ANGULARJS LAB VIVA QUESTION BANK WITH ANSWERS

1. Knowing the Editor

Q: **Q:** What is AngularJS?

Ans: AngularJS is a JavaScript-based front-end framework developed by Google to create dynamic, single-page web applications (SPAs).

Q: **Q:** What are the main features of AngularJS?

Ans: Data Binding, Dependency Injection, Directives, MVC Architecture, Routing, and Filters.

Q: **Q:** What are directives in AngularJS?

Ans: Directives are special HTML attributes that extend HTML functionality, e.g., ngModel, ngFor, ngIf.

Q: **Q:** What is data binding?

Ans: Data binding synchronizes data between the model (TypeScript code) and the view (HTML template).

Q: **Q:** Difference between AngularJS and Angular?

Ans: AngularJS (1.x) is JavaScript-based, while Angular (2+) uses TypeScript and a component-based architecture.

2. Implementing Components

Q: **Q:** What is a component?

Ans: A component controls a portion of the screen called a view; it includes a .ts, .html, and .css file.

Q: **Q:** How to create a component?

Ans: Using Angular CLI: ng generate component componentName

Q: **Q:** What is @Component?

Ans: It defines a class as a component and provides configuration metadata like selector, template, and style.

Q: **Q:** Difference between Component and Directive?

Ans: Components have templates; directives don’t.

Q: **Q:** What is encapsulation in components?

Ans: It isolates component styles and logic.

3. Implementing Templates

Q: **Q:** What is a template?

Ans: The HTML view associated with a component.

Q: **Q:** What is interpolation?

Ans: Using {{ }} to display component data in HTML.

Q: **Q:** What is property binding?

Ans: Binding component properties to HTML elements using [property]=value.

Q: **Q:** What is event binding?

Ans: Capturing events from the DOM using (event)=function().

Q: **Q:** Difference between *ngIf and *ngFor?

Ans: *ngIf conditionally displays elements; *ngFor iterates lists.

4. Creating Routing Applications

Q: **Q:** What is routing?

Ans: It enables navigation between different components without reloading the page.

Q: **Q:** Why use routing?

Ans: To create single-page applications (SPAs).

Q: **Q:** What is RouterModule?

Ans: A module that provides routing services and directives.

Q: **Q:** What is ?

Ans: A placeholder in the template where routed components are displayed.

Q: **Q:** What are route guards?

Ans: They protect routes (e.g., authentication).

5. Displaying a List

Q: **Q:** What is *ngFor?

Ans: A structural directive used to loop through arrays in templates.

Q: **Q:** What is trackBy?

Ans: Optimizes rendering by tracking items via unique identifiers.

Q: **Q:** What are pipes?

Ans: Transform data in the template, e.g., {{ name | uppercase }}.

Q: **Q:** How to filter or sort lists?

Ans: Using pipes or component logic.

Q: **Q:** How to show message if list empty?

Ans: Using *ngIf.

6. Adding Services

Q: **Q:** What is a service?

Ans: A reusable class for business logic or data management.

Q: **Q:** Why use services?

Ans: To share data and logic across components.

Q: **Q:** What is dependency injection?

Ans: The process of providing required objects (services) to components.

Q: **Q:** What is a singleton service?

Ans: A service with only one instance used across the app.

Q: **Q:** What is providedIn: 'root'?

Ans: Makes the service available globally.

7. Adding Navigation

Q: **Q:** How to navigate between pages?

Ans: Using router links or navigation methods.

Q: **Q:** What is used for?

Ans: Defines clickable links for navigation.

Q: **Q:** What is ?

Ans: Placeholder where routed component gets loaded.

Q: **Q:** What are route parameters?

Ans: Variables in the route path (e.g., /user/:id).

Q: **Q:** What are child routes?

Ans: Routes nested under another route.

8. Getting Data from a Server

Q: **Q:** How to get data from a server?

Ans: Using HttpClient service to make HTTP requests.

Q: **Q:** What is HttpClientModule?

Ans: A module providing HTTP communication features.

Q: **Q:** How to make an HTTP GET request?

Ans: this.http.get('url').subscribe(data => { });

Q: **Q:** What are Observables?

Ans: Streams of data that can be observed asynchronously.

Q: **Q:** How to handle errors?

Ans: Using catchError from RxJS.

9. Using Published Libraries

Q: **Q:** What are published libraries?

Ans: Ready-made packages that extend app functionality.

Q: **Q:** How to install a library?

Ans: npm install library-name

Q: **Q:** Examples of libraries

Ans: Angular Material, Bootstrap, Chart.js

Q: **Q:** What is npm?

Ans: Node Package Manager for managing dependencies.

Q: **Q:** What are dependencies and devDependencies?

Ans: Dependencies: required at runtime; devDependencies: needed only during development.

10. Creating User Defined Libraries

Q: **Q:** What is a user-defined library?

Ans: A custom reusable module developed by the user.

Q: **Q:** Why create one?

Ans: For modularity and reuse across multiple projects.

Q: **Q:** Command to create a library

Ans: ng generate library library-name

Q: **Q:** How to build it?

Ans: ng build library-name

Q: **Q:** Can we publish to npm?

Ans: Yes, using npm publish.