

1. Django Built-in Email System

Django has a **powerful built-in email framework**. You don't need any extra packages for basic email sending.

⚙️ 2. Configure Email Settings in `settings.py`

✅ Option 1 — Using Gmail (most common)

python

Copy code

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'your_email@gmail.com'
EMAIL_HOST_PASSWORD = 'your_app_password'
```

🔒 Important:

Don't use your normal Gmail password.

You must create an **App Password** from your Google Account → Security → App Passwords.
(Enable 2FA first.)

✉️ 3. Sending an Email

You can send an email anywhere (views, tasks, management commands, etc.):

python

```
from django.core.mail import send_mail

send_mail(
    subject='Welcome to MySite!',
    message='Thank you for signing up.',
    from_email='your_email@gmail.com',
    recipient_list=['user@example.com'],
    fail_silently=False,
)
```

4. Sending HTML Emails

You can include both plain text and HTML versions:

```
python

from django.core.mail import EmailMultiAlternatives

subject = 'Welcome!'
from_email = 'your_email@gmail.com'
to = ['user@example.com']

text_content = 'Thank you for joining our site!'
html_content = '<h2>Welcome!</h2><p>Thanks for signing up 🚀</p>'

msg = EmailMultiAlternatives(subject, text_content, from_email, to)
msg.attach_alternative(html_content, "text/html")
msg.send()
```

The recipient's email client will automatically choose the **HTML** version if it supports it — otherwise, it falls back to the plain text.

7. Sending Attachments

```
python

from django.core.mail import EmailMessage

email = EmailMessage(
    subject='Monthly Report',
    body='Attached is your report.',
    from_email='your_email@gmail.com',
    to=['user@example.com'],
)
email.attach_file('reports/monthly.pdf')
email.send()
```

Sending a HTML file with Dynamic content

2. Load and Render Template in View

You'll use Django's `render_to_string()` to convert your HTML template into a string with variables replaced dynamically.

```
from django.core.mail import EmailMultiAlternatives
from django.template.loader import render_to_string
from django.utils.html import strip_tags

def send_dynamic_email(request):
    context = {
        'user_name': 'Ajay',
        'user_email': 'ajay@example.com',
        'site_name': 'MyAwesomeSite'
    }

    html_content = render_to_string('emails/welcome_email.html', context)
    text_content = strip_tags(html_content) # Fallback text for plain email clients

    email = EmailMultiAlternatives(
        subject='Welcome to MyAwesomeSite!',
        body=text_content,
        from_email='your_email@gmail.com',
        to=['ajay@example.com'],
    )
    email.attach_alternative(html_content, "text/html")
    email.send()

    return HttpResponse("Email sent with HTML template!")
```