

## **Accelerated Angular Part 5: Styling with Bootstrap**

In <u>Part Four</u>, we introduced Angular Components and Directives. We then showed you how to implement them by extending the Tasks component to display a table with all the tasks assigned to the authenticated user. At this point, our code is functional, but our pages could be prettier. Let's face it: most developers, myself included, are not web designers, so we need some help from that department. The good news is that we don't have to be, and there are a number of available solutions. In this installment, we dive into one of those solutions and investigate how we can use Bootstrap to make our app look and feel more professional. The sample code for this installment is available on GitHub.

## **Key Concepts**

In this section, we provide background and context on CSS and Bootstrap.

#### **CSS**

<u>Cascading Style Sheets (CSS)</u> is a language for managing the appearance of web pages. It allows you to separate the presentation of those pages from its underlying (HTML) content. CSS defines selectors that map to individual HTML elements. For each element, you can set properties (attributes) and the values to set for the properties. In the following example, we put the color and weight of a top-level heading (H1) element.

```
h1 {
    color:red;
    font-weight: bold;
}
```

In addition to mapping to HTML elements, you can create class selectors that apply to a group of elements. You can also target specific elements by creating ID selectors.

#### **Bootstrap**

<u>Bootstrap</u> is a popular, flexible, and easy-to-use CSS framework. It was created by Twitter and made available as an open-source project. Bootstrap was designed to operate with Javascript, and specific versions are available for popular frameworks, such as React, Angular, and Vue.js. We will be using <u>NG Bootstrap</u>.

#### **Installing Bootstrap**

You can install NG Bootstrap with the <u>Angular CLI ng add</u> command. In your IDE's integrated terminal, open the project folder and type:



## ng add @ng-bootstrap/ng-bootstrap

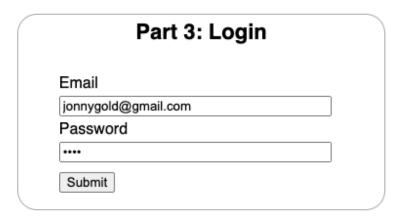
If this doesn't work, follow the detailed instructions on the NG Bootstrap website's <u>Getting Started</u> <u>page</u>. Now, all you need to do is update src/styles.scss. If you are using Angular 19, add the following to the top of the file

@use "bootstrap/scss/bootstrap";

If you are using an earlier Angular version, replace @use with @import:

## **Applying Bootstrap to the Login Page**

Before we installed Bootstrap, our login page looked like this:



After installing and referencing Bootstrap, it looks like this:



It looks similar to the original version, with minor changes to some elements. Now, let's comment out the original styles in src/styles.scss. After refreshing the page, it looks like this:



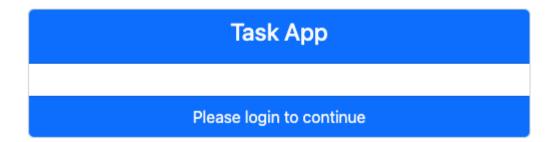
Part 3: Log	in
Email	
jonnygold@gmail.com	
Password	
••••	
Submit	

The box that surrounded the form disappeared, and the form was shoved to the left edge of the page. Now, let's replace our existing Login page with an improved Bootstrap version. Open src/app/login/login.component.html and replace the current contents of the file with the following:

```
<div class="container d-lg-flex">
<div class="mx-auto p-2" style="margin-top: 100px;width:40%"></div>
</div>
```

This adds a flexible container element that can expand and contract in relation to the browser window. Inside the container, we added a div element that will position the form elements on the page. Next, let's add a card component to hold our login form.

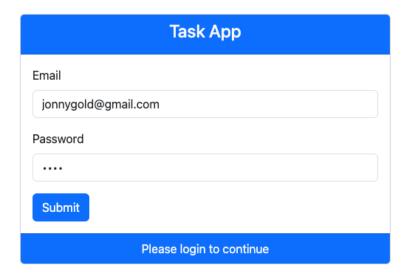
The Card's header includes a header and footer. The header displays the Application name, and the footer provides instructions.





In the card's body section, let's add our form with the appropriate Bootstrap CSS classes.

When we refresh the page, it looks much improved.



## **Updating the Tasks Page**

Before we applied Bootstrap, the Tasks page looked like this:



Using our updated Login page, let's log in and view our user's assigned tasks. The Tasks page now looks like this:



# **Tasks**

Name	Description	Added	Updated	Status
A task	My first task	2024-05-01T22:00:00.00	0Z 2024-05-01T14:16:01.119	Zdo
Another task	My second task	2024-05-01T22:00:00.00	0Z 2024-05-01T14:16:01.119:	Zdoing
task	My task	2024-05-01T22:00:00.00	0Z 2024-05-01T14:16:01.119:	Zdone
Yet another task	My fourth task	2024-05-01T22:00:00.00	0Z 2024-05-01T14:16:01.119	Zdo
New Task	My new task description	n 2024-05-01T22:00:00.00	0Z 2024-05-01T14:16:01.119	Zdo
Go to login				
New Task	•			

This is a slight improvement, but it's still not taking full advantage of Bootstrap.

Open src/app/tasks/tasks.component.html. Modify the opening table tag by adding the following Bootstrap CSS classes.

#### 

Now the table looks like this.

Tasks				
Name	Description	Added	Updated	Status
A task	My first task	2024-05-01T22:00:00.000Z	2024-05-01T14:16:01.119Z	do
Another task	My second task	2024-05-01T22:00:00.000Z	2024-05-01T14:16:01.119Z	doing
task	My task	2024-05-01T22:00:00.000Z	2024-05-01T14:16:01.119Z	done
Yet another task	My fourth task	2024-05-01T22:00:00.000Z	2024-05-01T14:16:01.119Z	do
New Task	My new task description	2024-05-01T22:00:00.000Z	2024-05-01T14:16:01.119Z	do
Go to login				

## **Conclusion and What's Next**

In this installment, showed you the basics of Cascading Style Sheets (CSS) and how to use them to change how our app looks. We then introduced Bootstrap and how we can use it to improve the appearance of our simple HTML pages. In the next installment, we will continue to use Bootstrap as we explain how to create and style reusable Angular components.