

# Django - Python Web Framework

Django - A High-Level Python Web Framework

What is Django?

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites.

Key Features:

- Fast development
- Built-in admin panel
- User authentication
- ORM for database interaction
- Templating system
- URL routing
- Protection against common web attacks

Architecture: MTV (Model - Template - View)

1. Model:

Defines database schema using Python classes.

Example:

```
class Post(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
```

2. View:

Contains logic for handling requests and responses.

Example:

```
def post_list(request):
    posts = Post.objects.all()
    return render(request, 'post_list.html', {'posts': posts})
```

3. Template:

HTML file that displays data passed from the view.

Example:

```
{% for post in posts %}
    <h2>{{ post.title }}</h2>
    <p>{{ post.content }}</p>
{% endfor %}
```

Why Use Django?

Reason	Benefit
Secure	Prevents common attacks like SQL Injection
Fast	Built-in tools speed up development
Scalable	Used by Instagram, Pinterest, etc.
Simple	Pythonic, clean, and easy to learn

## Django - Python Web Framework

| Extensible | Supports plugins, REST APIs (via DRF) |

### Use Case Examples:

- Blogs and news sites
- E-commerce platforms
- APIs with Django REST Framework
- CRM, ERP, and admin systems

### Summary:

Django is ideal for building powerful web applications quickly with less code, strong security, and good scalability.