



# Full Stack Python with Django - Beginner Demo

Understand how websites are built with Python

Presented by: MS.AMBHIKA

[PROVOKE TRAININGS]

**CALL : 96031 84671**

# Agenda

- ▶ What is Full Stack Development?
- ▶ What is Django?
- ▶ How Django Builds a Website
- ▶ Backend: Models, Views, URLs
- ▶ Frontend: Templates and Styling
- ▶ Connecting to Database
- ▶ Simple API Example
- ▶ Live Demo Idea
- ▶ Summary and Q&A

**CALL : 96031 84671**

# What is Full Stack Development?

- ▶ ‘Full Stack’ means building a complete website or app – both the parts users see and the parts behind the scenes.
- ▶ Frontend: What users see (buttons, forms, text, colors).
- ▶ Backend: What happens behind (data, logic, calculations).
- ▶ Database: Where the data is stored safely.
- ▶ Example: An online store - you see products (frontend), prices are managed (backend), and info is stored (database).

**CALL : 96031 84671**

# What is Django?

- ▶ Django is a tool (framework) built using Python that helps you build websites easily.
- ▶ It gives ready-made features like login pages, database connections, and admin panels.
- ▶ You don't need to start from scratch – Django helps you move faster.
- ▶ Used by popular sites like Instagram and Pinterest.

**CALL : 96031 84671**

# How Django Builds a Website

- ▶ Django uses a simple idea called MVT (Model - View - Template):
- ▶ Model: Defines what data looks like (e.g., name, price, email).
- ▶ View: Handles what the user asks for (e.g., show list of products).
- ▶ Template: The actual page users see (HTML).
- ▶ Together, they make a complete website!

**CALL : 96031 84671**

# Steps to Create a Django Project

- ▶ 1. Install Django using: `pip install django`
- ▶ 2. Create a new project: `django-admin startproject myapp`
- ▶ 3. Create an app inside the project: `python manage.py startapp shop`
- ▶ 4. Add your data model and connect it to the database.
- ▶ 5. Create a webpage and link it to a URL.
- ▶ 6. Run the server and open it in your browser!

**CALL : 96031 84671**

# Example: Product Listing Website

- ▶ Goal: Create a simple website that shows a list of products.
- ▶ Steps:
  - ▶ - Create a Product model with name and price fields.
  - ▶ - Add product data using Django's admin panel.
  - ▶ - Write a view to fetch and show all products.
  - ▶ - Use a template to display them nicely on a webpage.

**CALL : 96031 84671**

# Example Code: models.py

- ▶ `from django.db import models`
- ▶ `class Product(models.Model):`
- ▶ `name = models.CharField(max_length=100)`
- ▶ `price = models.FloatField()`
- ▶ `# This defines how a Product looks in the database.`

**CALL : 96031 84671**

# Example Code: Template (HTML Page)

- ▶ `<h2>Product List</h2>`
- ▶ `<ul>`
- ▶ `{% for p in products %}`
- ▶ `<li>{{ p.name }} - ${{ p.price }}</li>`
- ▶ `{% endfor %}`
- ▶ `</ul>`
  
- ▶ Django automatically replaces `{{ p.name }}` and `{{ p.price }}` with real data!

**CALL : 96031 84671**

# Adding Simple API

- ▶ APIs allow data sharing between systems or apps.
- ▶ With Django REST Framework (DRF), you can create APIs easily.
- ▶ Example: Get list of products as JSON data.
  
- ▶ `from rest_framework import viewsets`
- ▶ `class ProductViewSet(viewsets.ModelViewSet):`
- ▶  `queryset = Product.objects.all()`
- ▶  `serializer_class = ProductSerializer`
  
- ▶ Visit `/api/products/` to see data.

**CALL : 96031 84671**

# Connecting the Frontend

- ▶ You can make your website beautiful using HTML, CSS, and Bootstrap.
- ▶ You can also use JavaScript to make pages dynamic.
- ▶ Example:
- ▶ `fetch('/api/products/')`
- ▶ `.then(res => res.json())`
- ▶ `.then(data => console.log(data));`
  
- ▶ This gets data from your API and shows it on the webpage.

**CALL : 96031 84671**

# Connecting to Database

- ▶ Django connects to databases like SQLite, MySQL, or PostgreSQL.
- ▶ Default: SQLite (comes built-in).
- ▶ To apply database changes:
- ▶ `python manage.py makemigrations`
- ▶ `python manage.py migrate`
  
- ▶ Django automatically creates the tables needed for your models.

**CALL : 96031 84671**

# Live Demo Idea (For 5-10 mins)

- ▶ Show creating a simple project step-by-step:
  - ▶ 1. Start Django project
  - ▶ 2. Add a model (like Product)
  - ▶ 3. Add some data using Admin panel
  - ▶ 4. Create a template and show the data
  - ▶ 5. Run the website and show it in a browser
  
- ▶ A complete website in just a few commands!


**CALL : 96031 84671**

# Summary

- ▶  Django helps you build websites quickly with Python.
- ▶  You can handle both frontend and backend easily.
- ▶  It includes everything: Database, Admin panel, APIs, and Templates.
- ▶  Ideal for beginners and professionals alike.
  
- ▶ Start simple – build, test, and explore!

**CALL : 96031 84671**

# Q&A

- ▶ Thank you for joining!
- ▶ Any questions or feedback?
- ▶  Contact: [Your Email or Website]

**CALL : 96031 84671**

# Diagram: Django MVT Architecture



How it works:

- 1 User sends a request through the browser.
- 2 Django matches it with a URL and calls the correct View.
- 3 The View talks to the Model, which accesses the Database.
- 4 Data is passed to the Template and shown to the user.

**CALL : 96031 84671**

# Diagram: Full Stack Python Flow

**Frontend (HTML, CSS, JavaScript)**






**Backend (Python + Django)**



**Database (SQLite / MySQL / PostgreSQL)**

Explanation:

-  Frontend is what the user sees and interacts with.
-  Backend (Django) handles logic and connects with the database.
-  Database stores all the data used by the website.

**CALL : 96031 84671**