**Que 1.Reactive and template driven form ?**

**Template Driven Forms Features**

* Easy to use
* Suitable for simple scenarios and fails for complex scenarios
* Similar to AngularJS
* Two way data binding(using [(NgModel)] syntax)
* Minimal component
* Automatic track of the form and its data(handled by Angular)
* Unit testing is another challenge

**Reactive Forms Features**

* More flexible, but needs a lot of practice
* Handles any complex scenarios
* No data binding is done (immutable data model preferred by most developers)
* More component code and less HTML markup
* Reactive transformations can be made possible such as
  + Handling a event based on a debounce time
  + Handling events when the components are distinct until changed
  + Adding elements dynamically
* Easier unit testing

A side by side pros and cons

Que 2. Dynamic Routing

this.id = this.activatedRoute.snapshot.paramMap.get('id');

console.log(this.id);

Que 3. Dom sanitization?

import { DomSanitizer, SafeResourceUrl, SafeUrl } from '@angular/platform-browser';

**define variable**-

trustedDashboardUrl : SafeUrl;

**Use it in constructor like this-**

constructor(

private sanitizer: DomSanitizer) {}

**Specifiy URL like this-**

this.trustedDashboardUrl =

this.sanitizer.bypassSecurityTrustResourceUrl

("URL");

See if this helps.

Que 6. Remove # tag from URL

You need to change your LocationStrategy to PathLocationStrategy

By default angular use the PathLocationStrategy, that said looks like you're probably defining HashLocationStrategy yourself. Look for some sort of this code in your project:

RouterModule.forRoot(routes, { useHash: true })

**In this case just remove the useHash like so:**

RouterModule.forRoot(routes)

**Or maybe**

[Location, {provide: LocationStrategy, useClass: HashLocationStrategy}]

**In this case change HashLocationStrategy to PathLocationStrategy like so:**

[Location, {provide: LocationStrategy, useClass: PathLocationStrategy}]