AngularJS is a popular open-source framework that simplifies web development by creating interactive single-page applications (SPAs). Unlike traditional websites that load new pages for each click, SPAs offer a smoother user experience by updating content on the same page. AngularJS makes this possible by transforming static HTML into dynamic content that adapts to user interactions. Features like data binding and dependency injection streamline development, saving you time and effort. With regular updates and a large community, AngularJS ensures your web applications stay modern and efficient. AngularJS is a JavaScript Framework

1

Differences AngularJS & Angular:
Language:AngularJS: Uses JavaScript. **Angular**: Uses TypeScript, a superset of JavaScript that offers static typing and object-oriented features, leading to more robust and maintainable code. **Architecture: AngularJS**: Employs the Model-View-Controller (MVC) or Model-View-ViewModel (MVVM) architectural pattern. **Angular**: Utilizes a component-based architecture, where applications are built from self-contained components. **Performance**:**Angular**: Generally faster due to its optimized architecture, one-way data flow, and features like lazy loading and server-side rendering (Angular Universal). **AngularJS**: Can experience performance limitations, especially in larger applications, due to its two-way data binding and digest cycle. **Data Binding**: **AngularJS**: Primarily uses ng-model for two-way binding and ng-bind for one-way binding, relying on the ng directive approach. **Angular**: Uses []
for property binding and () for event binding, simplifying data flow.

1

Creating Project You develop appsin the context of an Angular workspace. To create a new workspace and initial starter app: 1. Run the CLI command ng new and provide the name my-app, as shown here: ng new my-app 2. The ng new command prompts you for information about featuresto include in the initial app. Accept the defaults by pressing the Enter or Return key. The Angular CLI installsthe necessary Angular npm packages and other dependencies. This can take a few minutes.The CLI creates a new workspace and a simple Welcome app, ready to run.

1

MVC Framework The Model-View-Controller is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects. MVC Components: Following are the components of MVC: **Model**: The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data. **View**: The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with. **Controller**: Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

1

The Angular **CLI** is a command-line interface tool which allows you to scaffold, develop, test, deploy, and maintain Angular applications directly from a command shell. Angular CLI is published on npm as the @angular/cli package and includes a binary named ng.

In AngularJS, a **component** is a special type of directive that is used to create reusable UI elements with their own logic, template, and behavior.A component in AngularJS:
1 Is a simplified version of a directive. 2 Encapsulates template (HTML), controller (logic), and bindings (data inputs/outputs). 3 Promotes reusable, modular, and testable code. There is no fixed number of components — you can create as many as your application needs. Each UI section or feature can be built as a separate component for modularity. **Module** – Defines an application.Component – Reusable, self-contained UI block. **Controller** – Handles logic (used more in older AngularJS before components). **Directive** – Extends HTML behavior. **Service/Factory** – Provides reusable business logic or data. **Filter** – Formats data for display. **Routing** – Defines navigation between views.

1

install Node and NPM on windows
Step 1: Download Node.js
1 Go to the official Node.js website: "https://nodejs.org". 2 You'll see two versions: LTS (Long Term Support) — recommended for most users.Current — includes the latest features but may be less stable. For AngularJS or Angular projects, choose LTS. 3 Click Windows Installer (.msi) to download.
Step 2: Install Node.js and npm
1 Run the installer you downloaded. 2 Follow the setup wizard:-> Accept the license agreement. -> Choose the default installation path. -> Leave the default options checked (especially the box that says "Install npm package manager"). -> Optional: Check "Automatically install necessary tools" — this installs Python and Visual Studio Build Tools (useful for some npm packages). 3 Click Install and wait for it to finish. 4 When done, click Finish.
Step 3: Verify Installation
Open Command Prompt (cmd) or PowerShell, then run:node -v, Then check npm:npm -v
Step 4: Using Node for AngularJS
You don't need the Angular CLI. You just use Node and npm to manage dependencies or run a local server. Example: npm init -y, npm install http-server --save-dev, Then you can start a simple local server: npx http-server.

1

create your first Angular Project.
Step 1: Install Prerequisites
You'll need: A web browser (Chrome, Firefox, etc.). A text editor (VS Code, Sublime Text, etc.). Node.js and npm (optional) if you want to use a local web server.
Step 2: Create a Project Folder
Create a new folder for your project, for example: C:\Projects\MyFirstAngularJSApp
Step 3: Create Basic Files : index.html,app.js
Step 4: Add AngularJS Library
You can include AngularJS from a CDN.
In index.html, add:

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
 <meta charset="utf-8">
 <title>My First AngularJS App</title>
 <!-- Load AngularJS -->
 <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
 <script src="app.js"></script>
</head>
<body>
 <div ng-controller="myController">
  <h1>{{ message }}</h1>
 </div>
</body>
</html>
```

Step 5: Create the AngularJS Application
In app.js, define your AngularJS module and controller:

```
var app = angular.module("myApp", []);
app.controller("myController",
function($scope) {
  $scope.message = "Hello, AngularJS!";
});
```

Step 6: Run the Application
Option 1 — Open directly in browser:

Just double-click index.html — AngularJS will run locally.

Option 2 — Run via local server (recommended):
If you have Node.js, run a local server: npx http-server.

Step 7: Verify Output
Hello, AngularJS!

1

Features of Angular:
Build app-like experiences using modern web platform capabilities. High performance, offline, and zero-step installation. You can develop Cordova, Ionic, or NativeScript mobile apps with these strategies. With the addition of Angular to Mac, Windows, and Linux, you can create desktop-installed apps using the same methods as for the web. Furthermore, you can use native OS APIs on Windows and Linux. Due to Angular's advanced JavaScript virtual machine optimizations, hand-written code with the productivity of a framework is achieved without the problems of frameworks. The first view of your application can be served on Node.js, .NET, PHP, and other servers nearinstantaneously in HTML and CSS by also laying the groundwork for SEO-optimized sites. When using the new Component Router, which automatically splits the Angular code into code that is needed for rendering only the part of the page that is requested, users only need to load code necessary for the page to be displayed. UI views can be created quickly with template syntax that issimple and powerful. Quickly build modules and tests, then instantly deploy them using Command Line Tools. In addition to intelligent code completion, instant errors, and other feedback, popular editors and IDEs offer this capability. You can track whether or not you've violated any rules by checking Karma for unit tests. An Angular tutorial using very little code to build complex choreography and animation timelines. Use ARIA-compliant components, developer guides, and built-in a11y test infrastructure to create
accessible applications

1

Running Project The Angular CLI includes a server, for you to build and serve your app locally. 1. Navigate to the workspace folder,such as my-app. 2. Run the following command: cd my-app ng serve --open

1

Files used in Angular App folder: Angular App files which are mainly used in your project are given below: src folder: This is the folder which contains the main code filesrelated to your angular application. app folder: The app folder contains the files, you have created for app components. app.component.css: This file containsthe cascading style sheets code for your app component. app.component.html: This file contains the html file related to app component. This is the template file which is used by angular to do the data binding. app.component.spec.ts: This file is a unit testing file related to app component. This file is used along with other unit tests. It is run from Angular CLI by the command ng test. app.component.ts: This is the most important typescript file which includes the view logic behind the component. app.module.ts: This is also a typescript file which includes all the dependencies for the website. This file is used to define the needed modules to be imported, the components to be declared and the main component to be bootstrapped.

1