**MODULE: 8**

**PYTHON: Python DB and Framework**

* **HTML in Python:-**

1. **Write a Python program to render an HTML file using Django’s template system.**
2. **Write a Django project that renders an HTML file displaying “Welcome to Doctor Finder” on the home page.**

* **CSS in Python:-**

1. **Create a CSS file to style a basic HTML template in Django.**
2. **Write a Django project to display a webpage with custom CSS styling for a doctor profile page.**

* **JavaScript with Python:-**

1. **Create a Django project with JavaScript-enabled form validation.**
2. **Write a Django project where JavaScript is used to validate a patient registration form on the client side.**

* **Django Introduction:-**

1. **Write a short project using Django’s built-in tools to render a simple webpage.**
2. **Write a Python program to create a Django project and understand its directory structure.**

* **Virtual Environment:-**

1. **Set up a virtual environment for a Django project.**
2. **Write a Python program to create and activate a virtual environment, then install Django in it.**

* **Project and App Creation:-**

1. **Create a Django project with an app to manage doctor profiles.**
2. **Write a Python program to create a Django project and a new app with in the project called doctor.**

* **MVT Pattern Architecture:-**

1. **Build a simple Django app showcasing how the MVT architecture works.**
2. **Write a Django project with models, views, and templates to display doctor information.**

* **Django Admin Panel:-**

1. **Set up and customize the Django admin panel to manage a "Doctor Finder" project.**
2. **Write a Django project to create an admin panel and add custom fields for managing doctor information.**

* **URL Patterns and Template Integration:-**

1. **Create a Django project with URL patterns and corresponding views and templates.**
2. **Write a Django project where URL routing is used to navigate between different pages of a “Doctor Finder” site (home, profile, contact).**

* **Form Validation using JavaScript:-**

1. **Write a Django project to implement JavaScript form validation for a user registration form.**
2. **Write a Django project that uses JavaScript to validate fields like email and phone number in a registration form.**

* **Django Database Connectivity (MySQL or SQLite):-**

1. **Set up database connectivity for a Django project.**
2. **Write a Django project to connect to an SQLite/MySQL database and manage doctor records.**

* **ORM and Query Sets:-**

1. **Perform CRUD operations using Django ORM.**
2. **Write a Django project that demonstrates CRUD operations (Create, Read, Update, Delete) on doctor profiles using Django ORM.**

* **Django Forms and Authentication:-**

1. **Create a Django project for user registration and login functionality.**
2. **Write a Django project to handle user sign up, login, password reset, and profile updates.**

* **CRUD Operations using AJAX:-**

1. **Implement AJAX in a Django project for performing CRUD operations.**
2. **Write a Django project that uses AJAX to add, edit, or delete doctor profiles without refreshing the page.**

* **Customizing the Django Admin Panel:-**

1. **Customize the Django admin panel for better management of records.**
2. **Write a Django project that customizes the admin panel to display more detailed doctor information (e.g., specialties, availability).**

* **Payment Integration Using Paytm:-**

1. **Implement Paytm payment gateway in a Django project.**
2. **Write a Django project that integrates Paytm for handling payments in the "Doctor Finder" project.**

* **GitHub Project Deployment:-**

1. **Deploy a Django project to GitHub for version control.**
2. **Write a step-by-step guide to deploying the “Doctor Finder” project to GitHub.**

* **Live Project Deployment (PythonAnywhere):-**

1. **Deploy a Django project to PythonAnywhere.**
2. **Write a Django project and deploy it on PythonAnywhere, making it accessible online.**

* **Social Authentication:-**

1. **Implement Google and Facebook login for the Django project.**
2. **Write a Django project to allow users to log in using Google or Facebook.**

* **Google Maps API:-**

1. **Use Google Maps API to display doctor locations in the "Doctor Finder" project.**
2. **Write a Django project to display doctor locations using Google Maps API.**