Tops Technology

Module 16)
Python DB and Framework

Presented By : Nandni Vala

JavaScript with Python

1.Using JavaScript for client-side interactivity in Django templates.

- To use JavaScript for client-side interactivity in Django templates:
- **Include JavaScript Files**: Add `<script>` tags in your Django templates to include your JavaScript files. Use `{% static %}` for referencing static files, e.g.,
- <script src="{% static 'ja\script.js' %}</script>"
- **Pass Data from Django to JavaScript**: Use template variables to pass data, e.g.,
- <script>
-     const userData = {{ user_data|safe }};
- </script>
- **AJAX for Dynamic Updates**: Use JavaScript (e.g., Fetch API or libraries like Axios) to send and receive data from Django views without reloading the page.

- **DOM Manipulation**: Use JavaScript to manipulate the DOM for interactivity.
- **Use CSRF Tokens**: Include the CSRF token in AJAX requests for POST methods, e.g.,
- const csrftoken = document. Query Selector ('[name=csrfmiddlewaretoken]') . value;

2.Linking external or internal JavaScript files in Django.

To link external or internal JavaScript files in Django:

1. **For Internal JavaScript Files:**

- ➢ Save the JavaScript file in the static directory of your app (e.g., static/js/script.js).
- ➢ Load the static files in your template with {% load static %}.
- ➢ Link the file in your template using the <script> tag:

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

2. **For External JavaScript Files:**

- ➢ Directly include the external file's URL in the <script>
- ➢ <script src="https://example.com/external-script.js"></script>

➢Best Practices:

➢ Place the `<script>` tags at the end of the `<body>` section for better performance.

➢ Use the `defer` or `async` attributes to optimize loading:

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