3. JavaScript with Python

## **Using JavaScript for client-side interactivity in Django templates.**

**Theory:**  
JavaScript is widely used in Django projects to add interactivity and enhance the user experience on web pages. It allows developers to handle client-side actions such as button clicks, form validation, and dynamic page updates without reloading the page.

**Using JavaScript for Client-Side Interactivity in Django Templates**  
JavaScript helps make Django templates interactive. For example, you can use JavaScript to validate user inputs, show alerts, or dynamically change HTML content.  
In a Django template, JavaScript code can be added directly inside <script> tags or linked externally.

**Example:**

<script>

function greetUser() {

alert("Welcome to Doctor Finder!");

}

</script>

<button onclick="greetUser()">Click Me</button>

This function runs on the client side without needing a server reload, improving the overall user experience.

## **Linking external or internal JavaScript files in Django.**

**Linking External or Internal JavaScript Files in Django**  
Django allows linking JavaScript files either internally (inside templates) or externally (as separate .js files in the static folder).

**Steps for Linking External JavaScript Files:**

1. Create a folder named static/js/ in your Django app.
2. Save your JavaScript file inside that folder (e.g., script.js).
3. In your HTML template, load and link it using Django’s static tag:

{% load static %}

<script src="{% static 'js/script.js' %}"></script>

This setup helps in managing all scripts centrally and keeping templates clean.