STEP 1: Use App-Level Template Namespacing (VERY IMPORTANT)

To avoid template name conflicts between `pages` and `pages1`, structure templates like this:

```
pages/
    templates/
        pages/
            home.html
            about.html
            contact.html

pages1/
    templates/
        pages1/
            home.html
            about.html
            contact.html
```

□ Why?
If both apps have `home.html`, Django won't know which one to load.
By using folders (`pages/` and `pages1/`), you namespace them.

□ STEP 2: Configure settings.py

Open:

`project/settings.py`

1□ Add both apps in INSTALLED_APPS:

```
INSTALLED_APPS = [
    ...
    'pages',
    'pages1',
]
```

2□ Make sure TEMPLATES setting has:

```
'APP_DIRS': True,
```

Example:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

□ STEP 3: Create views.py in each app

□ pages/views.py

```
from django.shortcuts import render

def home(request):
    return render(request, 'pages/home.html')

def about(request):
    return render(request, 'pages/about.html')

def contact(request):
    return render(request, 'pages/contact.html')
```

□ pages1/views.py

```
from django.shortcuts import render

def home(request):
    return render(request, 'pages1/home.html')

def about(request):
    return render(request, 'pages1/about.html')

def contact(request):
    return render(request, 'pages1/contact.html')
```

Notice:

- `pages/home.html`
- `pages1/home.html`

This prevents template conflicts.

□ STEP 4: Create urls.py in each app

□ pages/urls.py

```
from django.urls import path
from . import views

app_name = 'pages'

urlpatterns = [
    path('', views.home, name='home'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
]
```

□ pages1/urls.py

```
from django.urls import path
from . import views

app_name = 'pages1'

urlpatterns = [
    path('', views.home, name='home'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
]
```

□ `app_name` is VERY important to avoid URL name conflicts.

□ STEP 5: Configure Main Project urls.py

Open:

`project/urls.py`

Add:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('pages.urls')),           # default app
    path('pages1/', include('pages1.urls')),   # second app
]
```

Now:

- `http://127.0.0.1:8000/` → pages app
- `http://127.0.0.1:8000/pages1/` → pages1 app

No conflicts □

□ STEP 6: Use URL Names Properly in Templates

Inside templates use:

For pages app:

```
<a href="{% url 'pages:about' %}">About</a>
```

For pages1 app:

```
<a href="{% url 'pages1:about' %}">About</a>
```

□ This avoids URL name conflicts.

□ FINAL PROJECT STRUCTURE (Correct One)

```
project/
    project/
        settings.py
        urls.py
    pages/
        templates/
            pages/
                home.html
                about.html
                contact.html
        views.py
        urls.py
    pages1/
        templates/
            pages1/
                home.html
                about.html
                contact.html
        views.py
        urls.py
```

□ Conflict Prevention Checklist

✔ Use app_name in urls.py
✔ Use namespaced templates (pages/home.html)
✔ Use include() in main urls.py
✔ Use {% url 'appname:viewname' %}
✔ Keep APP_DIRS = True

If you want to use a **normal way without namespacing (`pages:about`)**, you can — but you must avoid conflicts carefully.

I’ll show you the correct and safe way □

□ OPTION 1: Using Normal URL Name (No `app_name`)

Instead of:

```
<a href="{% url 'pages:about' %}">About</a>
```

You can use:

```
<a href="{% url 'about' %}">About</a>
```

But to make this work properly, follow these steps.

□ STEP 1: Remove `app_name` from urls.py

□ pages/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
]
```

□ pages1/urls.py

□ IMPORTANT: You cannot use same names (`home`, `about`, `contact`) in both apps if you remove namespace.

So change names in second app:

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('', views.home, name='home1'),
    path('about/', views.about, name='about1'),
    path('contact/', views.contact, name='contact1'),
]
```

□ STEP 2: Main project urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('pages.urls')),
    path('pages1/', include('pages1.urls')),
]
```

□ STEP 3: Use in Templates

In pages app template:

```
<a href="{% url 'about' %}">About</a>
<a href="{% url 'contact' %}">Contact</a>
```

In pages1 app template:

```
<a href="{% url 'about1' %}">About</a>
<a href="{% url 'contact1' %}">Contact</a>
```

□ VERY IMPORTANT

If both apps use:

```
name='about'
```

without `app_name`, Django will throw:

```
Reverse for 'about' is ambiguous
```

So you must:

- Either use different names (`about1`, `about2`)
- OR use namespace (`pages:about`) ← Recommended way

□ Which Method is Better?

Method	Safe?	Recommended?
Without namespace	□ Risky in big projects	No
With namespace (`pages:about`)	□ Very safe	YES

For small learning projects → normal way is fine.
For real projects → always use namespace.