GANATAN

Bootstrap with Angular 20

Complete Guide

Update on May 29, 2025

Bootstrap is the most popular CSS framework for designing modern, responsive, and optimized web interfaces.

In this tutorial, we will integrate **Bootstrap 5.3.6** into an application developed with **Angular 20.0.0** , applying best practices to ensure performance, maintainability, and scalability.

Learn how to **install Bootstrap** and configure its styles and components with Angular.

What we are going to do

- **What is Bootstrap?**
  We are going to talk about bootstrap and its usefulness in websites.
- **Creating our Angular project**
  We will use an existing project containing the essential functionality.
  The project was built with Angular CLI and uses Routing and Lazy Loading.
- **Bootstrap Prototype**
  Why and how to create a Bootstrap prototype?
- **Version 4 versus Version 5**
  Which Bootstrap Version Should You Choose?
- **Integration into our Angular project**
  How to integrate Bootstrap using Angular CLI?
- **Perform the Tests**
  We will test our application through the unit and end-to-end tests built into Angular.
- **Source code**
  For those of you in a hurry, the complete code of the project.
  https://github.com/ganatan/angular-react-bootstrap

What is Bootstrap?

The pages of a **website** are written using 3 types of **computer languages** :

- **HTML language**

  HyperText Markup Language

  It allows you to create and represent the content of a web page and its structure.

- **The CSS language**

  Cascading Style Sheets

  It is used to describe the presentation of a page.

- **The Javascript language**

  It allows you to create interactive features in the page.

There are many tools that can make a developer's life easier.
In particular, **CSS Frameworks** , which are a kind of graphical toolbox.

There are a huge number of **CSS frameworks** , among which we could mention

- Foundation
- Materialize CSS
- Bootstrap
- Pure
- Bulma
- Tailwind

Bootstrap is one of the most well-known and widely used CSS frameworks.

Bootstrap has been **open source** since 2011.
Its **use** is therefore **free** .

It allows you to create responsive sites.
This means that it allows you to adapt a website to all types of devices (computer, smartphone or tablet).
The current version is **5.3.6**

Creating the Angular project

To be able to continue this tutorial we obviously need to have certain elements

- **Node.js** : The JavaScript platform
- **Git** : The version control software.
- **Angular CLI** : The tool provided by Angular.
- **Visual Studio Code** : A code editor.

You can check out the following tutorial which explains in detail how to do it
- https://www.ganatan.com/tutorials/demarrer-avec-angular

We will use an existing project whose characteristics are
- Generated with Angular CLI
- Routing
- Lazy loading

You can install this project on your workstation using the following commands.

```
# Create a demo directory (the name is arbitrary here)
mkdir demo

# Go to this directory
cd demo

# Get the source code on your workstation
git clone https://github.com/ganatan/angular-lazy-loading.git

# Go to the directory that was created
cd angular-lazy-loading
cd frontend-angular

# Run the dependencies (or libraries) installation
npm install

# Run the program
npm run start

# Check its operation by launching the command in your browser
http://localhost:4200/
```

Why Create a Bootstrap Prototype?

To simplify there are 3 types of web developers
- **Frontend**
- **Backend**
- **Fullstack**

We could add a 4th profession
- **Web Designer**

Still simplifying, the **first 3** take care of the **logical** part and the **fourth** of the **graphic** part.

This is why we will first create a graphic prototype.
This prototype will serve as the basis for the design phase of your website.

Some reasons for this choice.
- It is more difficult and expensive to make changes in code rather than in a prototype.
- If you're working in a team, it's easier to know who does what.
- There are many prototyping tools available.
- It facilitates the evolution and improvement of the interface.
- It is faster and easier to present a prototype to a potential customer.

SO.

To do this, it is not necessary to know Angular.
Only knowledge of HTML, CSS and a little javascript will be necessary.
It will even be possible to **subcontract** to a Web Designer or graphic designer partner.

These are some of the elements from this prototype that we will integrate into our Angular application.

The architecture of this prototype and the choice of names is of course arbitrary.
You can adapt it to your liking.

Quick summary of prototyping

Designing a website will go through several stages.

One of them is **prototyping** .

Prototyping involves creating a non-final **copy** of what the **final product** might be.

It allows you to design a User Interface or **UI** (User Interface)

There are 2 stages of prototyping.

- **Horizontal prototyping**

  Create a static mockup.

  Draw a wireframe

  Define areas and components

- **Vertical prototyping**

  Integrate features

In this tutorial we will simply create a **static model.**

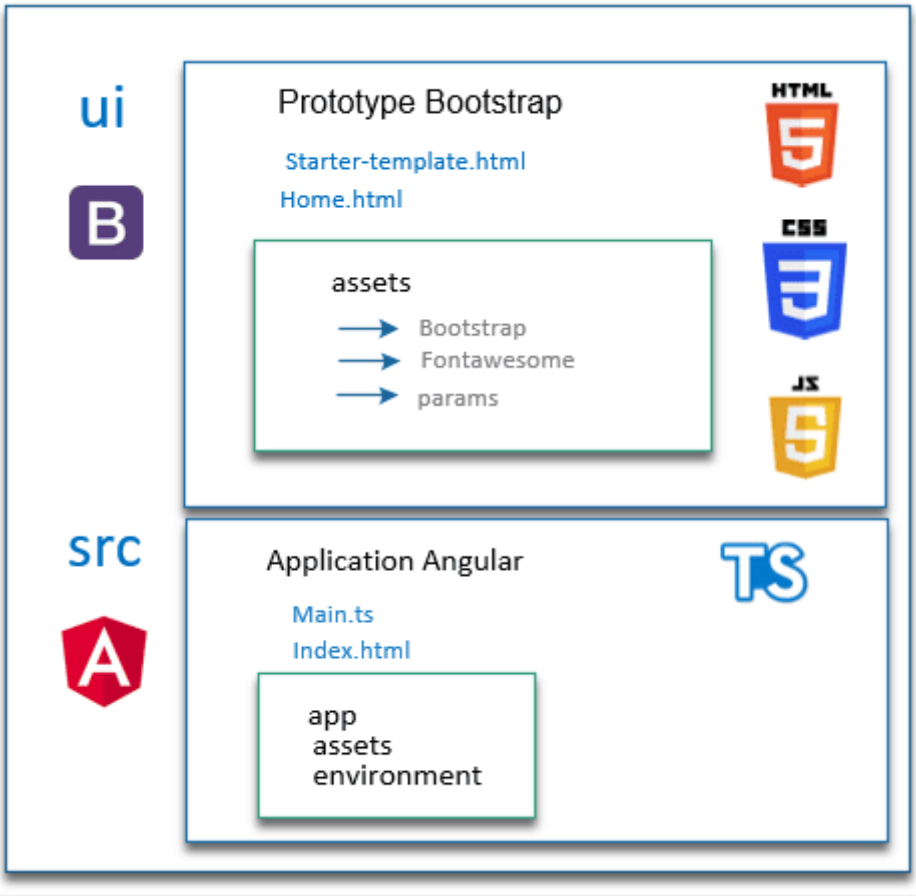
How to create a bootstrap prototype?

To understand what we are doing, we will not use any prototyping tools.

We will design this prototype **manually** and from **scratch** .

The directory that will contain this prototype will be **UI** or User Interface.

An image will allow us to have an overview of our prototype within our Angular application.



Let's move on to practice.

We will search the web for all the necessary elements that we will place in the tree structure of our Angular project.

We will finally obtain the following tree structure.

```
|-- node_modules/          (contient les librairies node)
|-- src/          (contient le code source de notre application angular)
|-- ui/                    ( contient notre prototype bootstrap)
    |-- assets
        |-- bootstrap
            |-- css
            |-- js
        |-- fontawesome
            |-- css
            |-- js
            |-- webfonts
        |-- params
            |-- css
            |-- images
            |-- js
package.json
```

Version 4 vs Version 5

Bootstrap 4 is really nice.

But version 5 was released on **June 16, 2020.**

The latest version is **5.3.6**

It is interesting to see that this version brings two major changes.

- The removal of Jquery.

  And consequently the use of Vanilla JavaScript (in other words, JavaScript, nothing but JavaScript)
- Dropping compatibility with IE 10 and 11 (Internet Explorer)

  It was not too early

The final project will use version 5, I tested it, it works, so why not!
To hell with caution.
We'll have time to be careful when we're dead.

The elements of the prototype

So let's get started!

In the **ui** directory create an **assets directory.**

In this **assets** directory create the following 3 directories

- **bootstrap**
- **fontawesome**
- **params**

Now let's collect the various elements useful for our prototype.

Bootstrap Elements

- Go to the bootstrap site (download part)

https://v5.getbootstrap.com/docs/5.3/getting-started/download/

- Download the compiled CSS and JS files (**compiled CSS and JS** / **download**)
- The file we are interested in is **bootstrap-5.3.6-dist.zip**
- Unzip the bootstrap-5.3.6-dist.zip file
- Copy the **CSS** and **JS** directories into the **ui/assets/bootstrap** directory

The elements of fontawesome

- Go to the fontawesome website (Download section)

https://fontawesome.com/download

- **Download font Awesome free for the web
Download** the compiled CSS and JS files
- The file we are interested in is **fontawesome-free-6.5.2-web.zip**
- Unzip the fontawesome-free-6.5.2-web.zip file
- Copy the **CSS, JS** and **webfonts** directories into the **ui/assets/fontawesome** directory

Noticed

Any installed files will not be used.

In this case we can clean up and leave only the essentials.

For fontawesome we will only keep the files all.css , all.min.css and all.js and all.min.js

You will find the final result on the source code repository indicated at the end of the tutorial.

In the source code you will find the directory
- **ui** (bootstrap 5 code)

Our first Bootstrap page

We will create our first page using the examples provided by Bootstrap as inspiration.

We will use the examples page
- https://v5.getbootstrap.com/docs/5.3/examples/

Arbitrarily I give you the final result of an example page.
The different parts follow.

```
<style>
  body {
    padding-top: 3.5rem;
    font-family: "Roboto", sans-serif;
  }
 </style>
```

style header

```
<!-- HEADER -->
<style>
       .navbar.navbar-dark .navbar- nav  . nav -item . nav -link {
     color : white;
     font-weight :  500       border - top : 1px solid #09238d;
     border - bottom : 1px solid #09238d;
}

       .navbar.navbar-dark .navbar- nav  . nav -item . nav -link:hover {
     color : yellow;
     border - top : 1px solid yellow;
     border - bottom : 1px solid yellow;
}

.nga-navbar {
          -webkit-box-shadow:  0  2px 5px  0  rgba( 0 ,  0 ,  0 ,  0.16 ),  0  11px 10px  0  rgba( 0 ,  0 ,  0 ,
0.12 );
          box-shadow:  0  2px 5px  0  rgba( 0 ,  0 ,  0 ,  0.16 ),  0  11px 10px  0  rgba( 0 ,  0 ,  0 ,  0.12 );
     background - color : #09238d;
}

.nga-logo {
     font-weight :  700 }

.nga-logo:hover {
     color : rgba( 255 ,  255 ,  255 ,  0.75 );
}

.nga-btn-navbar {
     color : #fff;
     background - color : #1976d2;
     border - color : #0d6efd;
}

.nga-btn-navbar:hover {
     color : white;
     background - color : #0b5ed7;
     border - color : #0a58ca;
}
</style>
```

style footer

```
<!-- FOOTER -->
<style>
        .nga- footer       background - color : # 212121       color : white;
}

        .nga- footer a  {
      color : white;
text-decoration: none
}

        .nga- footer a :hover,
        .nga- footer a :focus {
      color : yellow;
text-decoration: underline;
}

        .nga- footer .hint {
      background - color : #1976d2;
}

        .nga- footer .hint:hover {
      opacity :  0.8 }
</style>
```

style home

```
<!-- HOME -->
<style>
  .nga-card {
    display: block;
    background-color: rgba(255, 255, 255, .8);
    box-shadow: 0 1px 3px rgba(0, 0, 0, .12), 0 1px 2px rgba(0, 0, 0, .24);
    border-radius: 2px;
    transition: all .2s ease-in-out;
    cursor: pointer;
  }

  .nga-card:hover {
    box-shadow: 0 10px 20px rgba(0, 0, 0, .19), 0 6px 6px rgba(0, 0, 0, .23);
  }
</style>
```

html header

```
<header class="navbar navbar-expand-md navbar-dark fixed-top nga-navbar">
  <nav class="container" aria-label="Main navigation">
    <a href="" class="navbar-brand" alt="Accueil" aria-label="Ganatan">
      <img src="./assets/params/images/logo/ganatan-logo.png" srcset="./assets/params/images/logo/ganatan-
logo.png,
      ./assets/params/images/logo/ganatan-logo@2x.png 2x" width="25" height="25" alt="Ganatan Logo">
      <span class="nga-logo mx-1">ganatan</span>
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"
      aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav mx-auto">
        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="">
            <i class="fas fa-home me-1"></i>Home</a>
        </li>
      </ul>
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="">
            <i class="far fa-question-circle me-1"></i>About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" href="">
            <i class="fas fa-envelope me-1"></i>Contact</a>
        </li>
      </ul>
      <form class="d-flex">
        <button type="button" class="btn btn-sm nga-btn-navbar me-2"><i class="fas fa-user-plus me-2"></
i>Sign
          up</button>
        <button type="button" class="btn btn-sm btn-outline-light me-2"><i
            class="fas fa-sign-in-alt me-2"></i>Login</button>
      </form>
    </div>
  </nav>
</header>
```

html main

```
<main>

 <div class="container py-5">
  <div class="row">
   <div class="col-12 col-sm-12 col-md-12 col-lg-3 col-xl-3 text-center mb-2">
    <h1 class="h5">
     <i class="fas fa-laptop fa-lg me-2 text-primary"></i>
     angular-starter
     <i class="fas fa-mobile-alt fa-lg ms-2 text-primary"></i>
    </h1>
   </div>
   <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-danger mb-2">
    <h2 class="h5">
     Angular 16.2.4<i class="fab fa-angular fa-lg ms-2"></i>
    </h2>
   </div>
   <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-primary mb-2">
    <h2 class="h5">
     Bootstrap 5.3.1<i class="fab fa-bootstrap fa-lg ms-2"></i>
    </h2>
   </div>
   <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-success mb-2">
    <h2 class="h5">
     Font Awesome 6.4.2<i class="fab fa-font-awesome-flag fa-lg ms-2"></i>
    </h2>
   </div>
  </div>
  <hr>
  <div class="row mb-2">
   <div class="col-md-12 text-center mb-4">
    <h3 class="h5">Features<i class="fas fa-list ms-2"></i></h3>
   </div>
  </div>
  <div class="row pt-2">
   <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
    <div class="nga-card bg-light mb-3">
     <a routerLink="/bootstrap">
      <div class="card-header">
       <div class="row">
        <div class="col-10 col-xl-10">
         <h5 class="card-title">Bootstrap</h5>
        </div>
        <div class="col-2 col-xl-2">
         <i class="fab fa-bootstrap fa-lg text-primary"></i>
        </div>
       </div>
      </div>
      <div class="card-body">
       <p class="card-text">How to use Buttons, Alerts, Pagination, Tables, Collapses</p>
      </div>
     </a>
    </div>
   </div>
  </div>
 </div>

</main>
```

Bootstrap integration in angular

Now let's move on to the **logical** part.

We're going to make this **home** page work in **our base Angular project** .

First we need to add the necessary libraries.

For this we use **npm** (node package manager) the dependency manager of **Nodes.js**

For bootstrap we will follow the advice on the official website.

https://v5.getbootstrap.com/docs/5.1/getting-started/download/

```
# Adding dependencies to package.json
npm install --save bootstrap
npm install --save @fortawesome/fontawesome-free
```

As we saw in the Getting Started with Angular tutorial we will modify the dependency version descriptors.

Regarding dependencies and their version the npm documentation is as follows

https://docs.npmjs.com/files/package.json#dependencies

Which will give us the following result.

```
"@fortawesome/fontawesome-free": "6.7.2",
"bootstrap": "5.3.6",
"rxjs": "7.8.2",
"tslib": "2.8.1",
"zone.js": "0.15.1"
```

Update

So let's reuse the elements of our prototype.

Copy the entire **params** directory from our prototype into our angular application at **src/assets**

We will modify the **angular.json** file to call the files necessary for our html pages to work.

CSS formatting files

- **index.css** (specific to our project via params)
- **all.min.css** (fontawesome specific via node_modules)
- **bootstrap.min.css** (bootstrap specific via node_modules)

Javascript script files

- **bootstrap.min.js** (bootstrap specific via node_modules)

  Noticed

  In our case we will use the **bootstrap.bundle.min.js** file

  We can then use Bootstrap components like **Toasts** , **Tooltips** or **Popovers**

  https://v5.getbootstrap.com/docs/5.3/components/tooltips/

  https://v5.getbootstrap.com/docs/5.3/components/popovers/

  https://v5.getbootstrap.com/docs/5.3/components/toasts/

- **index.js**

  To add the specific code to our project here

Note: In case of version 5
I prefer to add it on app.component at the end of page loading and avoid problems
This avoids using functions like document.ready for example

angular.json

```
      "build": {
        "builder": "@angular-devkit/build-angular:application",
        "options": {
          "outputPath": "dist/angular-starter",
          "index": "src/index.html",
          "browser": "src/main.ts",
          "polyfills": [
            "zone.js"
          ],
          "tsConfig": "tsconfig.app.json",
          "assets": [
            "src/favicon.ico",
            "src/assets"
          ],
          "styles": [
            "node_modules/@fortawesome/fontawesome-free/css/all.min.css",
            "node_modules/bootstrap/dist/css/bootstrap.min.css",
            "src/assets/params/css/fonts.googleapis.min.css",
            "src/styles.css"
          ],
          "scripts": [
            "node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
          ]
        },
```

src/style.css

```
body {
  padding - top :  3 .5rem;
    font-family:  "Roboto" " , sans-serif;
}
```

We will modify the following files which will contain the new interface

- **app.component.html**
- **app.component.css**
- **home.component.html**
- **home.component.ts**
- **home.component.css**
- **home.component.spec.ts**
- **environment.ts**
- **environment.development.ts**
- **app.component.spec.ts**
- **app.component.ts**
- **about.html**
- **signin.css**
- **contact.html**
- **notfound.html**

We will add the images used in the **assets/params/images/logo** directory

- **ganatan.png**

src/app/app.component.html

```
<header class="navbar navbar-expand-md navbar-dark fixed-top nga-navbar">
  <nav class="container" aria-label="Main navigation">
    <a routerLink="/" class="navbar-brand" alt="Accueil" aria-label="Ganatan">
      <img src="./assets/params/images/logo/ganatan-logo.png" srcset="./assets/params/images/logo/ganatan-logo.png,
      ./assets/params/images/logo/ganatan-logo@2x.png 2x" width="25" height="25" alt="Logo Ganatan">
      <span class="nga-logo mx-1">ganatan</span>
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"      aria-controls="navbarCollapse" aria-expanded=
"false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav mx-auto">
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLink="/">
            <i class="fas fa-home me-1"></i>Home</a>
        </li>
      </ul>
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" routerLink="/about">
            <i class="far fa-question-circle me-1"></i>About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLink="/contact">
            <i class="fas fa-envelope me-1"></i>Contact</a>
        </li>
      </ul>
      <form class="d-flex">
        <button type="button" class="btn btn-sm btn-primary me-2" routerLink="/signup"><i
            class="fas fa-user-plus me-2"></i>Sign up</button>
        <button type="button" class="btn btn-sm btn-outline-light me-2" routerLink="/login"><i
            class="fas fa-sign-in-alt me-2"></i>Login</button>
      </form>
    </div>
  </nav>
</header>

<main>
  <router-outlet></router-outlet>
</main>

<footer class="nga-footer">
  <div class="py-3 text-center" style="background-color: black;">
    <div class="container">
      <a href="{{ footerUrl }}">{{ footerLink }}</a>
    </div>
  </div>
</footer>
```

src/app/app.component.css

```
.navbar.navbar-dark .navbar- nav  . nav -item . nav -link {
    color : white;
    font-weight :  500     border - top : 1px solid #09238d;
    border - bottom : 1px solid #09238d;
}

    .navbar.navbar-dark .navbar- nav  . nav -item . nav -link:hover {
    color : yellow;
    border - top : 1px solid yellow;
    border - bottom : 1px solid yellow;
}

.nga-navbar {
        -webkit-box-shadow:  0  2px 5px  0  rgba( 0 ,  0 ,  0 ,  0.16 ),  0  11px 10px  0  rgba( 0 ,  0 ,  0 ,  0.12
        box-shadow:  0  2px 5px  0  rgba( 0 ,  0 ,  0 ,  0.16 ),  0  11px 10px  0  rgba( 0 ,  0 ,  0 ,  0.12
background - color : #09238d;
}

.nga-navbar-logo {
    font-weight :  700 }

.nga-navbar-logo:hover {
    color : rgba( 255 ,  255 ,  255 ,  0.75 }

.nga-logo {
    font-weight :  700 }

.nga-logo:hover {
    color : rgba( 255 ,  255 ,  255 ,  0.75 }

    .nga- footer {
    background - color : # 212121     color : white;
}

    .nga- footer a  {
    color : white;
text-decoration: none
}

    .nga- footer a :hover,
    .nga- footer a :focus {
    color : yellow;
text-decoration: underline;
}

    .nga- footer .hint {
    background - color : #1976d2;
}

    .nga- footer .hint:hover {
    opacity :  0.8 }
```

src/app/pages/general/home/home.component.html

```
<div class="container py-5">
  <div class="row">
    <div class="col-12 col-sm-12 col-md-12 col-lg-3 col-xl-3 text-center mb-2">
      <h1 class="h5">
        <i class="fas fa-laptop fa-lg me-2 text-primary"></i>
        {{ name }}
        <i class="fas fa-mobile-alt fa-lg ms-2 text-primary"></i>
      </h1>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-danger mb-2">
      <h2 class="h5">
        {{ version }}<i class="fab fa-angular fa-lg ms-2"></i>
      </h2>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-primary mb-2">
      <h2 class="h5">
        {{ bootstrap }}<i class="fab fa-bootstrap fa-lg ms-2"></i>
      </h2>
    </div>
    <div class="col-12 col-sm-12 col-md-12 col-lg-4 col-xl-3 text-center text-success mb-2">
      <h2 class="h5">
        {{ fontawesome }}<i class="fab fa-font-awesome-flag fa-lg ms-2"></i>
      </h2>
    </div>
  </div>
  <hr>
  <div class="row mb-2">
    <div class="col-md-12 text-center mb-4">
      <h3 class="h5">Features<i class="fas fa-list ms-2"></i></h3>
    </div>
  </div>
  <div class="row pt-2">
    <div class="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-2">
      <div class="card nga-card bg-light mb-3">
        <a routerLink="/bootstrap">
          <div class="card-header">
            <div class="row">
              <div class="col-10 col-xl-10">
                <h4 class="card-title h5">Bootstrap</h4>
              </div>
              <div class="col-2 col-xl-2">
                <i class="fab fa-bootstrap fa-lg text-primary"></i>
              </div>
            </div>
          </div>
          <div class="card-body">
            <p class="card-text">How to use Buttons, Alerts, Pagination, Tables, Collapses</p>
          </div>
        </a>
      </div>
    </div>
  </div>
</div>
```

src/app/pages/general/home/home.component.ts

```
import  { Component } from '@angular/core';
import  { CommonModule } from '@angular/common';
import  { RouterLink, RouterOutlet } from '@angular/router';

import  { environment } from '../../../../environments/environment';

@Component({
  selector: 'app-home',
  standalone: true,
  imports: [CommonModule, RouterLink, RouterOutlet],
  templateUrl: './home.component.html',
  styleUrl: './home.component.css'
})
 export class HomeComponent {
  name = environment.application.name;
  version = environment.application.version;
  bootstrap = environment.application.bootstrap;
  fontawesome = environment.application.fontawesome;
}
```

src/app/pages/general/home/home.component.css

```
.nga-card {
    display : block;
    background - color : rgba( 255 ,  255 ,  255 , . 8 );
        box-shadow:  0  1px 3px rgba( 0 ,  0 ,  0 , . 12 ),  0  1px 2px rgba( 0 ,  0 ,  0 , . 24 );
    border -radius: 2px;
transition: all .2s ease-in-out;
    cursor : pointer;
}

.nga-card:hover {
transform: translateY(-3px);
        box-shadow:  0  10px 20px rgba( 0 ,  0 ,  0 , . 19 ),  0  6px 6px rgba( 0 ,  0 ,  0 , . 23 );
}

    .nga-card  a  {
    color : black;
text-decoration: none;
}

    .nga-card  a :hover {
    color : #0d6efd;
text-decoration: none;
}
```

src/app/pages/general/home/home.component.spec.ts

```
import  { ComponentFixture, TestBed } from '@angular/core/testing';

import  { HomeComponent } from './home.component';
import  { RouterTestingModule } from '@angular/router/testing';

describe ('HomeComponent', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomeComponent>;

   beforeEach(async () => {
     await TestBed.configureTestingModule({
       imports: [HomeComponent, RouterTestingModule]
     })
       .compileComponents();

     fixture = TestBed.createComponent(HomeComponent);
     component = fixture.componentInstance;
     fixture.detectChanges();
   });

   it('should create', () => {
     expect(component).toBeTruthy();
   });
 });
```

src/environments/environments.ts

```
export  const environment = {
    application:
    {
      name: 'angular-bootstrap',
      version: 'Angular 18.0.2',
      bootstrap: 'Bootstrap 5.3.3',
      fontawesome: 'Font Awesome 6.5.2',
    }
  };
```

src/environments/environment.development.ts

```
export  const environment = {
    application:
    {
      name: 'angular-bootstrap',
      version: 'Angular 18.0.2',
      bootstrap: 'Bootstrap 5.3.3',
      fontawesome: 'Font Awesome 6.5.2',
    }
  };
```

src/app/app.component.spec.ts

```
import  { TestBed } from '@angular/core/testing';
import  { AppComponent } from './app.component';
import  { RouterTestingModule } from '@angular/router/testing';

describe ('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [AppComponent, RouterTestingModule],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have the 'angular-routing' title`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('angular-routing');
  });

});
```

src/app/app.component.ts

```
import  { Component, OnInit } from '@angular/core';
import  { CommonModule } from '@angular/common';
import  { RouterLink, RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [CommonModule, RouterLink, RouterOutlet],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent implements OnInit {
  title = 'angular-routing';
  footerUrl = 'https://www.ganatan.com';
  footerLink = 'www.ganatan.com';
  ngOnInit(): void {

    const navMain = document.getElementById('navbarCollapse');
    if (navMain) {
      navMain.onclick = function onClick() {
        if (navMain) {
          navMain.classList.remove("show");
        }
      }
    }
  }

}
```

src/app/pages/general/about/about.html

```
<div class="row">
  <div class="col-12 col-sm-12 col-md-3 col-lg-3 col-xl-3">
    <p class="text-center">about works!</p>
    <ul>
      <li><a routerLink="/about/experience">experience</a></li>
      <li><a routerLink="/about/skill">skill</a></li>
    </ul>
  </div>
  <div class="col-12 col-sm-12 col-md-9 col-lg-9 col-xl-9">
    <router-outlet></router-outlet>
  </div>
</div>
```

src/app/pages/general/contact/contact.html

```
<div class="row">
  <div class="col-12 col-sm-12 col-md-3 col-lg-3 col-xl-3">
    <p class="text-center">contact works!</p>
    <ul>
      <li><a routerLink="/contact/mailing">Mailing</a></li>
      <li><a routerLink="/contact/map">Map</a></li>
      <li><a routerLink="/contact/website">Website</a></li>
    </ul>
  </div>
  <div class="col-12 col-sm-12 col-md-9 col-lg-9 col-xl-9">
    <router-outlet></router-outlet>
  </div>
</div>
```

src/app/pages/general/signup/signup.html

```
<p class="text-center">signup works!</p>
```

src/app/pages/general/not-found/not-found.html

```
<p class="text-center">not-found works!</p>
```

Composants Bootstrap

The final project includes a module for testing Bootstrap components.

**The code being too long I do not indicate it in this tutorial.
However the complete source is available on Github.**

It is added via routing in the following files

app.routes.ts

```
{
  path: 'bootstrap',
  loadChildren: () => import('./pages/application/example-bootstrap/tutorial.module')
    .then(mod => mod.TutorialModule)
},
```

Tests

All that remains is to test the different Angular scripts.

```
# Development
npm run start
http://localhost:4200/

# Tests
npm run test

# Compilation
npm run build
```

Compilation errors

Compilation will generate an error.

The reason our CSS code exceeds a certain size recommended in the settings.

The settings are contained in the angular.json file.

These settings are restrictive in my opinion, you just need to modify them to remove the warning.

For the initial bundle the **maximumWarning** parameter defaults to **500kb**

We will use **maximumWarning** at **1mb**

For other components lThe **maximumWarning** parameter defaults to **2kb**

We will use **maximumWarning** at **4kb**

```
"production": {
  "budgets": [
    {
      "type": "initial",
      "maximumWarning": "1mb",
      "maximumError": "1mb"
    },
    {
      "type": "anyComponentStyle",
      "maximumWarning": "4kb",
      "maximumError": "4kb"
    }
  ],
```

Code Source

By following each of the tips I gave you in this guide you will end up with an Angular source code.

The source code obtained at **the end of this tutorial** is available on github https://github.com/ganatan/angular-react-bootstrap

If you like the source code, star it on github and off you go.

The next step will allow us to improve our structure.

The full tutorial is at the following address

- Step 5: Modules with Angular

The following steps will **get you a prototype application.**

- Step 6: Server Side Rendering with Angular
- Step 7: Progressive Web App with Angular
- Step 8: Search Engine Optimization with Angular
- Step 9: HttpClient with Angular

The following steps will help you improve this prototype

- Components with Angular
- Services with Angular
- Template Driven Forms with Angular
- Charts with Angular

This last step allows you **to obtain an example application**

- Build a Complete Web Application with Angular

The **source code for this final application** is available on GitHub

https://github.com/ganatan/angular-app