**Technical Test: Front End Developer**

1. Tolong jelaskan tentang Shadow DOM !

*Jawaban :*

Shadow DOM itu seperti membungkus elemen-elemen di dalam sebuah kotak yang tertutup rapat. Kotak ini melindungi elemen-elemen di dalamnya dari pengaruh atau campur tangan elemen-elemen dari luar kotak tersebut. Jadi, elemen-elemen di dalam kotak ini punya aturan sendiri yang tidak akan mengubah elemen-elemen di luar kotak.

Misalnya, kita punya sebuah tombol di dalam kotak ini. Gaya /Style(warna, bentuk, dll.) dan perilaku tombol itu hanya berlaku di dalam kotak, tidak mempengaruhi tombol-tombol di luar kotak. Ini membantu mencegah permasalahan di antara elemen-elemen yang berbeda di halaman web kita.

Jadi, Shadow DOM membantu menjaga elemen-elemen agar tidak saling bermasalah atau berantakan dan membuat kode web kita lebih rapi dan terkelola.

1. Apa itu SCSS Selector? Apa kelebihannya?

*Jawaban :*

SCSS Selector adalah bagian dari bahasa pemrograman CSS yang disebut SCSS (Sassy CSS). SCSS sendiri adalah ekstensi dari CSS yang memperkenalkan beberapa fitur baru dan lebih mudah ditulis. Selector dalam SCSS mengacu pada cara kita memilih elemen atau kelompok elemen di HTML yang akan diberi gaya.

Keuntungan SCSS Selector :

1. Nesting (Penggolongan Bertingkat): Dengan SCSS Selector, kita dapat menyusun gaya dengan cara yang lebih terstruktur dan terbaca. Contohnya, aturan gaya untuk elemen-elemen yang berada di dalam suatu kontainer dapat ditempatkan bersamaan di dalam blok yang sama.
2. Kode Lebih Rapi: SCSS memungkinkan kita untuk menulis kode gaya dengan lebih sedikit repetisi. Kita dapat menggunakan variabel, fungsi, dan mixin untuk menghindari penulisan ulang yang berlebihan.
3. Mudah Dimengerti: SCSS Selector membuat kode lebih intuitif dan mudah dimengerti karena strukturnya mirip dengan struktur HTML yang sesuai dengan hierarki elemen
4. Jelaskan Lifecycle hook pada Angular!

*Jawaban :*

1. ngOnChanges : Dipanggil setiap kali nilai input dari komponen berubah.
2. ngOnInit : Dipanggil setelah Angular menginisialisasi komponen dan mengatur properti input pertamanya.
3. ngDoCheck : Dipanggil saat deteksi perubahan dilakukan di seluruh aplikasi.
4. ngAfterContentInit: Dipanggil setelah Angular memasukkan konten eksternal ke dalam tampilan komponen.
5. ngAfterContentChecked: Dipanggil setiap kali deteksi perubahan dilakukan pada konten eksternal.
6. ngAfterViewInit: Dipanggil setelah Angular menginisialisasi tampilan dan komponen anak (jika ada).
7. ngAfterViewChecked: Dipanggil setiap kali deteksi perubahan dilakukan pada tampilan dan komponen anak.
8. ngOnDestroy: Dipanggil sebelum Angular menghancurkan komponen.