**Building a Sports Website with Django**

**1. Set Up Your Development Environment**

**1.1 Install Python and Django**

Ensure Python is installed on your system. Install Django using pip:

pip install django

**1.2 Create a Project Directory**

**Create a new directory for the project and navigate into it:**

cd sports\_website

**1.3 Create a Django Project**

Initialize a new Django project:

django-admin startproject sports\_website

**1.4 Create a Virtual Environment:**

Set up a virtual environment for managing dependencies:

Virtualenv venv

venv\Scripts\activate

**2. Create a Django App**

**2.1 Create a New App**

Within the project directory, create a new app named `sports`:

python manage.py startapp sports

**2.2 Add the App to Project Settings**

Open `sports\_website/settings.py` and add `'sports'` to the `INSTALLED\_APPS` list:

INSTALLED\_APPS = [

...

'sports',

]

**3. Define Models**

**3.1 Create a Model**

Define the `Article` model in `sports/models.py`:

from django.db import models

class Article(models.Model):

title = models.CharField(max\_length=200)

content = models.TextField()

published\_date = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.title

Article Model: Represents a sports article with a title, content, and publication date.

**3.2 Run Migrations**

Create and apply migrations to create the database table:

python manage.py makemigrations

python manage.py migrate

**4. Create and Register the Admin Interface**

**4.1 Register the Model in Admin**

Register the `Article` model in `sports/admin.py`:

from django.contrib import admin

from .models import Article

@admin.register(Article)

class ArticleAdmin(admin.ModelAdmin):

list\_display = ('title', 'published\_date')

search\_fields = ('title', 'content')

**4.2 Create a Superuser**

Create an admin user to access the Django admin interface:

python manage.py createsuperuser

**5. Create Views and Templates**

**5.1 Define a View**

Create a view in `sports/views.py` to display articles:

from django.shortcuts import render

from .models import Article

def home(request):

articles = Article.objects.all()

return render(request, 'home.html', {'articles': articles})

home View: Retrieves all Article objects from the database and renders them using the home.html template.

**5.2 Create a Template**

The home.html file located in the sports/templates/ directory is an HTML file that defines the structure and design of the web page

Set up the template directory and create `home.html` in `sports/templates/`:

<!DOCTYPE html>

<html>

<head>

<title>Sports Website</title>

<style>

body {

background-image: url('/static/images/background.jpg');

background-size: cover;

color: white;

text-align: center;

}

</style>

</head>

<body>

<h1>Welcome to the Sports Website</h1>

{% for article in articles %}

<h2>{{ article.title }}</h2>

<p>{{ article.content }}</p>

<p><em>{{ article.published\_date }}</em></p>

{% endfor %}

</body>

</html>

**5.3 Set Up URLs**

Maps URL paths to view functions within the app.

Configure URL routing in `sports/urls.py`:

from django.urls import path

from . import views

urlpatterns = [

path('', views.home, name='home'),

path('admin/', admin.site.urls),

path('home/', views.home, name='home'),

]

**6. Static Files**

**6.1 Add Static Files**

Create a `static` directory in the project root and add static files such as images or CSS files.

**6.2 Configure Static Files in `settings.py`**

Ensure the `STATIC\_URL` setting is configured correctly in `sports\_website/settings.py`:

STATIC\_URL = '/static/'

**7. Run and Test Your Website**

**7.1 Run the Development Server**

Start the Django development server:

python manage.py runserver

**7.2 Access the Website**

Open a browser and visit `http://127.0.0.1:8000/` to view the website.

**7.3 Admin Interface**

Access the admin interface at `http://127.0.0.1:8000/admin/` to manage `Article` objects.