

# Welcome to Day11

Python Django

# Web site

## Front-End



## Back-End

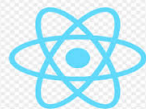


# Front-end options

## Front-End



 InterviewBit



React



ANGULAR



Vue.js

*Knockout.*

# Back-end options

## Back-End



 InterviewBit

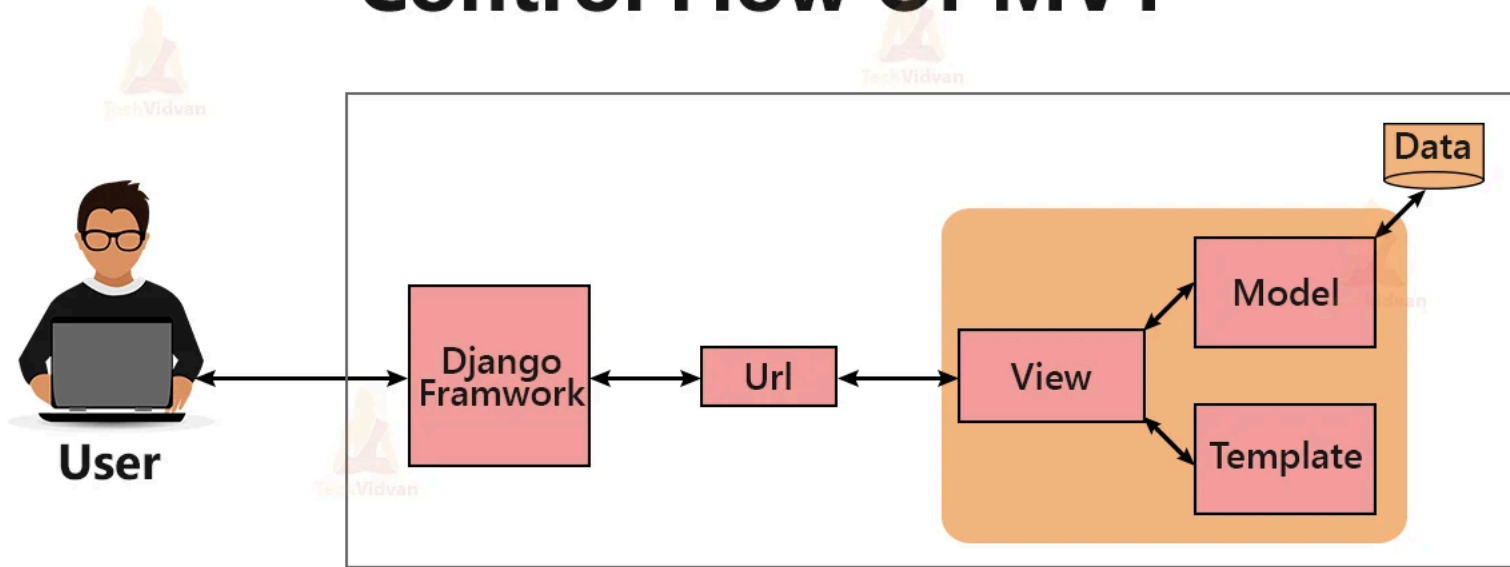


 Flask



# Request-Handling

## Control Flow Of MVT



# Why Framework?

- Quicker development (Pre-built packages)
- Consistent architecture
- Community support
- Security Handled

# Pre-requisites

- ☐ Python --version
- ☐ pip --version
- ☐ pip install Django

# Create eComm website

Mini-ecomm web

# Create project

```
django-admin startproject my_ecom
```

```
my_ecom/  
├─ manage.py  
└─ my_ecom/  
    ├─ __init__.py  
    └─ asgi.py  
        └─ settings.py  
            └─ asgi.py  
                └─ settings.py  
                    └─ urls.py  
                        └─ wsgi.py
```

# Launch project

```
cd my_ecom  
python manage.py runserver
```

# Create module

```
python manage.py startapp products
```

# Create new View

my\_ecom/products/views.py

```
from django.shortcuts import render
from django.http import HttpResponse

def products(request):
    return HttpResponse("Hello world! Welcome to products")
```

# Update URL Navigation

my\_ecom/products/product\_urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('products/', views.products, name='products'),
]
```

# Update Overall URL

my\_ecom/products/product\_urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path('products/', views.products, name='products'),
]
```

my\_ecom/my\_ecom/urls.py

```
from django.contrib import admin
from django.urls import include, path
urlpatterns = [
    path('', include('products.product_urls')),
    path('admin/', admin.site.urls),
]
```

# Hello world view

```
cd ..
```

```
my_ecom> python manage.py runserver
```

Create mini web  
page

# include template

my\_ecom/products/templates/myfirst.html

```
<html><body>
<h1>Hello World!</h1>
<p>Welcome to my first Django project!</p>
</body></html>
```

my\_ecom/products/views.py

```
from django.template import loader
...
def products(request):
    template = loader.get_template('myfirst.html')
    return HttpResponse(template.render())
```

# Settings

my\_ecom/my\_ecom/settings.py

```
INSTALLED_APPS = [  
    . . .  
    'products'  
]
```

**Include images**

# Include image

- Path of the image

```
products
  static
    cake_banana.jpg
```

my\_ecom/products/templates/myfirst.html

```
{% load static %}
<html>
<body>
<h1>Banana cake</h1>

<p>Welcome to my first Django project!</p>
</body>
</html>
```

# Image

```
my_ecom/my_ecom/settings.py
```

```
...
```

```
STATIC_URL = '/static/'
```

# Website layout

# Html layout

```
<head>
  <title>Product Listing</title>
  <style>
    header {
      background-color: #bb7409;
      color: white;
      padding: 2px;
      text-align: center;
    }
    .product-container {
      display: grid;
      grid-template-columns: repeat(3, auto);
      gap: 20px;
      padding: 20px;
    }
  </style>
</head>
```

# Html layout

```
<body>
<header> <h3>Welcome to My Store</h3>    </header>
<div class="product-container">
  <div class="product">
    <h2>Banana Cake</h2>
    
    <p>Delicious homemade banana cake</p>
  </div>
  <div class="product">
    <h2>Chocolate Cake</h2>
    
    <p>Rich and moist chocolate cake</p>
  </div>
</div>
</body>
```

# Dynamic content

# Dynamic Variables

```
1  def products(request):
2      myProducts = [
3          {"name": "Banana Cake", "price": 10.99, "image": "cake_banana.jpg"},
4          {"name": "Chocolate Cake", "price": 19.99, "image": "cake_choco.jpg"}
5      ]
6      # return HttpResponse("Hello world! Welcome to products page")
7      template = loader.get_template('myfirst.html')
8      context = {
9          "myProducts": myProducts
10     }
11     return HttpResponse(template.render(context, request))
```

# Dynamic Views

```
1  <div class="product-container">
2      {% for product in myProducts %}
3
4      <div class="product">
5          <h2>{{ product.name }}</h2>
6          
7          <p>Price: ${{ product.price }}</p>
8      </div>
9
10     {% endfor %}
11 </div>
```

**Response as json**

# json response

my\_ecom/my\_ecom/products/views.py

```
def product(request):  
    ...  
def product_json(request):  
    # In a real application, you would fetch product details from a database  
    product_details = [  
        {"id": 1, "name": "Banana Cake", "price": 10.99,  
         "description": "A delicious cake made with ripe bananas."},  
        {"id": 2, "name": "Coffee Cake", "price": 19.99,  
         "description": "A rich coffee cake with a hint of chocolate."},  
        {"id": 3, "name": "Vanilla Cake", "price": 5.99,  
         "description": "A classic vanilla cake."},  
    ]  
    return JsonResponse(product_details)
```

# include URL

my\_ecom/products/product\_urls.py

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

urlpatterns = [
    path('products/', views.products, name='products'),
    path('products/json/', views.product_json, name='product_json'),
]
```

# Individual product

## my\_ecom/my\_ecom/products/views.py

```
def product(request):  
    ...  
def product_json(request):  
    ...  
def product_json_detail(request, product_id):  
    product_details = [  
        {"id": 1, "name": "Banana Cake", "price": 10.99, "description": "A  
        {"id": 2, "name": "Coffee Cake", "price": 19.99, "description": "A  
        {"id": 3, "name": "Vanilla Cake", "price": 5.99, "description": "A  
    ]  
    product = product_details[product_id]  
    return JsonResponse(product)
```

# include URL

my\_ecom/products/product\_urls.py

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

urlpatterns = [
    path('products/', views.products, name='products'),
    path('products/json/', views.product_json, name='product_json'),
    path('products/json/<int:product_id>', views.product_json_detail,
         name='product_json_detail'),
]
```