**MEAN Stack**

Mean stack stands for Mongo DB, Express JS, Angular, and Node JS. It is a set of these technologies that are used to develop each end of a web application. Further details about these technologies are given as follows:

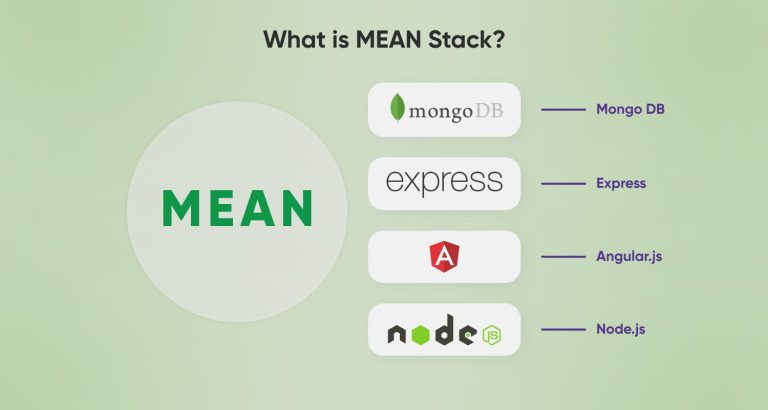
* [Mongo DB](https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/)**:**Its a database which is using BSON format to store data.
* [Express JS](https://www.geeksforgeeks.org/express-js/)**:**Its a framework of JavaScript which enables developer to establish server.
* [Angular](https://www.geeksforgeeks.org/angularjs/) **:**Angular is client side framework which is used to develop User Interface.
* [**Node JS**](https://www.geeksforgeeks.org/nodejs/)**:**Node JS is a run time environment of JavaScript which allows developer to execute JS code directly in console

**Working of MEAN Stack:**

Angular handles the implementation of the user interface (UI). Whenever the UI requires data, it sends a request to the server, prompting the server to retrieve the necessary information from MongoDB. Once the server successfully locates the required data, it sends the response back to the client side.

**Advantages of MEAN Stack:**

* Whole technologies requires only JavaScript to implement programs.
* Improved re-usability of code.
* Easy to learn.



**Experiment 1: ANGULAR INTRODUCTION**

**Angular**: Angular is an open-source web application framework maintained by Google and a community of developers. It is designed to build dynamic and interactive single-page applications (SPAs) efficiently. With Angular, developers can create robust, scalable, and maintainable web applications.

**History:**

Angular, initially released in 2010 by Google, has undergone significant transformations over the years. The first version, AngularJS, introduced concepts like two-way data binding and directives. However, as web development evolved, AngularJS faced limitations in terms of performance and flexibility.

**Prerequisites:**

* [**TypeScript**](https://www.geeksforgeeks.org/typescript/)
* [**HTML**](https://www.geeksforgeeks.org/html-tutorial/)
* [**CSS**](https://www.geeksforgeeks.org/css-tutorial/)
* [**JavaScript**](https://www.geeksforgeeks.org/javascript/)

**Why Angular?**

JavaScript is the most commonly used client-side scripting language. It is written into HTML documents to enable interactions with web pages in many unique ways.

**Here are some of the features of Angular**

**1. Custom Components**

Angular enables users to build their components that can pack functionality along with rendering logic into reusable pieces.

**2. Data Binding**

Angular enables users to effortlessly move data from JavaScript code to the view, and react to user events without having to write any code manually.

**3. Dependency Injection**

Angular enables users to write modular services and inject them wherever they are needed. This improves the testability and reusability of the same services.

**4. Testing**

Tests are first-class tools, and Angular has been built from the ground up with testability in mind. You will have the ability to test every part of your application—which is highly recommended.

**5. Comprehensive**

Angular is a full-fledged JavaScript framework and provides out-of-the-box solutions for server communication, routing within your application, and more.

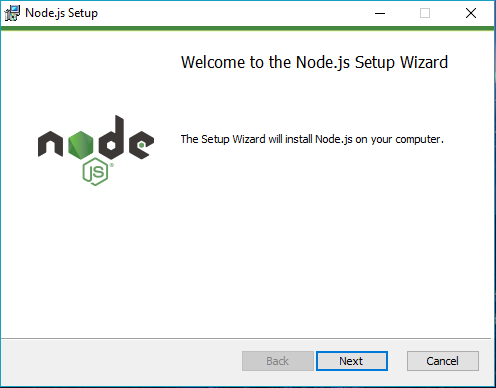
**6. Browser Compatibility**

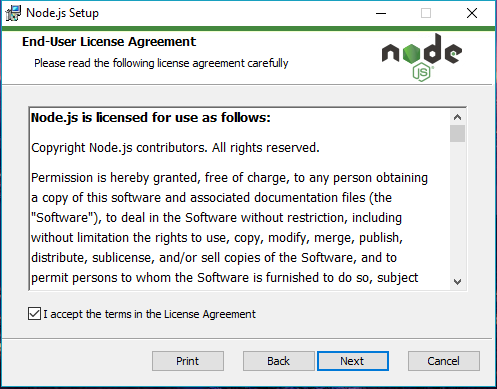
Angular works cross-platform and compatible with multiple browsers. An Angular application can typically run on all browsers (Eg: Chrome, Firefox) and operating systems, such as Windows, macOS, and Linux.

**Experiment 2: GETTING STRATED WITH ANGULAR**

**Step 1: Download the NodeJS**

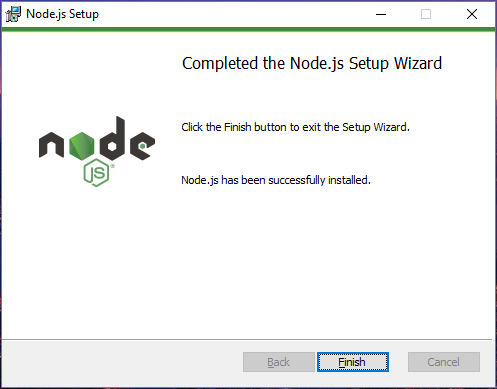
**Downloading the Node.js ‘.msi’ installer the first step to install Node.js on Windows is to download the installer. Visit the official Node.js website i.e)**[https://nodejs.org/en/download/](https://nodejs.org/en/download)

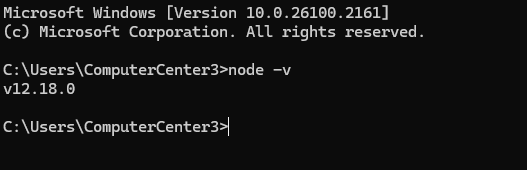




A screenshot of a computer

Description automatically generated

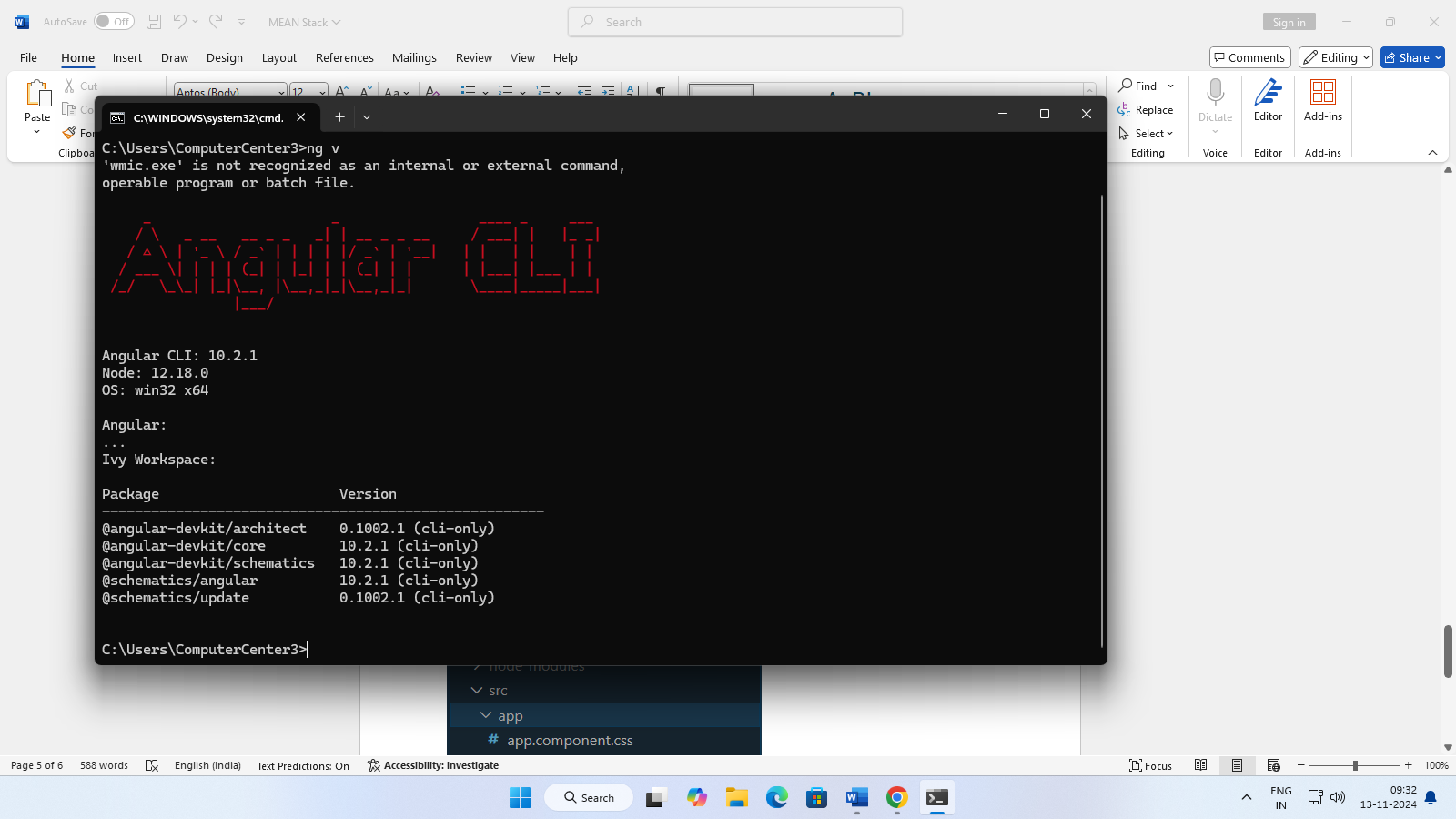


****

**Creating an Angular Application**

Step 1: Install Angular CLI: Angular CLI (Command Line Interface) is a powerful tool for scaffolding and managing Angular applications. Install it globally using npm:

**npm install -g @angular/cli**

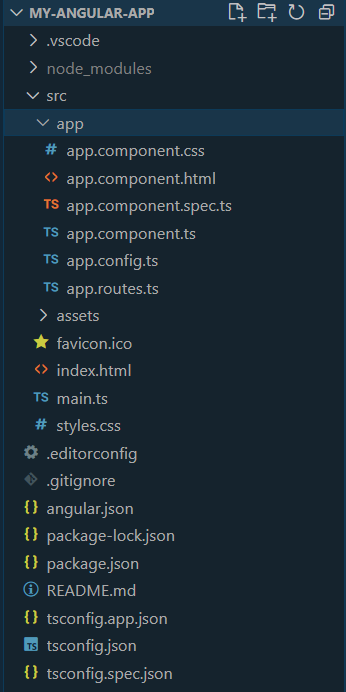


Step 2: Create a New Angular Project: Use Angular CLI to create a new Angular project. Navigate to the desired directory and run:

ng new my-angular-app

**Step 3: Navigate to the Project Directory:**Move into the newly created project directory:

**Folder Structure:**

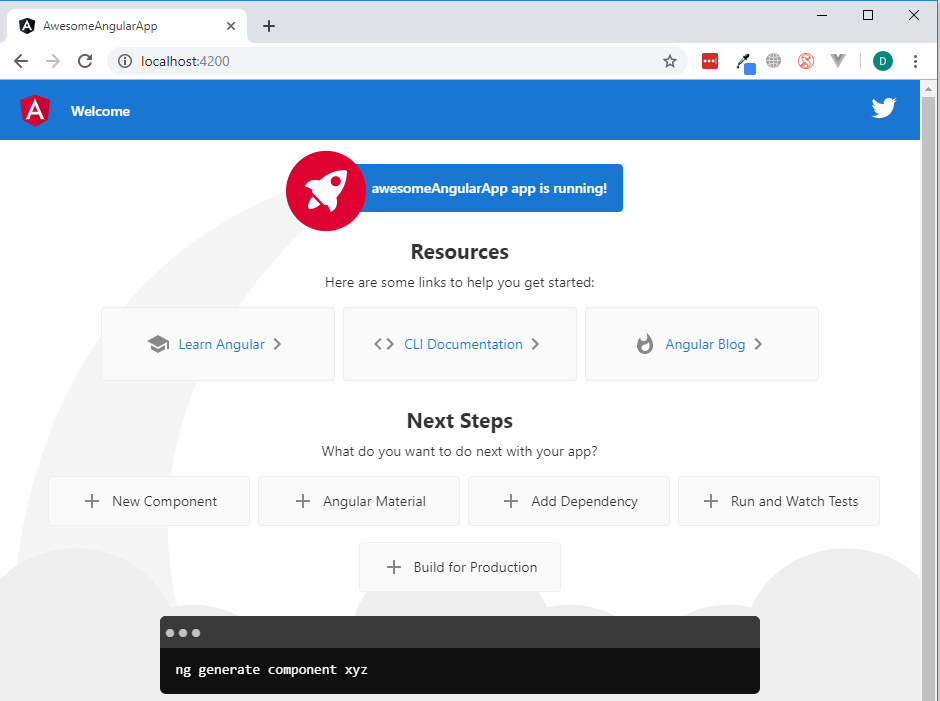


cd my-angular-app

**Step 4: Serve the Application:**Launch the development server to see your app in action:

ng serve

OUTPUT:



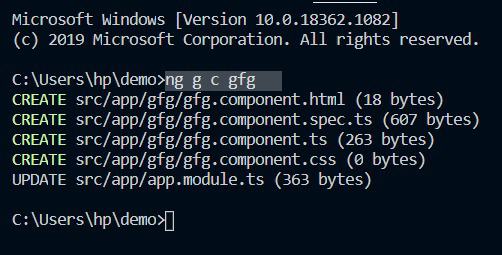
**Experiment 3: INTRODUCTION TO COMPONENTS**

Angular components are the building blocks of a UI in an Angular application. These components are associated with a template and are a subset of directives. The above image shows the classification tree structure. A root component, the App Component, branches out into other components, creating a hierarchy.

**Creating a Component in Angular 10:**

To create a component in any angular application, follow the below steps:

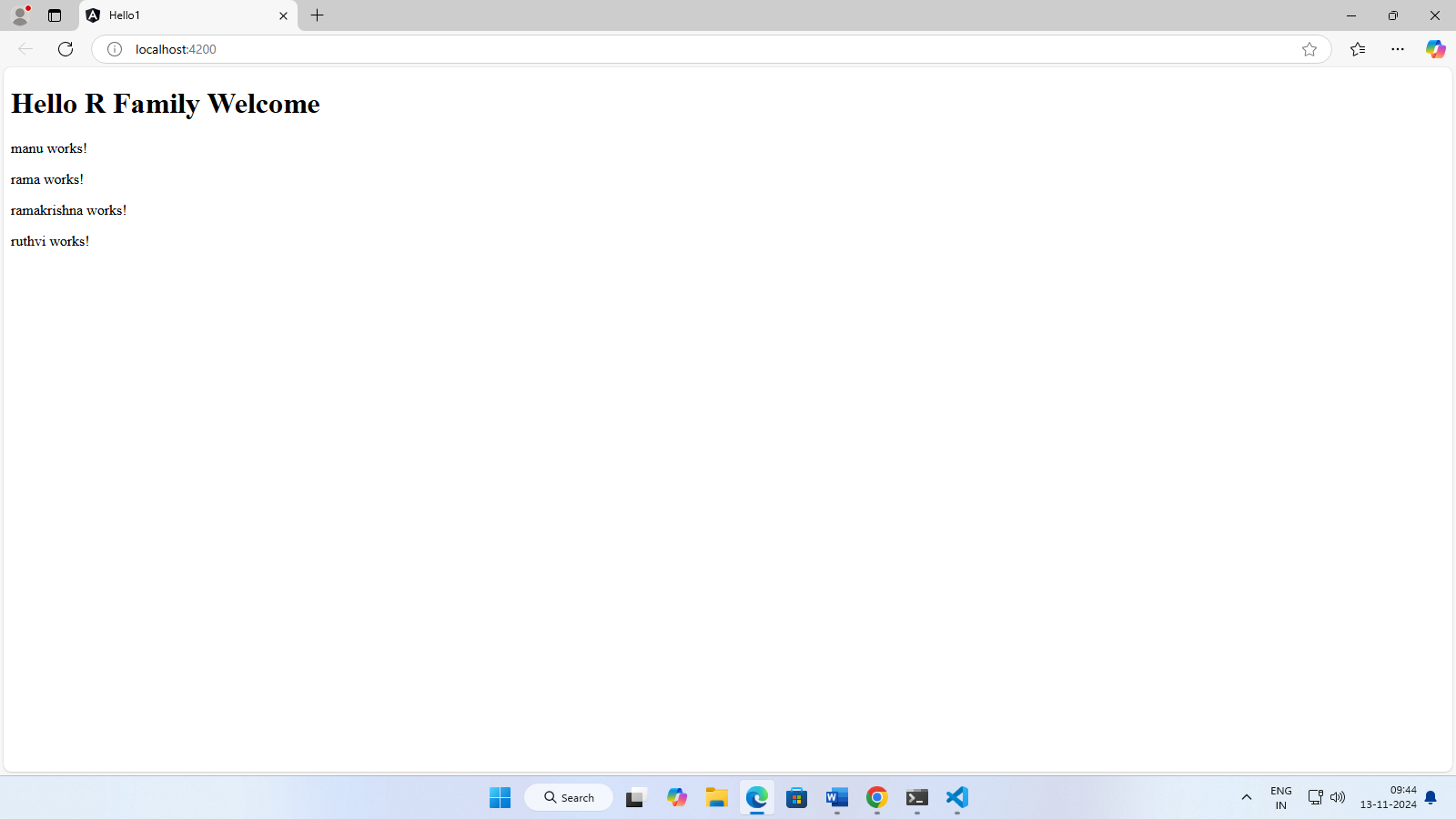
* Get to the angular app via your terminal.
* Create a component using the following command:
* ng g c <component\_name>
* OR
* ng generate component <component\_name>



Using a component in Angular :

* Go to the component.html file and write the necessary HTML code.
* Go to the component.css file and write the necessary CSS code.
* Write the corresponding code in component.ts file.
* Run the Angular app using ng serve –open

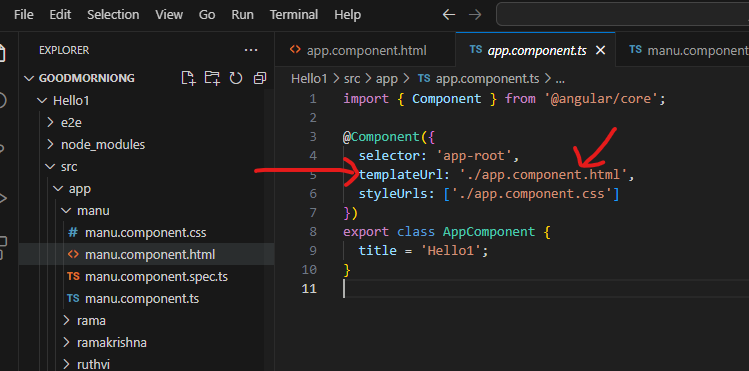
OUTPUT

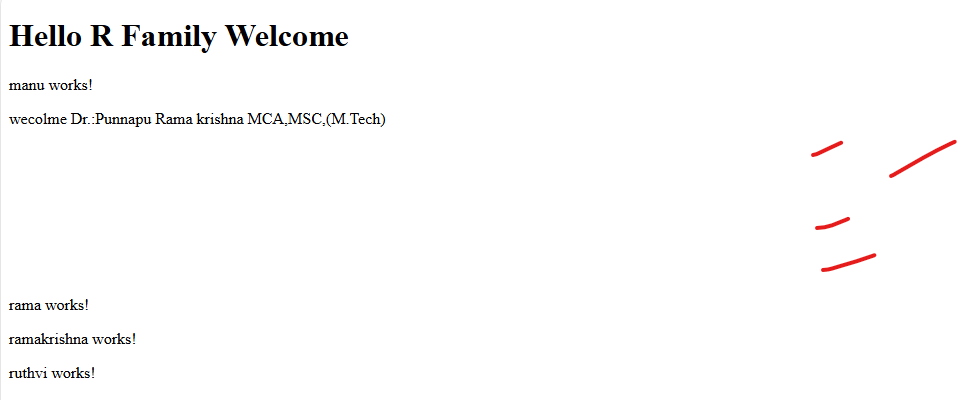


**EXPERIMENT 4 : TEMPLATES ,INTERPOLATION ,AND DIRECTIVES**

1. **TEMPLATES:** Introduction to components and templates

A *component* controls a patch of screen called a [*view*](https://v17.angular.io/guide/glossary#view). It consists of a TypeScript class, an HTML template, and a CSS style sheet. The TypeScript class defines the interaction of the HTML template and the rendered DOM structure, while the style sheet describes its appearance.



OUTPUT: 

**B)INTERPOLATION:**

 Interpolation is a way to transfer the data from a TypeScript code to an HTML template (view), i.e. it is a method by which we can put an expression in between some text and get the value of that expression. Interpolation basically binds the text with the expression value

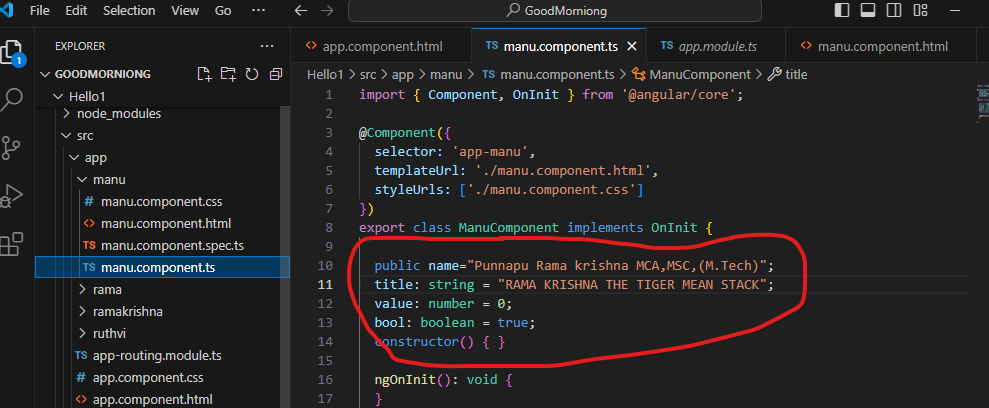
**Syntax:**

{{expression}}

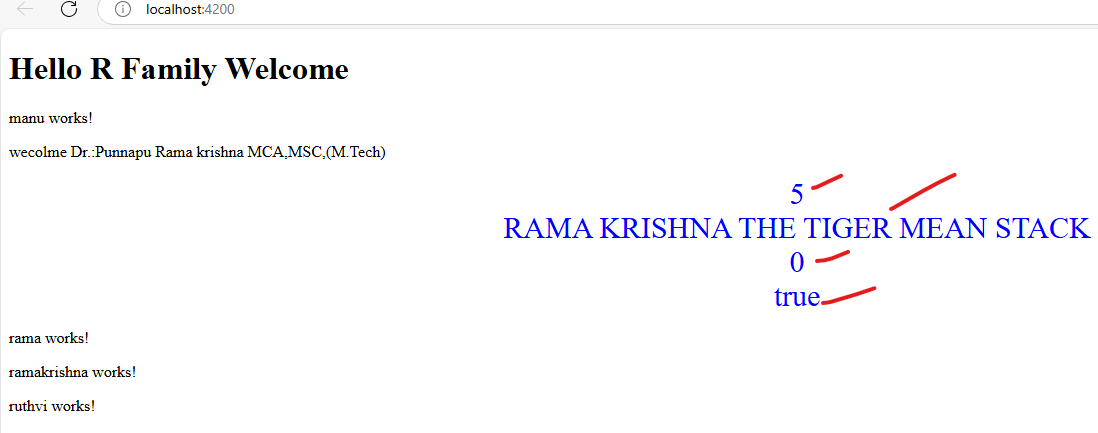
STEP1: manu.component.htmlA screen shot of a computer

Description automatically generated

Manu.component.ts



OUTPUT:



C)DIRECTIVES:

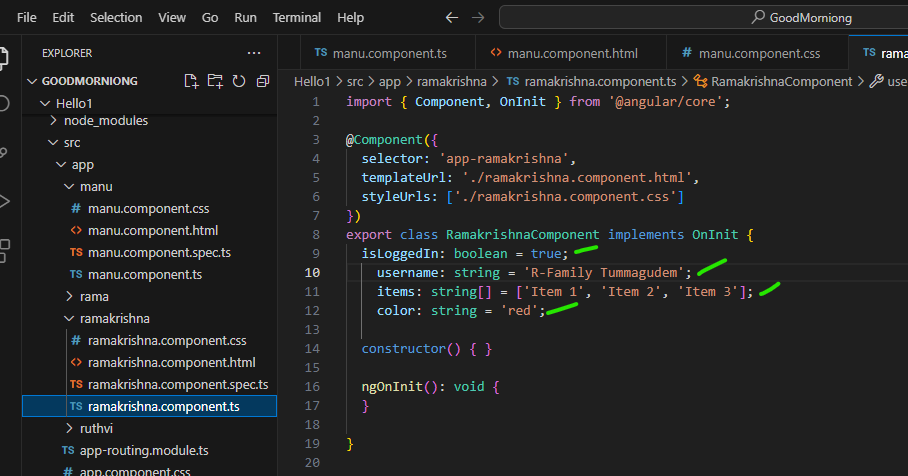
**Directives** are markers in the Document Object Model(DOM). Directives can be used with any controller or HTML tag which will tell the compiler what exact operation or behaviour is expected. There are some directives present that are predefined but if a developer wants he can create new directives (custom-directive).

[1. Component Directives](https://www.geeksforgeeks.org/built-in-directives-in-angular/#1-component-directives)

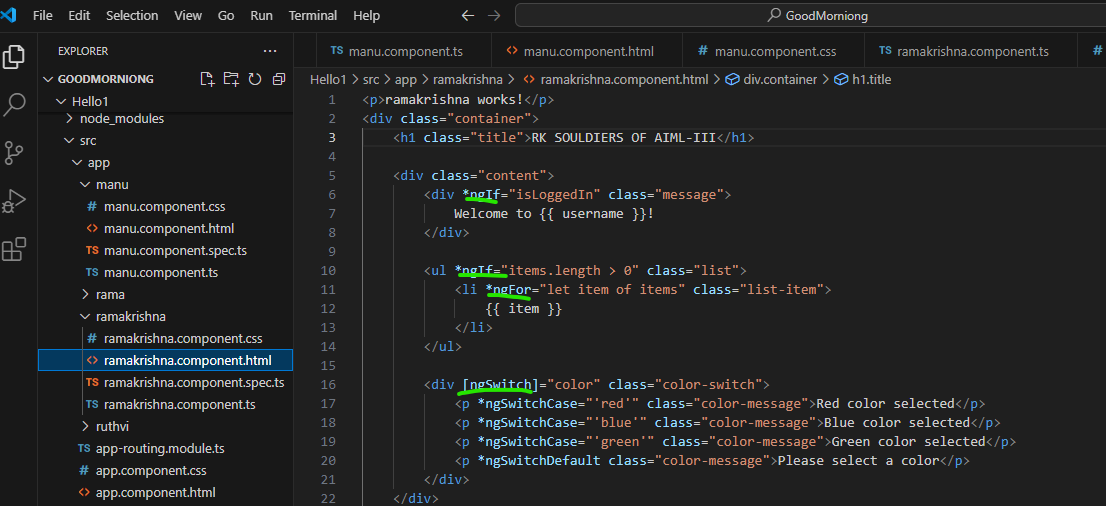
[2. Attribute Directives](https://www.geeksforgeeks.org/built-in-directives-in-angular/#2-attribute-directives)(ngClass ,ngStyle ,ngModal)

[3. Structural Directives](https://www.geeksforgeeks.org/built-in-directives-in-angular/#3-structural-directives)(ngIf,ngSwitch,ngFor)

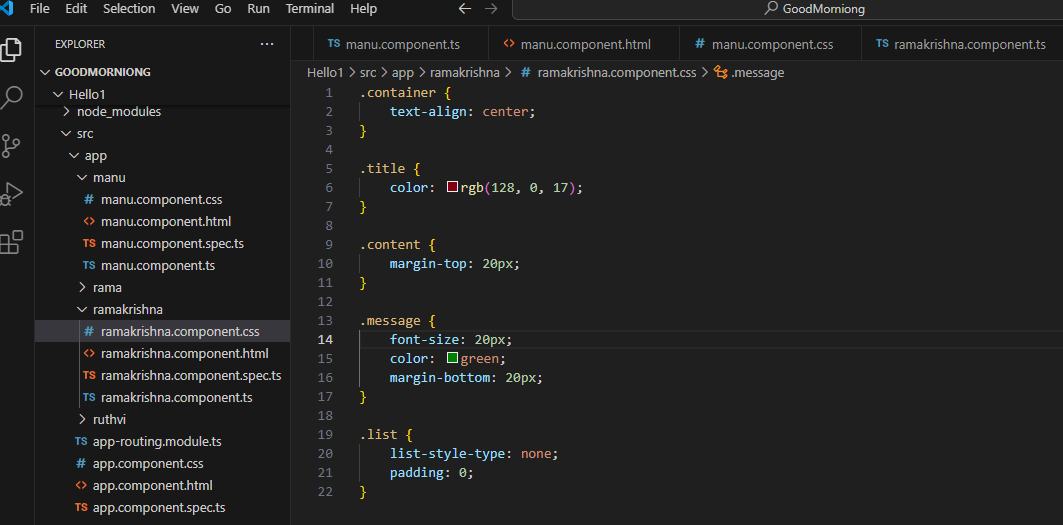
Ramakrishna.component.ts:



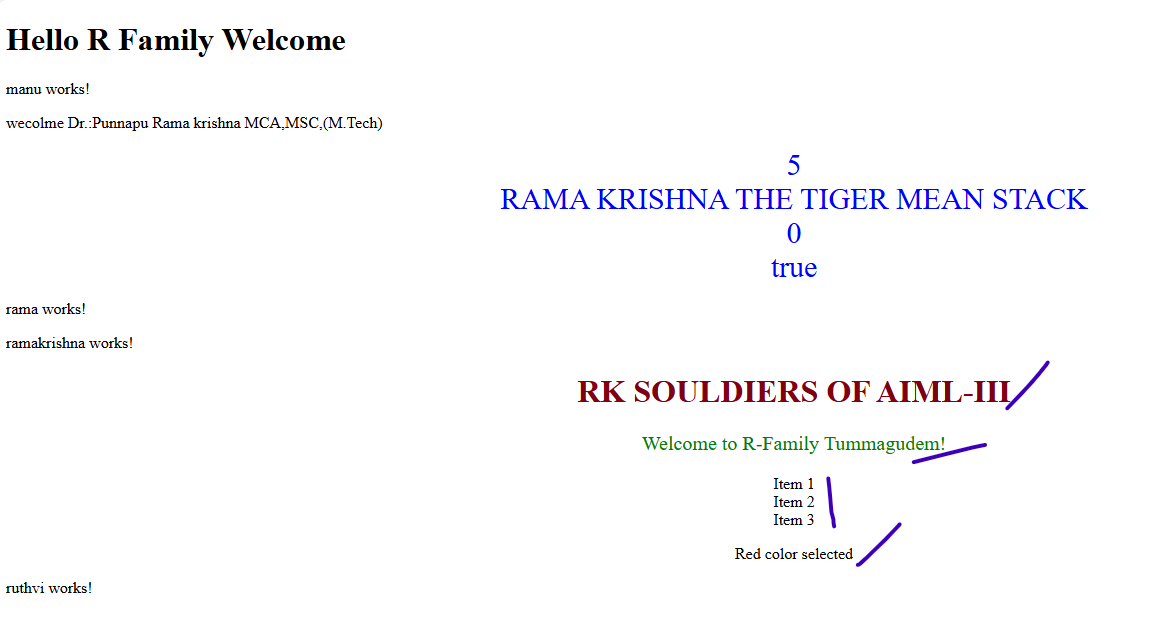
Ramakrishna.component.html:



Ramakrishna.component.css:



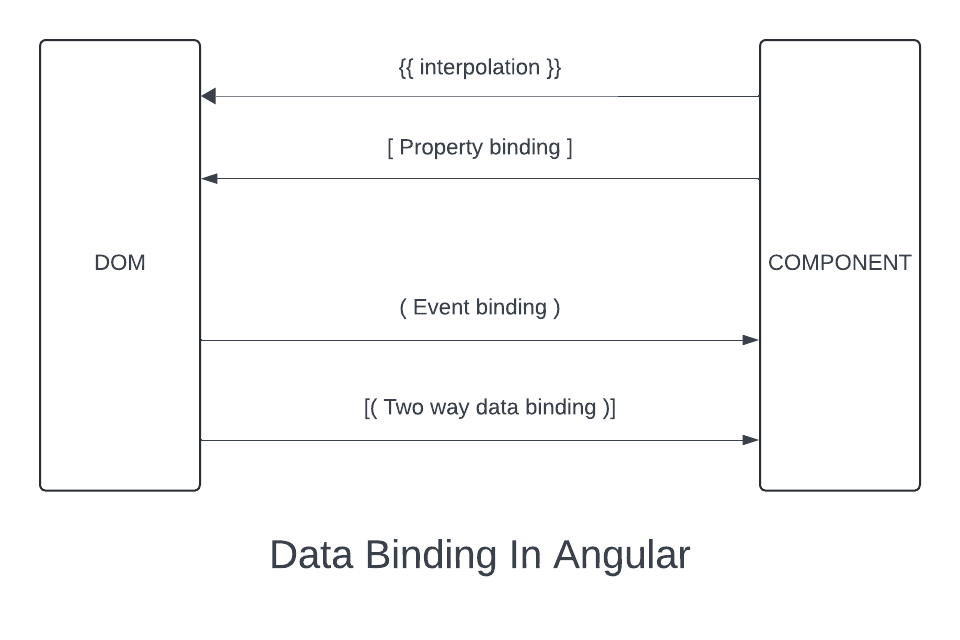
OUTPUT:



EXPERIMENT 5:DATA BINDING & PIPES:

1. DATA BINDING:

Data binding is the process that establishes a connection between the app UI and the data it displays. If the binding has the correct settings and the data provides the proper notifications, when the data changes its value, the elements that are bound to the data reflect changes automatically.



**Types of Data Binding in Angular**

1)One-Way Data Binding.

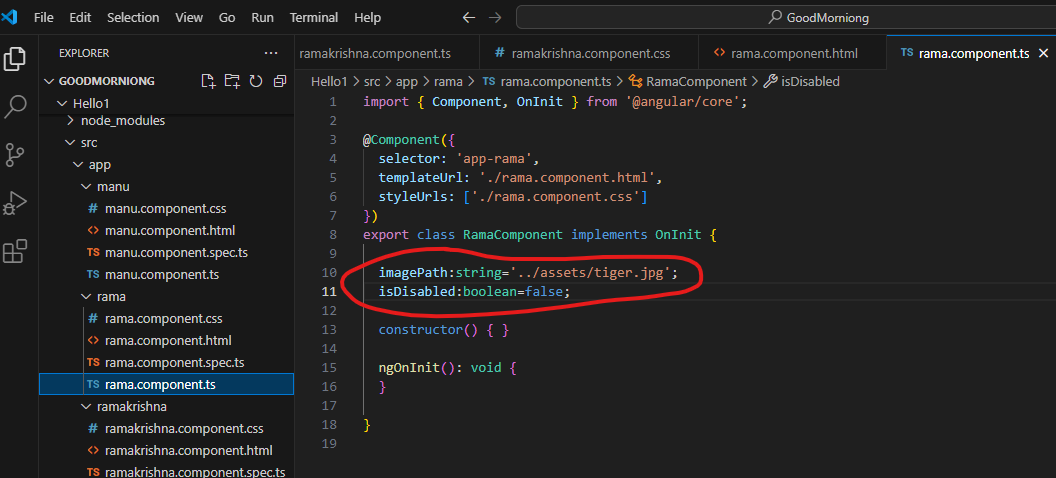
Interpolation.

Property Binding.

Event Binding.

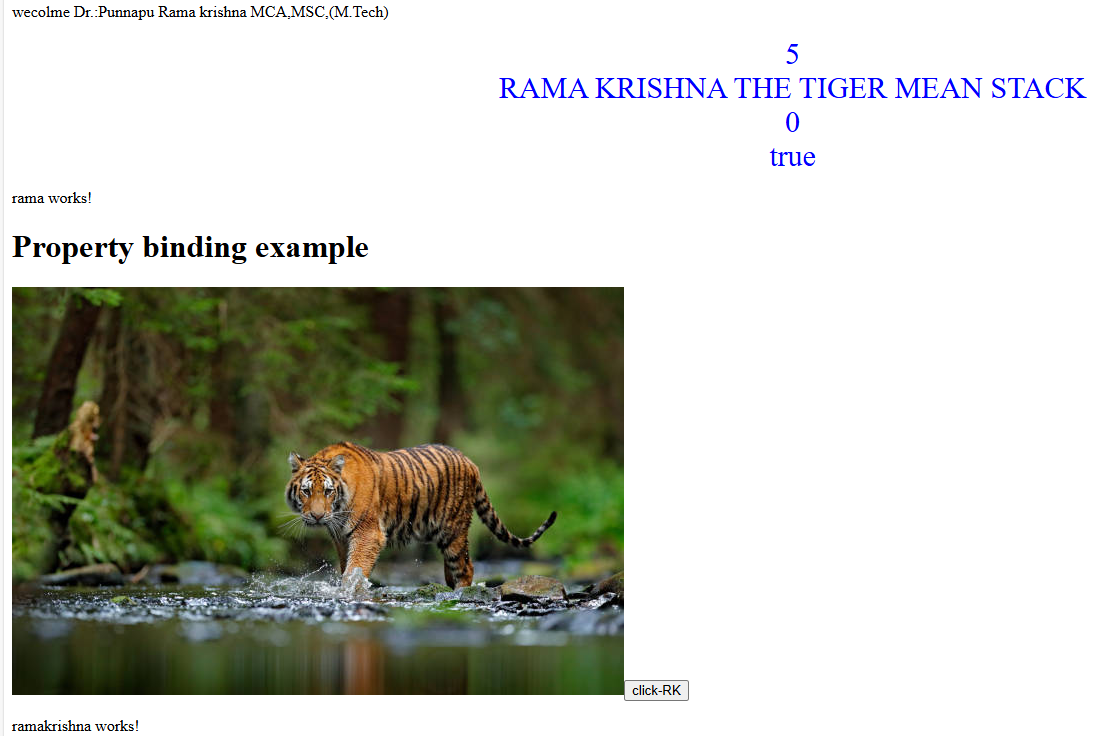
2)Two-Way Data Binding.( Property Binding + Event Binding)

Property Binding:



A screen shot of a computer

Description automatically generated

OUTPUT: 

B)PIPES: The pipe() function in [Angula**r**](https://www.geeksforgeeks.org/introduction-to-angularjs/) is used to chain multiple operators together to transform data. It receives an input value works on it and gives back a modified result. You can recognize pipes in your template expressions by the pipe symbol (**|**).

**1. Built-in Pipes**

Angular comes equipped with built in pipes that handle a variety of common formatting duties.

<p>Today's date: {{ today | date:'mediumDate' }}</p>

* **DatePipe**-The DatePipe is utilized for date formatting. It enables us to present dates, in styles like short, medium and full. For example we can utilize to exhibit the form of the date.

{{ myDate | date: "short" }}

* **UpperCasePipe -**This tool changes a text to capital letters. It requires a text, as input. Gives back the text in all capital letters.

{{ myString | uppercase }}

* **LowerCasePipe -**This particular pipe is utilized for changing a string to lowercase. Its functionality resembles that of the UpperCasePipe except it changes the string to lowercase instead.

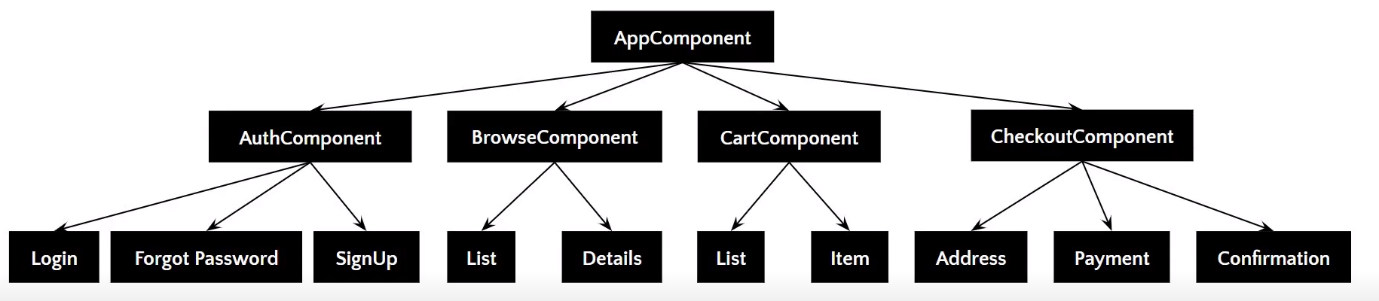
{{ myString | lowercase }}

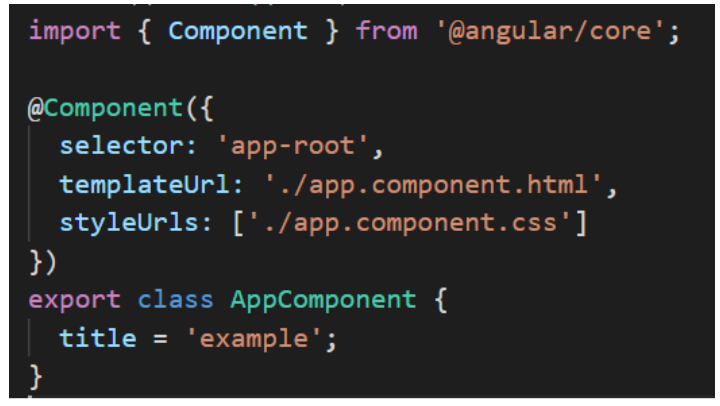
* **CurrencyPipe -**This tool helps to convert numbers into currency values. You input a number. It gives back a string showing the number, in the desired currency format.

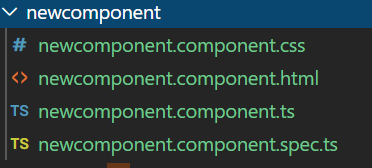
{{ myNumber | currency: "USD" }}

EXPERIMENT 6: MORE ON COMPONENTS

Angular components are the building blocks of a UI in an Angular application. These components are associated with a template and are a subset of directives.







OUTPUT: Components screen

EXPERIMENT 7 :BUILDING NESTED COMPONENTS