Building blocks of Angular.

The Angular application is made using the following:

Modules, Component, Template, Directives, Data Binding, Services, Dependency Injection, Routing

Transpiling: is the process of converting the typescript into javascript

ngOnChanges is the life cycle hook that gets executed whenever a change happens to the data that was bound to an input.

Components break up the application into smaller parts; whereas, Directives add behavior to an existing DOM element.

Module: in Angular refers to a place where you can group the components, directives, pipes, and services, which are related to the application.

@Input and @Output? communication of Angular Components, which are in Parent-Child Relationship; we use @Input in Child Component when we are passing data from Parent to Child Component and @Output is used in Child Component to receive an event from Child to Parent Component.

router.navigate ?

When we want to route to a component we use router.navigate.

Routing helps a user in navigating to different pages using links

this.router.navigate(['/component_name']);

ViewEncapsulation decides whether the styles defined in a component can affect the entire application or not.

Services help us in not repeating the code. With the creation of services, we can use the same code from different components.

Dependency Injection

When a component is dependent on another component the dependency is injected/provided during runtime

RouterOutlet is a substitution for templates rendering the components.

Data Binding

(i) Property Binding (ii) Event Binding (iii) Two-Way Data Binding.

Angular Lifecycle Hooks?

OnChange() - OnInit() - DoCheck() - AfterContentInit() - AfterContentChecked() - AfterViewInit() - AfterViewChecked() - OnDestroy().

package.json : It will be easy to manage the dependencies of the project

ngModel is a directive which can be applied on a text field. This a two-way data binding. ngModel is represented by [()] ---> [(ngModel)] ="myMsg"

Subscribe It is a method which is subscribed to an observable. Whenever the subscribe method is called, an independent execution of the observable happens

Observables VS Promises.

Observables are lazy, which means nothing happens until a subscription is made. Whereas Promises are eager; which means as soon as a promise is created, the execution takes place. Observable is a stream in which passing of zero or more events is possible and the callback is called for each event. Whereas, promise handles a single event.

AOT Compilation?

Ahead-of-Time compiler pre-compiles application components and their templates during the build process

Pipes: This feature is used to change the output on the template

In ng-Class, loading of CSS class is possible; whereas, in ng-Style we can set the CSS style.

forRoot creates a module that contains all the directives, the given routes, and the router service itself.

forChild creates a module that contains all the directives and the given routes, but does not include the router service. It registers the routers and uses the router service created at the root level.

Angular Universal: server-side rendering : A normal Angular application executes in the browser, rendering pages in the DOM in response to user actions. Angular Universal generates static application pages on the server through a process called server-side rendering (SSR). When Universal is integrated with your app, it can generate and serve those pages in response to requests from browsers. It can also pre-generate pages as HTML files that you serve later.

RxJS operator debounceTime() on a form control's valueChanges observable

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 code
- Automatic track of the form and its data(handled by Angular)
- Unit testing is another challenge

Reactive Forms/ Model-Driven 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

Angular7 features

Smaller and Faster, View Engine, Animation Package, Improved *ngIf and *ngFor

Application performance, Virtual Scrolling [ScrollingModule], Drag and Drop