# Django Course Curriculum

Duration: 15 Days (2 Hours/Day)
Target Audience: Undergraduate Engineering Students
Prerequisite: Basic Python Programming Knowledge

## Course Overview

This course is designed to teach students the fundamentals of Django, a powerful Python web framework for building dynamic, secure, and scalable web applications. By the end of the course, students will be able to create, deploy, and manage robust web applications using Django's Model-View-Template (MVT) architecture, integrate databases, and implement user authentication and RESTful APIs for real-world projects.

## Course Structure

### Chapter 1: Django Fundamentals and Project Setup

Day 1: Introduction to Django and Web Development Basics

- What is Django? Features and advantages
- Overview of web development and MVC/MVT architecture
- Installing Django and setting up virtual environments
- Mini Assignment: Install Django and create a new project

Day 2: Creating a Basic Django Project and App

- Creating a Django project and app
- Exploring project structure and settings.py
- Running the development server
- Environment variables and django-environ introduction
- Hands-on: Create your first project and run the server

### Chapter 2: Models, ORM, and Database

Day 3: Django Models and ORM Basics

- Defining models and fields
- Introduction to Django ORM and querying basics
- Indexing and database optimization concepts
- Hands-on: Create models for a simple app

Day 4: Database Migration and Advanced Querying

- Creating and applying migrations
- Using select_related and prefetch_related to optimize queries
- Custom managers and querysets
- Hands-on: Insert sample data and optimize queries

Day 5: Testing Models and Queries

- Introduction to Django testing framework
- Writing unit tests for models and ORM queries
- Hands-on: Create tests for models and queries

## Chapter 3: Views, URLs, and Templates

Day 6: Django Views and URL Routing

- Function-based views and URL configuration
- Introduction to request and response objects
- Hands-on: Create simple views and URL mappings

Day 7: Django Templates and Static Files

- Template syntax, inheritance, and filters
- Managing static files and media files (CSS, JS, images)
- Hands-on: Create HTML templates and link static files

Day 8: Template Performance and Caching

- Template performance tips
- Introduction to Django caching (per-view, template fragment caching)
- Hands-on: Cache a view or template fragment

## Chapter 4: Forms and User Input Handling

Day 9: Django Forms Basics

- Creating and rendering forms
- Handling GET and POST requests
- Hands-on: Build a form to create and edit model instances

Day 10: Advanced Form Handling and Validation

- Form validation, error handling, and custom validators
- Customizing form widgets and appearance

- Hands-on: Add custom validation and display errors

## Chapter 5: Admin Interface and Authentication

Day 11: Django Admin Interface

- Overview of the admin site
- Customizing the admin for models and registering models
- Hands-on: Customize Django admin

Day 12: User Authentication and Authorization

- User model, login, logout, registration basics
- Permissions and groups overview
- Hands-on: Implement authentication views and protect routes

## Chapter 6: Advanced Django Features and Security

Day 13: Class-Based Views and Middleware

- Introduction to class-based views (CBVs) vs function-based views
- Writing and using custom middleware
- Hands-on: Convert FBVs to CBVs and create middleware

Day 14: Security Best Practices and Performance Optimization

- CSRF protection, XSS prevention, secure password storage
- HTTPS basics and Django security settings
- Middleware impact on performance and optimization tips
- Hands-on: Implement security measures and optimize middleware

## Chapter 7: Capstone Project and Deployment

Day 15: Capstone Project Finalization and Deployment

- Finalizing models, views, forms, and authentication in the project
- Testing and debugging the project
- Deployment overview: Heroku, DigitalOcean, or Render
- Hands-on: Deploy the Django project and test live environment

## Post-Course

- Time: 1–2 weeks
- Objective: Build a full app with Git integration, real API, testing, and hosting.
- Evaluation: Review by mentor panel
- Output: Live project + GitHub Repo + Certificate of Excellence

## Learning Resources:

1. https://www.djangoproject.com/start/
2. https://learndjango.com/
3. https://www.w3schools.com/django/
4. https://www.fullstackpython.com/django.html
5. https://realpython.com/tutorials/django/
6. https://www.wscubetech.com/resources/django/free-course