

# Outline

## 1. Introduction to Django

- **What is Django?**
    - Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
    - It follows the Model-View-Template (MVT) architectural pattern.
  - **Why use Django?**
    - Built-in features like authentication, admin panel, ORM, and security.
    - Scalable and used by large companies like Instagram, Pinterest, and Spotify.
  - **Real-world examples of Django**
    - Instagram: Handles millions of users and media uploads.
    - Pinterest: Manages user-generated content and recommendations.
    - Spotify: Uses Django for backend services.
  - **Prerequisites**
    - Basic knowledge of Python and HTML/CSS.
- 

## 2. Setting Up Django

- **Installation**
    - Install Python and pip.
    - Install Django: `pip install django` .
  - **Creating a Django Project**
    - Command: `django-admin startproject myproject` .
  - **Running the Development Server**
    - Command: `python manage.py runserver` .
    - Access the server at `http://127.0.0.1:8000/` .
  - **File Structure of a Django Project**
    - `myproject/`
      - `settings.py` : Configuration for the project.
      - `urls.py` : URL routing.
      - `wsgi.py` : WSGI configuration for deployment.
    - `manage.py` : Command-line utility for managing the project.
- 

## 3. Building a Basic Django Application

- **Creating an App**
  - Command: `python manage.py startapp myapp` .
- **File Structure of a Django App**
  - `models.py` : Define database models.
  - `views.py` : Handle business logic and HTTP responses.
  - `urls.py` : Define app-specific URLs.
  - `templates/` : Store HTML templates.
- **Creating a Simple View**
  - Example:

```
from django.http import HttpResponse

def home(request):
    return HttpResponse("Welcome to my Django app!")
```

- **Mapping URLs**

- In `myapp/urls.py` :

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

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

- Include app URLs in `myproject/urls.py` :

```
from django.urls import include, path

urlpatterns = [
    path('', include('myapp.urls')),
]
```

- **Templates and Static Files**

- Create a `templates/` folder in the app.
- Example template: `myapp/templates/home.html` .
- Serve static files (CSS, JS, images) using the `static` directory.

---

## 4. Django Authentication

- **Why Authentication is Important**

- Securely manage user accounts and permissions.

- **Django's Built-in Authentication System**

- User model: `django.contrib.auth.models.User` .
- Authentication views: Login, Logout, Signup.

- **Implementing User Registration**

- Create a registration form.
- Example:

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class SignUpForm(UserCreationForm):
    email = forms.EmailField()
```

```
class Meta:
    model = User
    fields = ['username', 'email', 'password1', 'password2']
```

- **Implementing User Login and Logout**
    - Use Django's built-in views: `LoginView`, `LogoutView`.
  - **Protecting Views with Authentication**
    - Use the `@login_required` decorator.
- 

## 5. Project Ideas for College Students

- **Beginner Projects**
    - To-Do List App: Create, update, and delete tasks.
    - Blog Platform: Write and publish blog posts.
  - **Intermediate Projects**
    - E-commerce Website: Product listings, cart, and checkout.
    - Social Media App: User profiles, posts, and comments.
  - **Advanced Projects**
    - Online Quiz System: Create quizzes and track scores.
    - Job Portal: Post jobs and apply for them.
- 

## Additional Resources

- **Documentation:** [Django Official Documentation](#)
- **Tutorials:** Django for Beginners by William S. Vincent.
- **Youtube:** [Error By Night](#), [Coding For All](#)
- **Community:** Join Django forums and Slack channels.