Web Development using Django Framework

Django is a high-level Python web framework that follows the model-view-template (MVT) architectural. It provides a solid foundation for building robust and scalable web applications quickly and efficiently. Django promotes the concept of DRY (Don't Repeat Yourself) and emphasizes code reusability, modularity, and the principle of convention over configuration.

Model: The model component in Django corresponds to the data structure and business logic. It defines the database schema and provides an abstraction layer for working with data. Models interact with the database through an ORM (Object-Relational Mapper) and encapsulate the logic for data retrieval, manipulation, and validation.

View: In Django, the view component is responsible for handling HTTP requests and returning responses. It defines the logic of your application and acts as a bridge between the model and the template. Views process the input data, interact with the model to fetch or update data, and determine the appropriate template to render the response.

Template: The template component in Django represents the presentation logic. Templates are responsible for generating the final output, typically HTML pages, based on the data provided by the view. They provide a way to separate the presentation from the application logic. Django's template engine supports variables, filters, tags, and template inheritance, allowing for modular and reusable templates.

Prerequisite

To effectively learn Django, it is beneficial to have some prior knowledge and skills in the following areas:

1) Basic Python / C / C++

2) HTML / CSS / JavaScript

3) Database

4) Command Line Basics

Completing a Django course will provide you with a comprehensive understanding of the framework and enable you to build robust web applications.

Tentative Lecture Plan:

Topics Covered in the Course	**Week**	**Topics**
	1	Introduction to Python Syntax and List, Python Tuples and Sets, Python Dictionary, OOPS
	2	Scraping (Selenium & BeautifulSoup) and Pandas
	3	Django Routing, Data Rendering, Models & Data Passing Through URLs, Frontend & Backend Integration, Template Inheritance, Django Auth
	4	Static File Serving, Media File Serving, Extended Models, Email Configuration, Account Activation, Password Reset, Signals, Django Forms

<table>
<tr><td></td><td><table>
<tr><td>5</td><td>Django Rest Framework (DRF), Serialization (Data Parsing JSON), View Sets, API View, Routers</td></tr>
<tr><td>6</td><td>API Testing, JSON Web Token, DRF Auth</td></tr>
<tr><td>7</td><td>Django Schedulers & GIT, Bitbucket, Django Channels</td></tr>
<tr><td>8</td><td>Modify Admin Panel, Integration of Third- Party APIs, AWS, Deployment</td></tr>
</table></td></tr>
<tr><td><b>Hours Per Week</b></td><td>3.30 + 3.30 = 7hrs</td></tr>
<tr><td><b>Course Duration</b></td><td>2-Months (August-October)</td></tr>
<tr><td><b>Days & Timing</b></td><td>Monday - 5:00 PM to 8:30 PM
Tuesday - 5:00 PM to 8:30 PM</td></tr>
<tr><td><b>Activities</b></td><td>Minimum-4</td></tr>
<tr><td><b>Tasks</b></td><td>Minimum-4</td></tr>
</table>