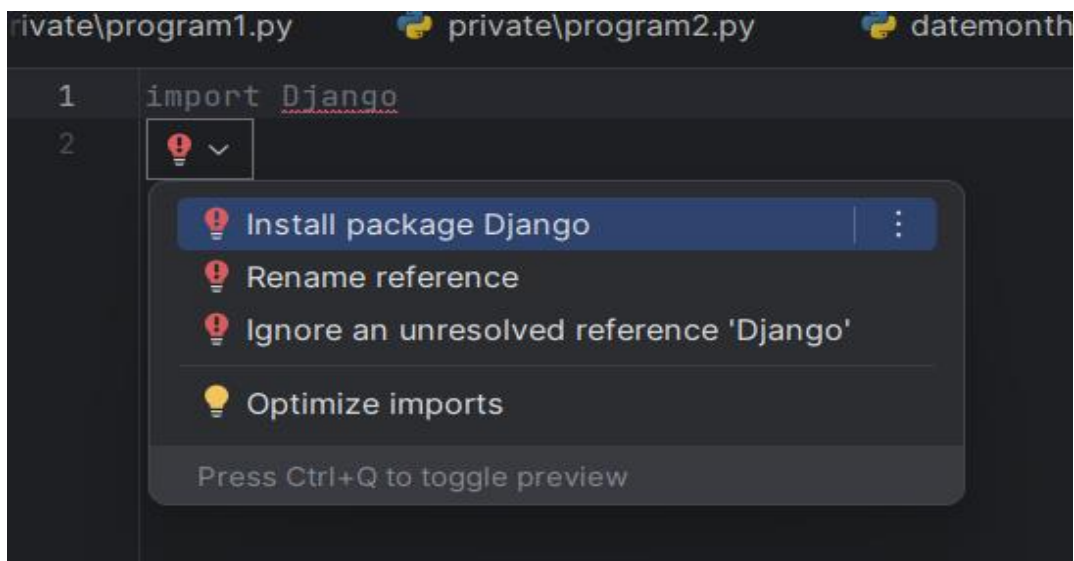


Step1 : Setting up Django:



- Click on install package to install Django

Create a Django project:

```
django-admin startproject myblog
cd myblog
```

Step2 : Create Django app for the blog:

```
python manage.py startapp blog
```

1. Create models,views ,templets

Blogs:

```
from django.db import models

class Post(models.Model):

    title = models.CharField(max_length=200)
    content = models.TextField()

    def __str__(self):
        return self.title
```

views:

```
from django.shortcuts import render
```

```

from .models import Post

def index(request):
    posts = Post.objects.all()
    return render(request, 'blog/index.html', {'posts': posts})

```

Templets:

```

<!DOCTYPE html>

<html>

<head>

    <title>My Blog</title>

</head>

<body>

    <h1>My Blog</h1>

    {% for post in posts %}

        <h2>{{ post.title }}</h2>

        <p>{{ post.content }}</p>

    {% empty %}

        <p>No posts available.</p>

    {% endfor %}

</body>

</html>

```

Step3. Configuration URL and templet

My blog:

```

from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('blog.urls')),
]

```

Blog :

```

from django.urls import path
from . import views

```

```
urlpatterns = [
    path("", views.index, name='index'),
]
```

Step4. user authorization and registration:

1.Update myblog/settings. py:

- Add 'django. contrib. auth', and 'django. contrib. sessions', to INSTALLED_APPS.
- Configure authentication backends:

```
AUTHENTICATION_BACKENDS = (
    'django. contrib. auth. backends. ModelBackend',
)
```

2.Create registration and login templates:

- blog/templates/registration/register. html
- blog/templates/registration/login. html

3.Implement registration and login views and URLs.

Step5 : Security and Code Quality

- Implement security measures like using CSRF tokens, validating user input, and securing passwords.
- Follow Django's best practices and coding standards while writing views, models, and templates.

Step 6: Run the Development Server

Finally, run the development server and check your blog application:

```
python manage. py runserver
```

Visit <http://localhost:8000> in your browser to see the blog posts and test the registration and login functionality.

Remember, this is a basic setup. In a real-world scenario, you would need to handle forms, implement user authentication and authorization properly, and secure your application further. Additionally, you might want to use Django's built-in admin interface for managing blog posts and users.

Make sure to refer to Django's official documentation for in-depth information on each topic:
<https://docs.djangoproject.com/en/stable/>