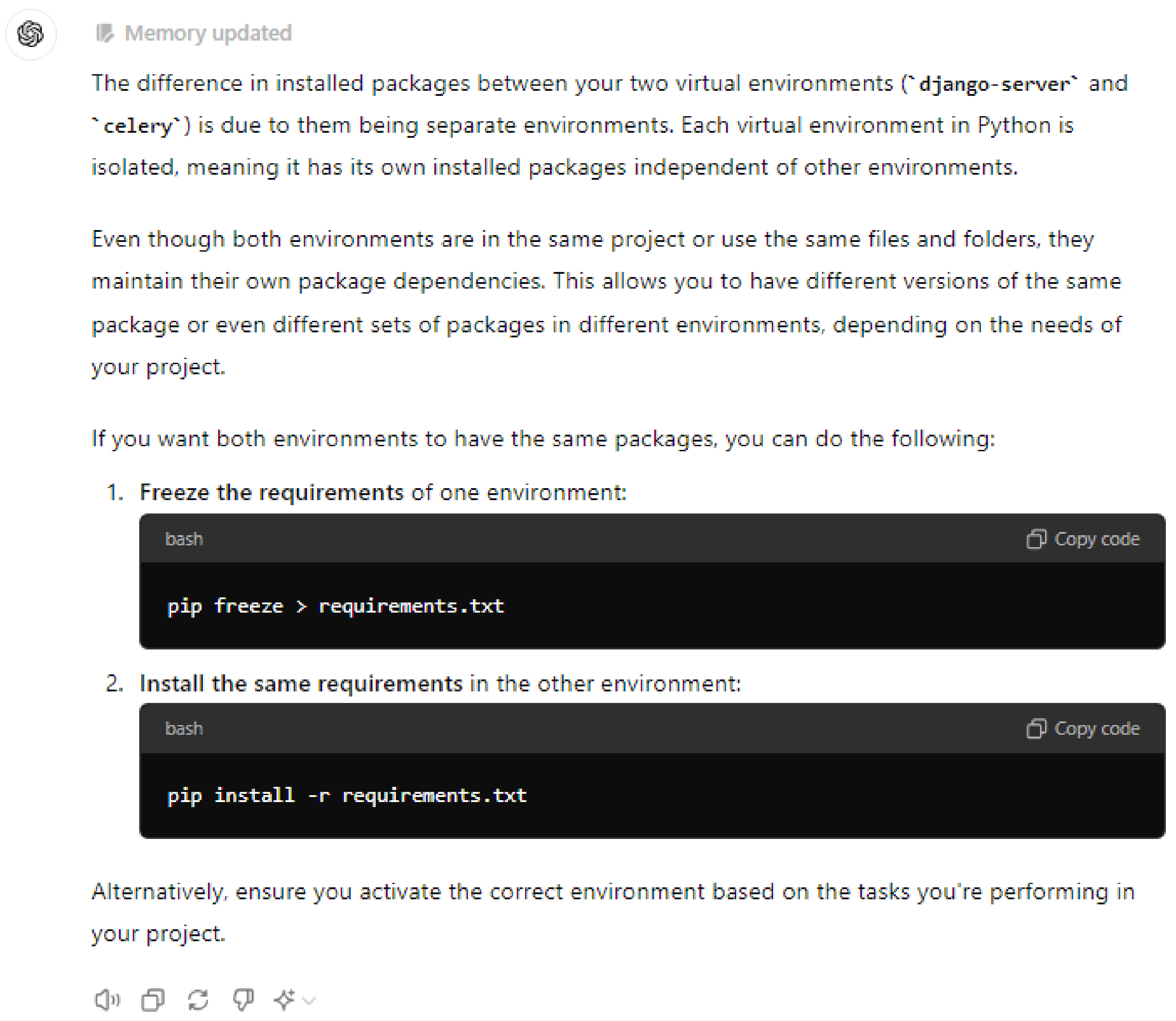


```python
from django.shortcuts import render, redirect
from .forms import EmailForm
from django.contrib import messages
from dataentry.utils import send_email_notification
from django.conf import settings
from .models import Subscriber
from .tasks import send_email_task

# Create your views here.


def send_email(request):
    if request.method == 'POST':
        # gets the user - provided values from the Email form
        email_form = EmailForm(request.POST, request.FILES)
        if email_form.is_valid():
            email_form = email_form.save()
```

```
    import_module(entry)
  File "C:\Users\Rosilie\AppData\Local\Programs\Python\Python39\lib\importlib\__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
  File "<frozen importlib._bootstrap>", line 984, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'crispy_bootstrap5'
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)
$ ls
autocommontasks_main/  dataentry/  db.sqlite3  emails/  env/  manage.py*  media/  Resources/  static/  templates/  uploads/
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)
$ pip install crispy-bootstrap5
Collecting crispy-bootstrap5
  Downloading crispy_bootstrap5-2024.2-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: django>=4.2 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from crispy-bootstrap5) (4.2.4)
Requirement already satisfied: django-crispy-forms>=2 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from crispy-bootstrap5) (2.3)
Requirement already satisfied: asgiref<4,>=3.6.0 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (3.7.2)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (0.4.4)
Requirement already satisfied: tzdata in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (2023.3)
Requirement already satisfied: typing-extensions>=4 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from asgiref<4,>=3.6.0->django>=4.2->crispy-bootstrap5) (4.12.2)
Downloading crispy_bootstrap5-2024.2-py3-none-any.whl (24 kB)
Installing collected packages: crispy-bootstrap5
Successfully installed crispy-bootstrap5-2024.2
```

IT APPEARS THAT THE VIRTUAL ENVIRONMENT OF DJANGO-SERVER TERMINAL AND CELERY TERMINALS ARE DIFFERENT.

Plugging this confusion in CHATGPT, it shows this message:



Memory updated

The difference in installed packages between your two virtual environments (`django-server` and `celery`) is due to them being separate environments. Each virtual environment in Python is isolated, meaning it has its own installed packages independent of other environments.

Even though both environments are in the same project or use the same files and folders, they maintain their own package dependencies. This allows you to have different versions of the same package or even different sets of packages in different environments, depending on the needs of your project.

If you want both environments to have the same packages, you can do the following:

1. **Freeze the requirements** of one environment:

```bash
pip freeze > requirements.txt
```

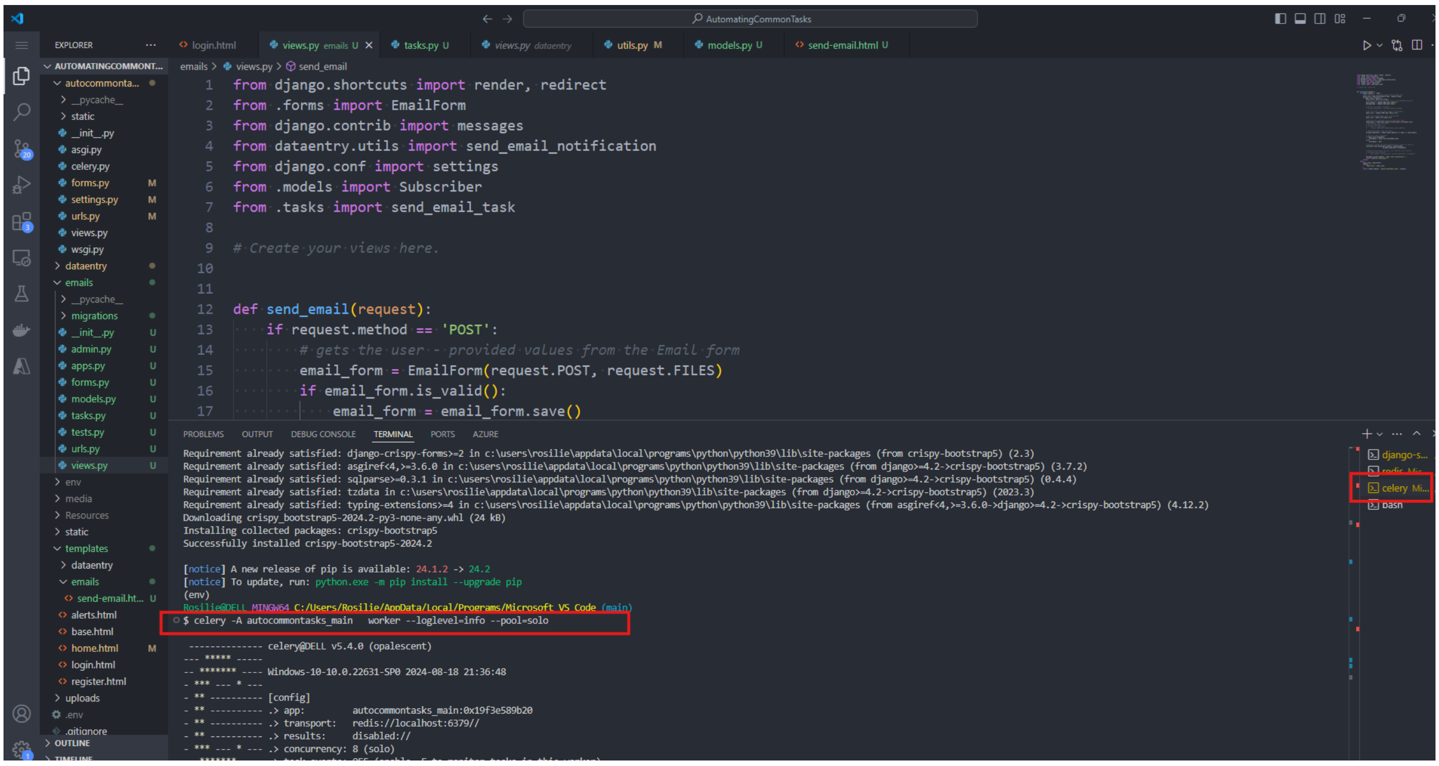
2. **Install the same requirements** in the other environment:

```bash
pip install -r requirements.txt
```

Alternatively, ensure you activate the correct environment based on the tasks you're performing in your project.

4 We run our celery command again:

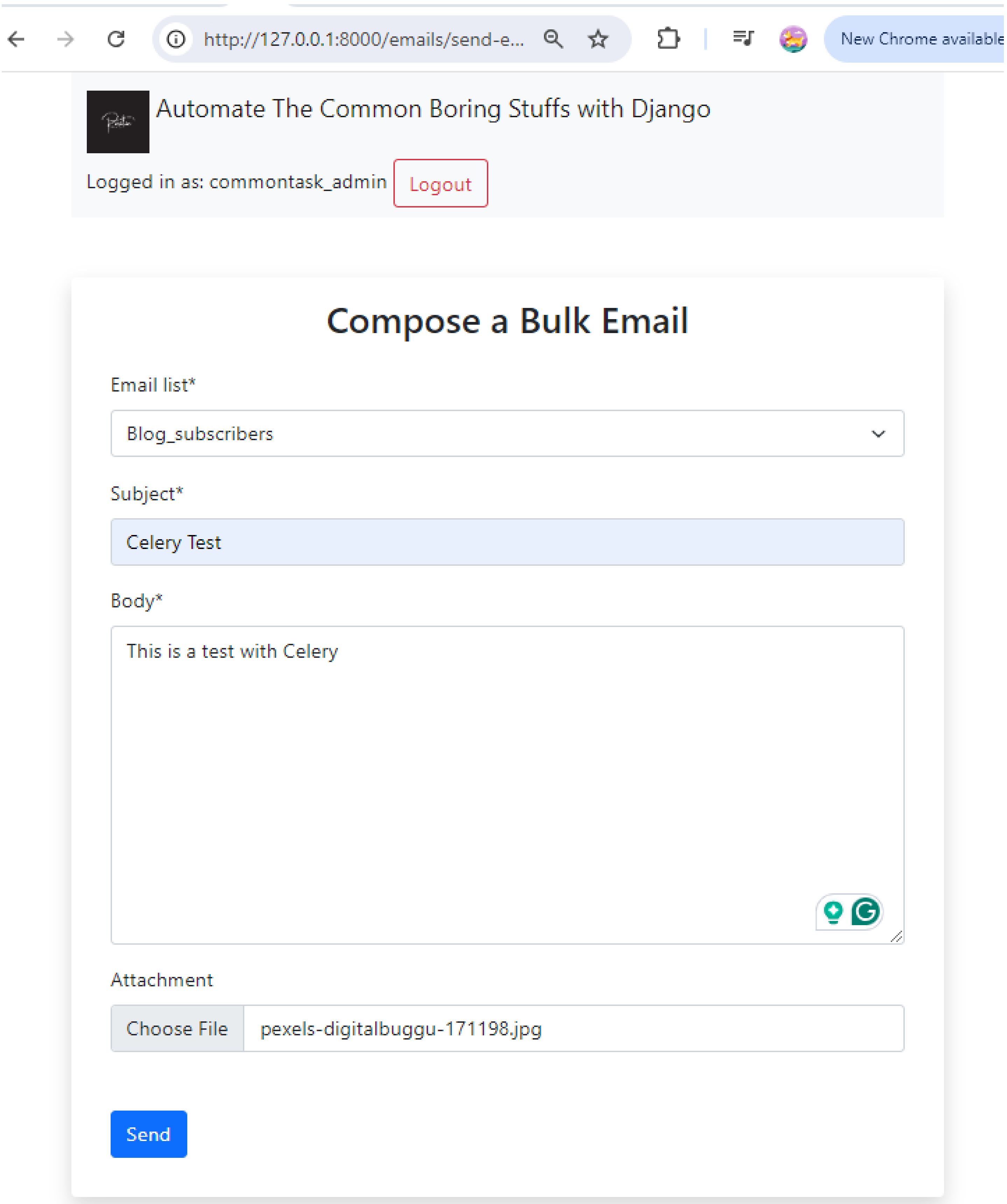
```
$ celery -A autocommontasks_main   worker --loglevel=info --pool=solo
```



```python
from django.shortcuts import render, redirect
from .forms import EmailForm
from django.contrib import messages
from dataentry.utils import send_email_notification
from django.conf import settings
from .models import Subscriber
from .tasks import send_email_task

# Create your views here.


def send_email(request):
    if request.method == 'POST':
        # gets the user - provided values from the Email form
        email_form = EmailForm(request.POST, request.FILES)
        if email_form.is_valid():
            email_form = email_form.save()
```

```
Requirement already satisfied: django-crispy-forms>=2 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from crispy-bootstrap5) (2.3)
Requirement already satisfied: asgiref<4,>=3.6.0 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (3.7.2)
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (0.4.4)
Requirement already satisfied: tzdata in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from django>=4.2->crispy-bootstrap5) (2023.3)
Requirement already satisfied: typing-extensions>=4 in c:\users\rosilie\appdata\local\programs\python\python39\lib\site-packages (from asgiref<4,>=3.6.0->django>=4.2->crispy-bootstrap5) (4.12.2)
Downloading crispy_bootstrap5-2024.2-py3-none-any.whl (24 kB)
Installing collected packages: crispy-bootstrap5
Successfully installed crispy-bootstrap5-2024.2

[notice] A new release of pip is available: 24.1.2 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(env)
Rosilie@DELL MINGW64 C:/Users/Rosilie/AppData/Local/Programs/Microsoft VS Code (main)
$ celery -A autocommontasks_main   worker --loglevel=info --pool=solo

 -------------- celery@DELL v5.4.0 (opalescent)
--- ***** -----
-- ******* ---- Windows-10-10.0.22631-SP0 2024-08-18 21:36:48
- *** --- * ---
- ** ---------- [config]
- ** ---------- .> app:         autocommontasks_main:0x19f3e589b20
- ** ---------- .> transport:   redis://localhost:6379//
- ** ---------- .> results:     disabled://
- *** --- * --- .> concurrency: 8 (solo)
```

We load our URL link: `http://127.0.0.1:8000/emails/send-email/`


http://127.0.0.1:8000/emails/send-e...

New Chrome available

Automate The Common Boring Stuffs with Django

Logged in as: commontask_admin Logout

Compose a Bulk Email

Email list*

Blog_subscribers

Subject*

Celery Test

Body*

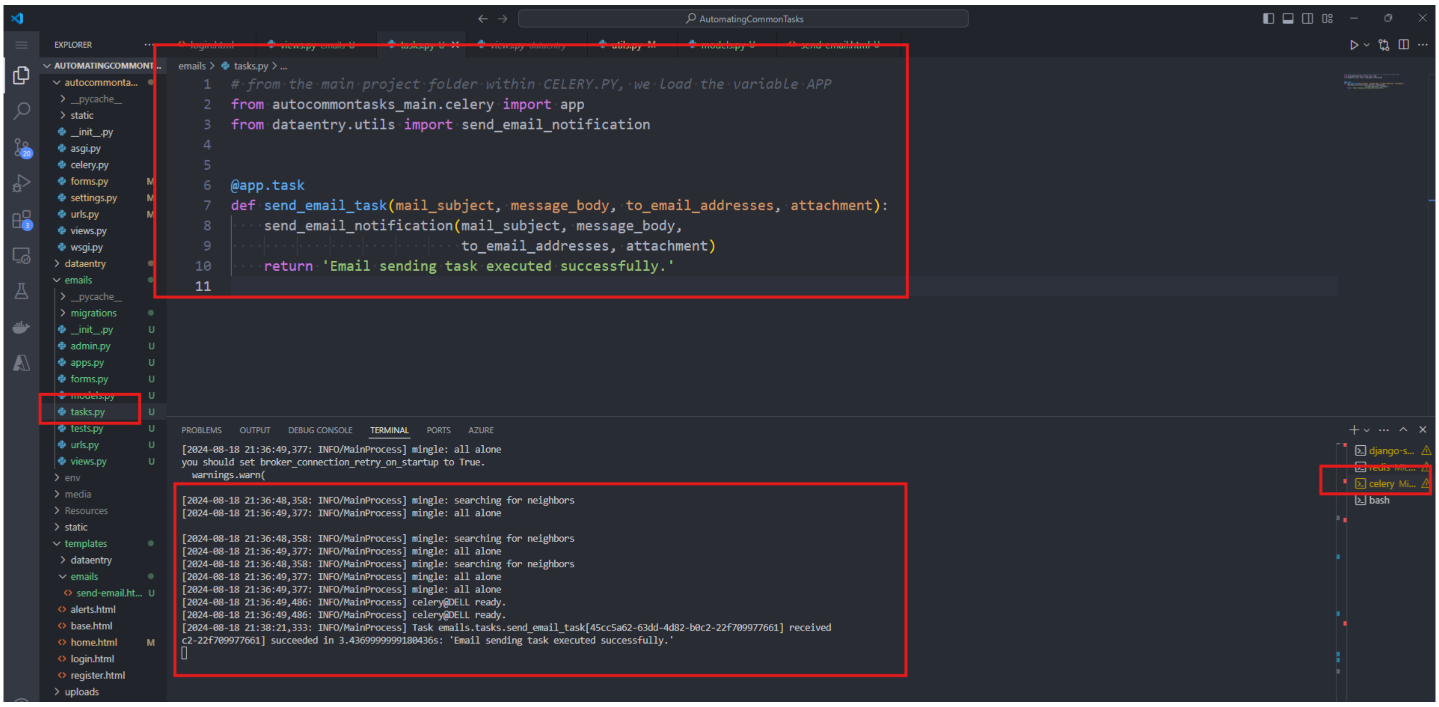
This is a test with Celery

Attachment

Choose File pexels-digitalbuggu-171198.jpg

Send

Our Celery terminal shows this:



```python
# from the main project folder within CELERY.PY, we load the variable APP
from autocommontasks_main.celery import app
from dataentry.utils import send_email_notification


@app.task
def send_email_task(mail_subject, message_body, to_email_addresses, attachment):
    send_email_notification(mail_subject, message_body,
                            to_email_addresses, attachment)
    return 'Email sending task executed successfully.'
```

```
[2024-08-18 21:36:49,377: INFO/MainProcess] mingle: all alone
you should set broker_connection_retry_on_startup to True.
  warnings.warn(

[2024-08-18 21:36:48,358: INFO/MainProcess] mingle: searching for neighbors
[2024-08-18 21:36:49,377: INFO/MainProcess] mingle: all alone

[2024-08-18 21:36:48,358: INFO/MainProcess] mingle: searching for neighbors
[2024-08-18 21:36:49,377: INFO/MainProcess] mingle: all alone
[2024-08-18 21:36:48,358: INFO/MainProcess] mingle: searching for neighbors
[2024-08-18 21:36:49,377: INFO/MainProcess] mingle: all alone
[2024-08-18 21:36:49,377: INFO/MainProcess] mingle: all alone
[2024-08-18 21:36:49,486: INFO/MainProcess] celery@DELL ready.
[2024-08-18 21:36:49,486: INFO/MainProcess] celery@DELL ready.
[2024-08-18 21:38:21,333: INFO/MainProcess] Task emails.tasks.send_email_task[45cc5a62-63dd-4d82-b0c2-22f709977661] received
c2-22f709977661] succeeded in 3.4369999999180436s: 'Email sending task executed successfully.'
```

5.