

Name of Student: Pushkar Sane		
Roll Number: 45		Practical Number: 9
Title of Lab Assignment: Introduction to Angular Setup for local development environment. Angular Architecture, Create an application to demonstrate directives and pipes.		
DOP: 22-09-2023		DOS: 23-09-2023
CO Mapped: CO5	PO Mapped: PO3, PO5, PSO1, PSO2	Signature:

Practical No. 9

Aim: Introduction to Angular Setup for local development environment. Angular Architecture, Create an application to demonstrate directives and pipes.

Description:

Angular is a popular open-source web application framework developed by Google and a community of individual developers. It is designed to simplify the process of building dynamic, single-page web applications. To set up an Angular development environment, you'll need to follow several steps:

Prerequisites:

Before you start with Angular, ensure that you have the following prerequisites installed on your system:

1. Node.js:

Angular uses Node.js for its build tools and package management. You can download and install it from Node.js website.

2. npm (Node Package Manager):

npm comes bundled with Node.js. It is used to install Angular CLI and other dependencies.

3. Angular CLI (Command Line Interface):

Angular CLI is a powerful command-line tool that makes it easy to create, build, test, and deploy Angular applications. To install Angular CLI, open your terminal or command prompt and run the following command:

npm install -g @angular/cli

This will globally install the Angular CLI tool.

Create a New Angular Project:

After Angular CLI is installed, you can create a new Angular project by running the following command:

ng new your-project-name

Replace `your-project-name` with your desired project name. Angular CLI will prompt you to configure various project settings, including whether to include Angular routing and which stylesheets (CSS, SCSS, etc.) to use.

Run the Development Server:

Navigate into your project folder:

`cd your-project-name`

Start the development server:

`ng serve`

This command will compile your Angular application and serve it on a development server. By default, you can access your application at <http://localhost:4200/>.

Project Structure:

Angular projects have a specific directory structure:

1. `src/`: This is where your application's source code resides.
2. `app/`: This folder contains the main application code.
3. `index.html`: The main HTML file for your application.
4. `angular.json`: Configuration file for Angular CLI.
5. `package.json`: Lists project dependencies and scripts.
6. `tsconfig.json`: Configuration for TypeScript.

Creating Components:

In Angular, your application is built using components. You can generate a new component using Angular CLI:

`ng generate component component-name`

Building and Deploying:

When you're ready to deploy your Angular application, you can use Angular CLI to build it:

`ng build --prod`

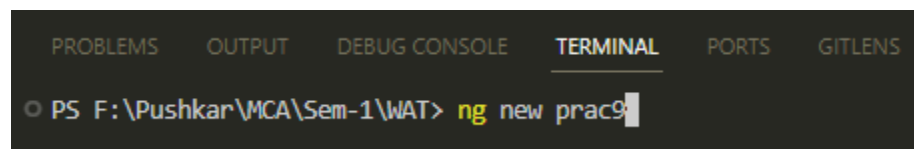
This will create a `dist/` directory with optimized production-ready code. You can then host this folder on a web server.

This is a basic introduction to setting up an Angular development environment. Angular offers a rich ecosystem for building complex web applications with features like dependency injection, routing, and powerful templating. As you become more familiar with Angular, you can explore its documentation and community resources to build robust web applications.

1. Create an application to demonstrate directives and pipes

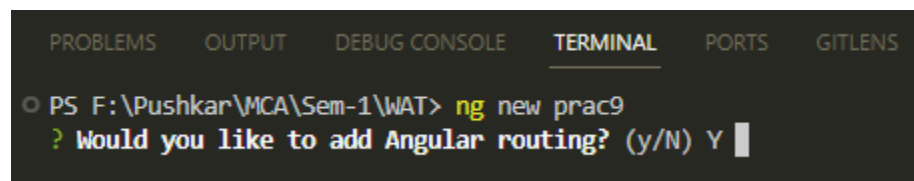
Steps to execute the practical:

1. Create a folder “practical 9”
2. Open the terminal (please use cmd, avoid using powershell) and navigate to “practical 9” folder by using the “cd” command
3. Install Angular CLI globally by running the following command
`npm install -g @angular/cli`
4. Create a new angular project by running the following command
`ng new prac9`



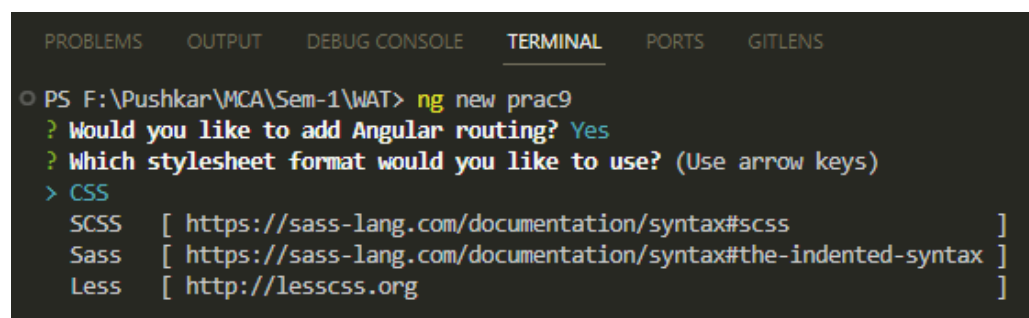
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac9
```

5. When prompted for Angular Routing, Enter “y”



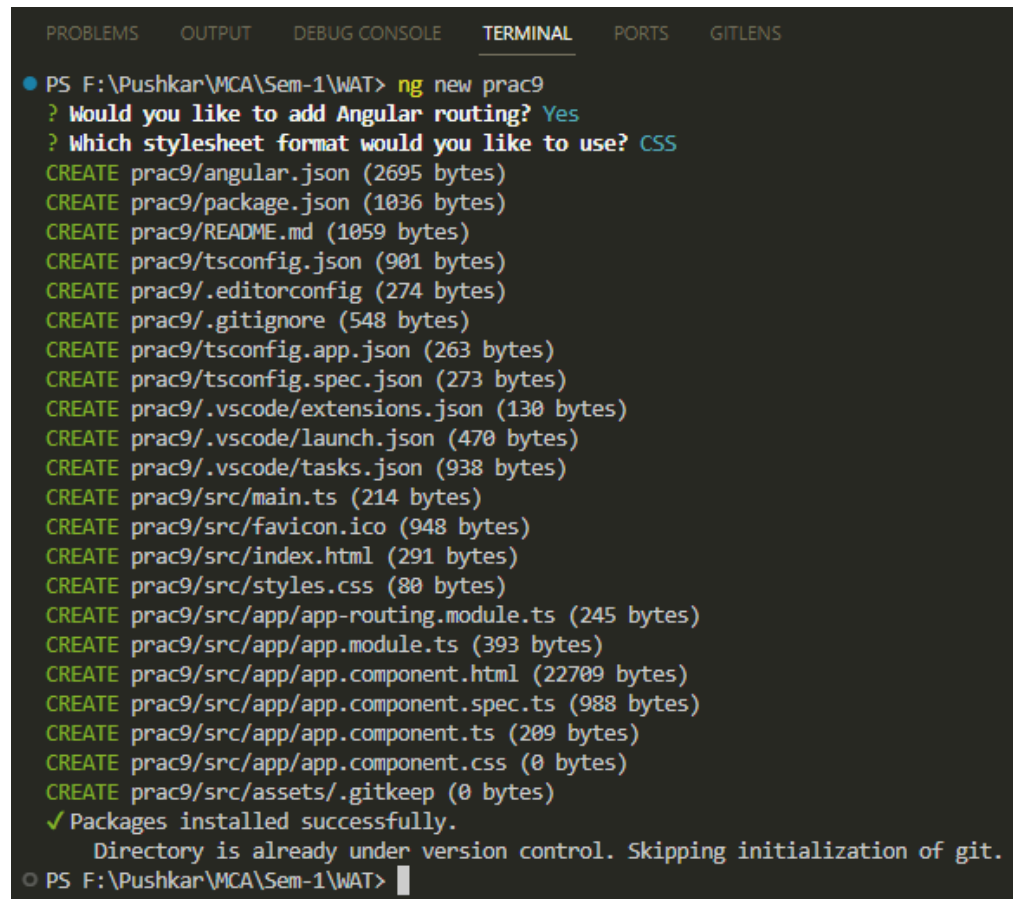
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac9
? Would you like to add Angular routing? (y/N) Y
```

6. When prompted for CSS, use arrow keys to select “CSS” (it is the first option and is selected by default) and then press Enter



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac9
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
```

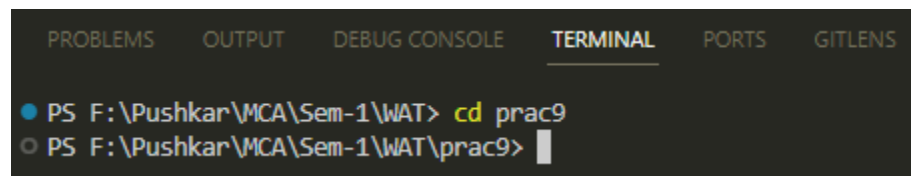
7. The new Angular project is created



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
● PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac9
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE prac9/angular.json (2695 bytes)
CREATE prac9/package.json (1036 bytes)
CREATE prac9/README.md (1059 bytes)
CREATE prac9/tsconfig.json (901 bytes)
CREATE prac9/.editorconfig (274 bytes)
CREATE prac9/.gitignore (548 bytes)
CREATE prac9/tsconfig.app.json (263 bytes)
CREATE prac9/tsconfig.spec.json (273 bytes)
CREATE prac9/.vscode/extensions.json (130 bytes)
CREATE prac9/.vscode/launch.json (470 bytes)
CREATE prac9/.vscode/tasks.json (938 bytes)
CREATE prac9/src/main.ts (214 bytes)
CREATE prac9/src/favicon.ico (948 bytes)
CREATE prac9/src/index.html (291 bytes)
CREATE prac9/src/styles.css (80 bytes)
CREATE prac9/src/app/app-routing.module.ts (245 bytes)
CREATE prac9/src/app/app.module.ts (393 bytes)
CREATE prac9/src/app/app.component.html (22709 bytes)
CREATE prac9/src/app/app.component.spec.ts (988 bytes)
CREATE prac9/src/app/app.component.ts (209 bytes)
CREATE prac9/src/app/app.component.css (0 bytes)
CREATE prac9/src/assets/.gitkeep (0 bytes)
✓ Packages installed successfully.
  Directory is already under version control. Skipping initialization of git.
○ PS F:\Pushkar\MCA\Sem-1\WAT>
```

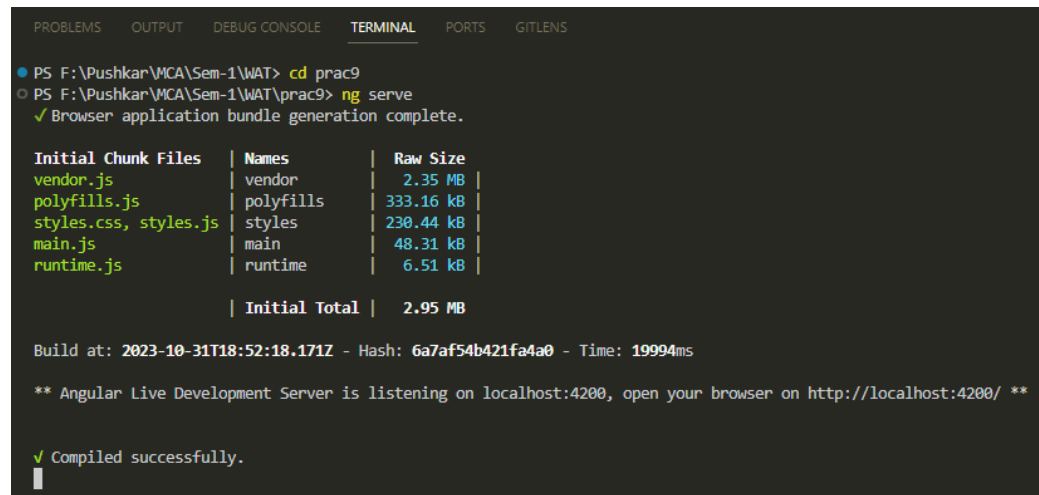
8. Navigate to the prac9 folder by using the cd command

cd prac9



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
● PS F:\Pushkar\MCA\Sem-1\WAT> cd prac9
○ PS F:\Pushkar\MCA\Sem-1\WAT\prac9>
```

9. Run the following command to serve the angular project on a server
ng serve



```
PS F:\Pushkar\MCA\Sem-1\WAT> cd prac9
PS F:\Pushkar\MCA\Sem-1\WAT\prac9> ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
vendor.js           | vendor         | 2.35 MB
polyfills.js        | polyfills      | 333.16 kB
styles.css, styles.js | styles         | 230.44 kB
main.js             | main           | 48.31 kB
runtime.js          | runtime        | 6.51 kB

| Initial Total | 2.95 MB

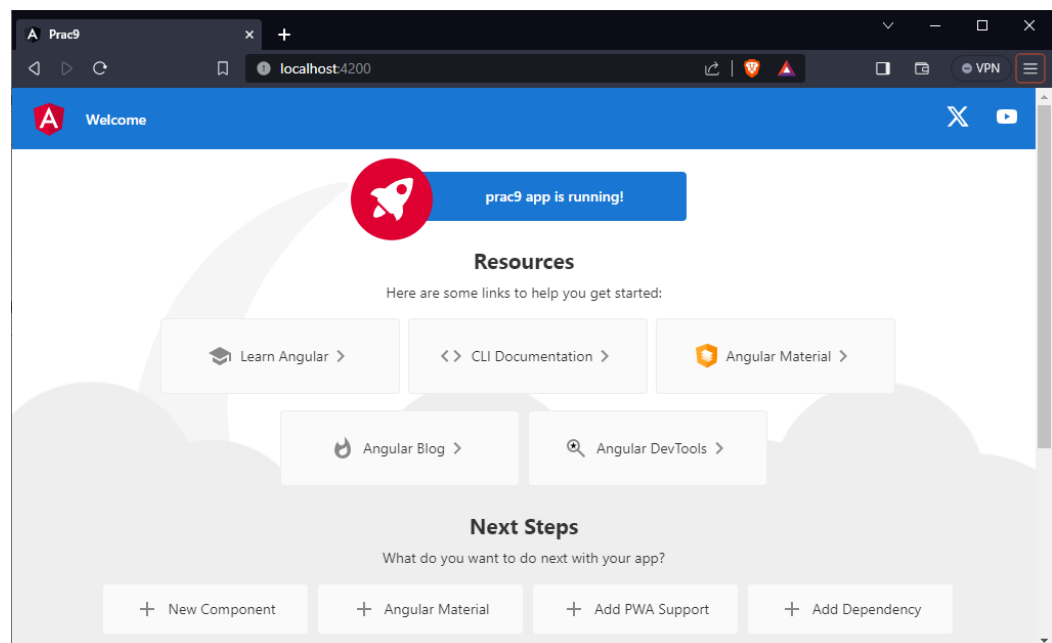
Build at: 2023-10-31T18:52:18.171Z - Hash: 6a7af54b421fa4a0 - Time: 19994ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

10. The server is running and will automatically re-compile the project when we make any changes.
11. Open any browser and go to the following link to see the output (this is the default output whenever any new Angular project is made)

<http://localhost:4200/>

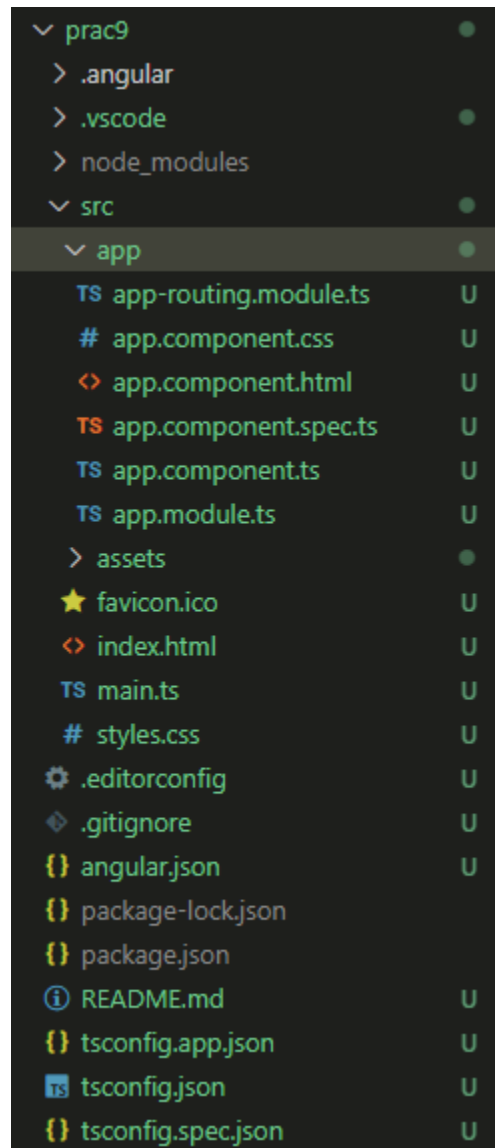


12. We will remove the default code that's generated and add our practical code.

We will make changes to the following 2 files in the "app" folder (prac9/src/app/)

app.component.html

app.component.ts



13. Open the **app.component.html** file and delete all the code

14. Write the following code in **app.component.html**:

<!-- directive -->

<!-- ngIf directive -->

<!-- reference:- <https://www.javatpoint.com/angular-8-ngif-directive> -->

```
<h1> Directives </h1>
```

```
<h2> ngIf </h2>
```

```
n is = {{n}}
```

```
<div *ngIf="n > 2; else elseBlock">
```

```
  n is greater than 2
```

```
</div>
```

```
<ng-template #elseBlock>
```

```
  n is not greater than 2
```

```
</ng-template>
```

```
<!-- ngFor directive -->
```

```
<!-- reference:- https://www.javatpoint.com/angular-8-ngfor-directive -->
```

```
<h2> ngFor </h2>
```

```
<li *ngFor="let item of items">
```

```
  {{item}}
```

```
</li>
```

```
<!-- piping -->
```

```
<!-- reference:- https://www.tutorialspoint.com/angular8/angular8\_pipes.htm -->
```

```
<h1> Piping </h1>
```

```
<h2> Adding parameter </h2>
```

```
<!-- adding parameter -->
```

```
<div>
```

```
  Today's date :- {{presentDate}}
```

```
</div>
```

```
<div>
```

```
  Date with uppercase :- {{presentDate | date:'fullDate' | uppercase}}
```

```
<br />
```

```
  Date with lowercase :- {{presentDate | date:'medium' | lowercase}}
```



```
<br />
</div>

<!-- currency pipe -->
<h2> Currency pipe </h2>

<div>
  <p>{{ amount | currency:'EUR'}}</p>
  <p>{{ amount | currency:'INR' }}</p>
</div>
<router-outlet></router-outlet>
```

15. Save the **app.component.html** file.

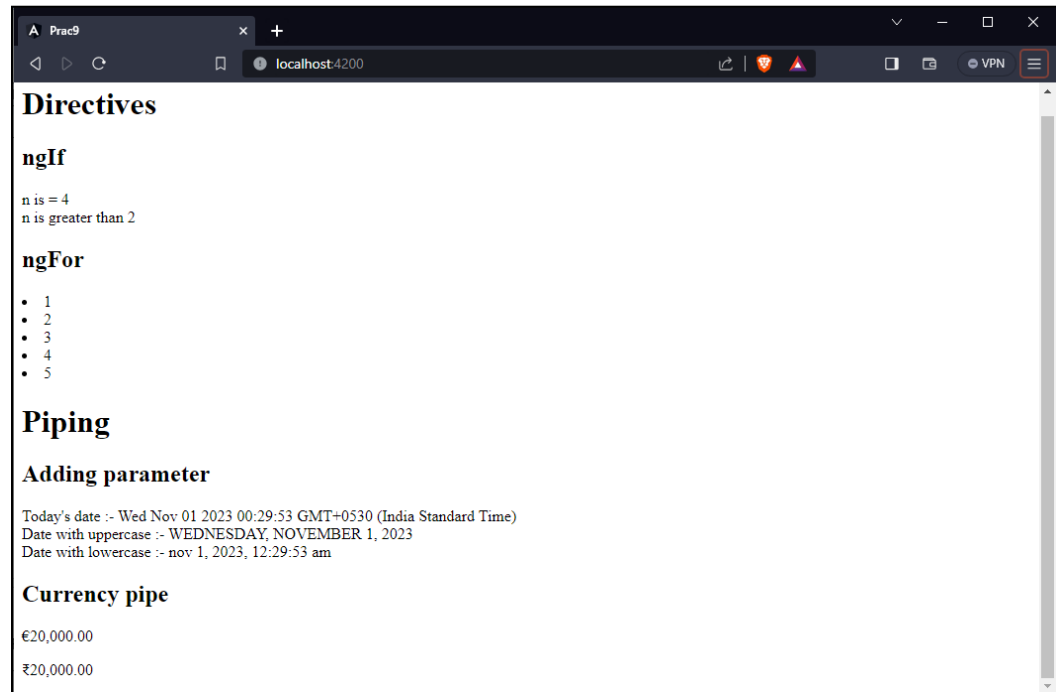
16. Open the **app.component.ts** and make the following changes in it. The app.component.ts code should look like this:

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = 'prac9';
  // added variables
  n: number = 4;
  presentDate: Date = new Date();
  amount: number = 20000;
  items: number[] = [1, 2, 3, 4, 5];
}
```

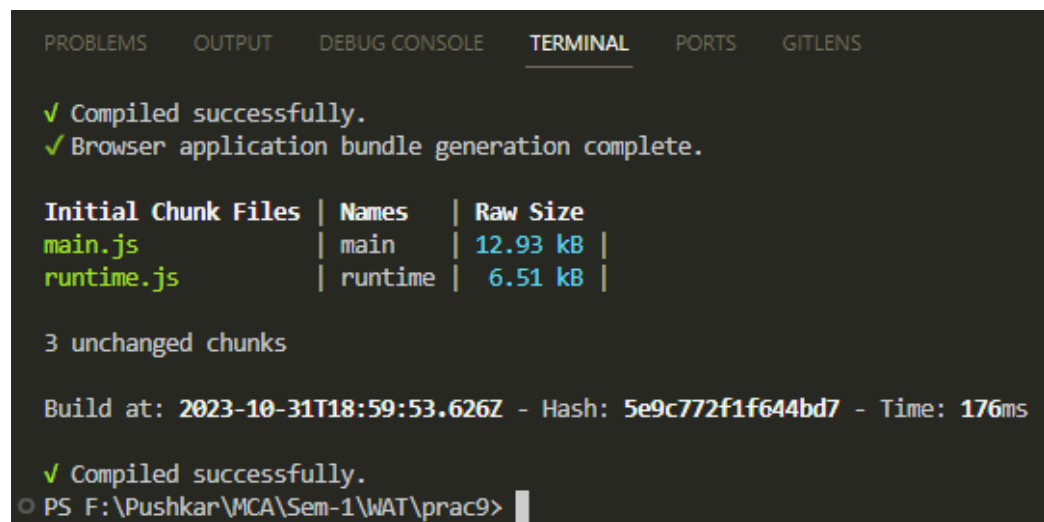
17. Save the **app.component.ts** file

18. Go to the browser to see the output at **http://localhost:4200/**



19. Go to the terminal and stop the server by pressing Ctrl + C

When we see the prompt as `"....\prac9>"` then it means that the server has stopped.



Code:**app.component.html**

```
<!-- directive -->
<!-- ngIf directive -->
<!-- reference:- https://www.javatpoint.com/angular-8-ngif-directive -->

<h1> Directives </h1>

<h2> ngIf </h2>

n is = {{n}}

<div *ngIf="n > 2; else elseBlock">
  n is greater than 2
</div>

<ng-template #elseBlock>
  n is not greater than 2
</ng-template>

<!-- ngFor directive -->
<!-- reference:- https://www.javatpoint.com/angular-8-ngfor-directive -->

<h2> ngFor </h2>
<li *ngFor="let item of items">
  {{item}}
</li>

<!-- piping -->
<!-- reference:- https://www.tutorialspoint.com/angular8/angular8_pipes.htm -->

<h1> Piping </h1>
<h2> Adding parameter </h2>
```

```
<!-- adding parameter -->
<div>
  Today's date :- {{presentDate}}
</div>

<div>
  Date with uppercase :- {{presentDate | date:'fullDate' | uppercase}}
<br />
  Date with lowercase :- {{presentDate | date:'medium' | lowercase}}
<br />
</div>

<!-- currency pipe -->
<h2> Currency pipe </h2>

<div>
  <p>{{ amount | currency:'EUR'}}</p>
  <p>{{ amount | currency:'INR' }}</p>
</div>

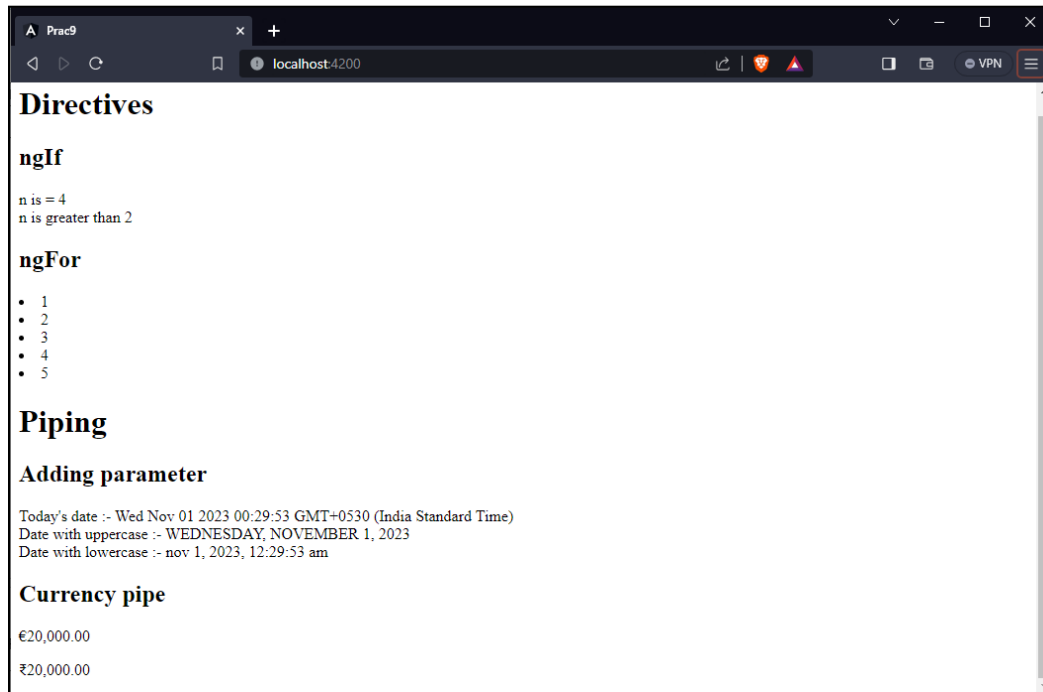
<router-outlet></router-outlet>
```

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = 'prac9';
  // added variables
  n: number = 4;
```

```
presentDate: Date = new Date();  
amount: number = 20000;  
items: number[] = [1, 2, 3, 4, 5];  
}
```

Output:

Conclusion: We've explored an introduction to Node.JS, Advantages and Disadvantages, Node.js Process Model, Traditional Web Server Model, Installation and some programs with node.js.