



## **Angular Level 2 Certification mini-project**

**[angulartraining.com](http://angulartraining.com)**

This document is online at  
<https://bit.ly/at-certification-stock>



## **GOAL: Build a small stock tracker application that displays the current stock price and news sentiment information**

For this project, we recommend using the online IDE StackBlitz and creating a new Angular project: <https://stackblitz.com/>. This will make it easy to submit your code as a single link once you're done with the project.

If you're unfamiliar with StackBlitz and want to use your own IDE and tools, we recommend using GitHub to host the project and publish a built version of your code using [GitHub pages](#).

To submit your work, you'll need to provide either the link to your StackBlitz project or a link to a Git repository that contains your code, along with a URL to test the app in a browser.

### **Important rules and notes - please read carefully:**

You have to **write the code yourself**. Submitting code already submitted by a friend or colleague is not allowed and will result in disqualification from the certification exam.

**Code quality and best practices matter.** Please refrain from using the **any** type in Typescript or having a single component in your application.

It should go without saying that **your application will be disqualified and your certification exam failed if the application has bugs, doesn't implement all of the features, or doesn't follow best practices.**

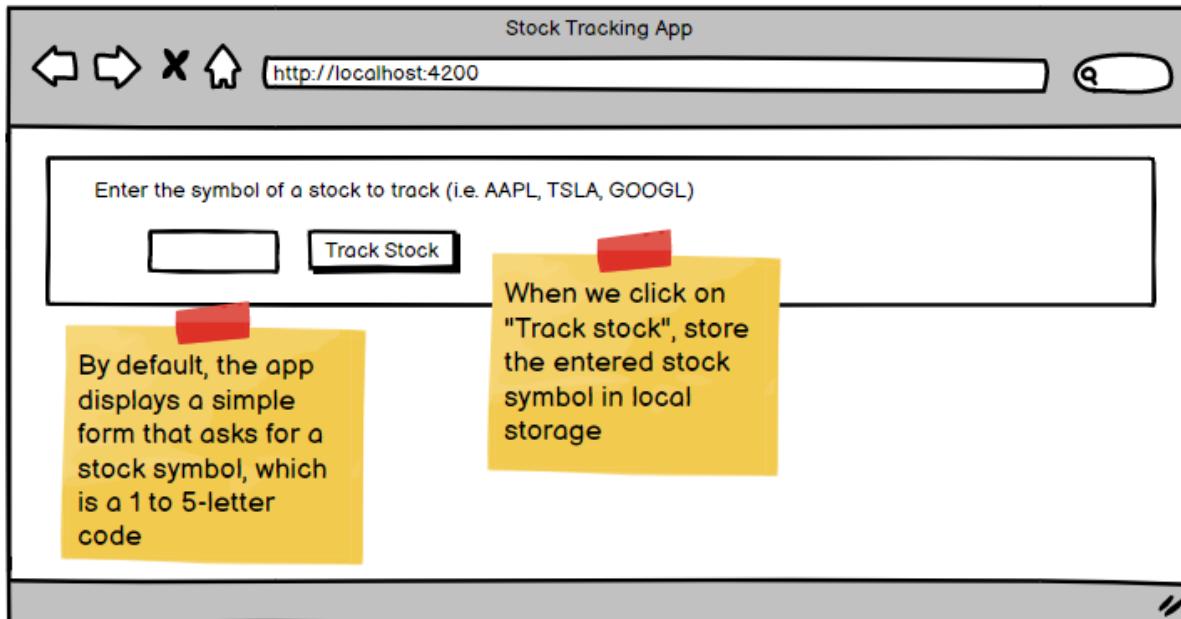
**Finally**, once your application is working and implemented following best practices (using types and proper component architecture), **submit your work** by going to <https://angulartraining.com/certification/level2-step2.html>

Please double-check your application. If any of the features listed in the following pages is not working, you will be disqualified.

If you have any doubts or questions, send an email to Alain at [contact@angulartraining.com](mailto:contact@angulartraining.com)

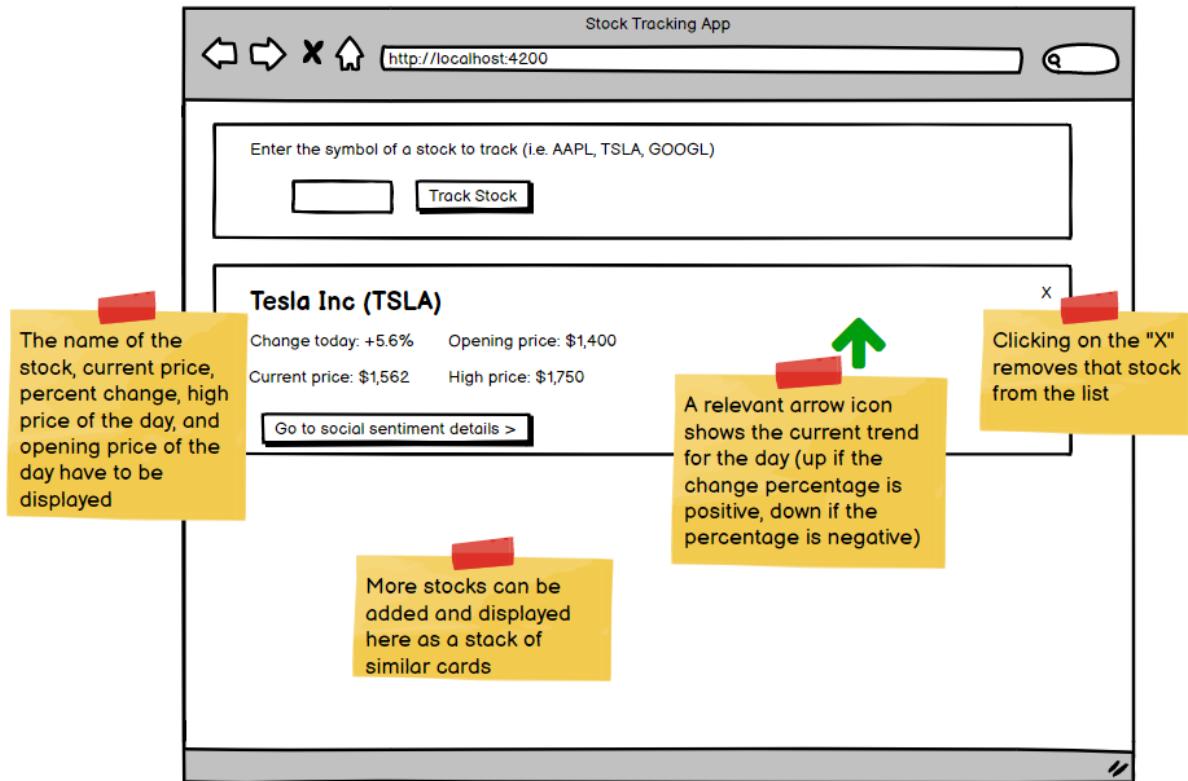


# STEP #1 - Implement the ability to enter a stock symbol and store all entered symbols in local storage



- Note that styling is not that important for this project, and the layout of your application can be different as long as the application works properly. This is an Angular certification, not a CSS or HTML certification.
- The text input should have an HTML attribute ID equal to **stockInput** and the button should have an ID equal to **trackBtn**
- You can see an example of implementation with all of the needed features in this video: <https://www.youtube.com/watch?v=GV5bjnwJEsM>

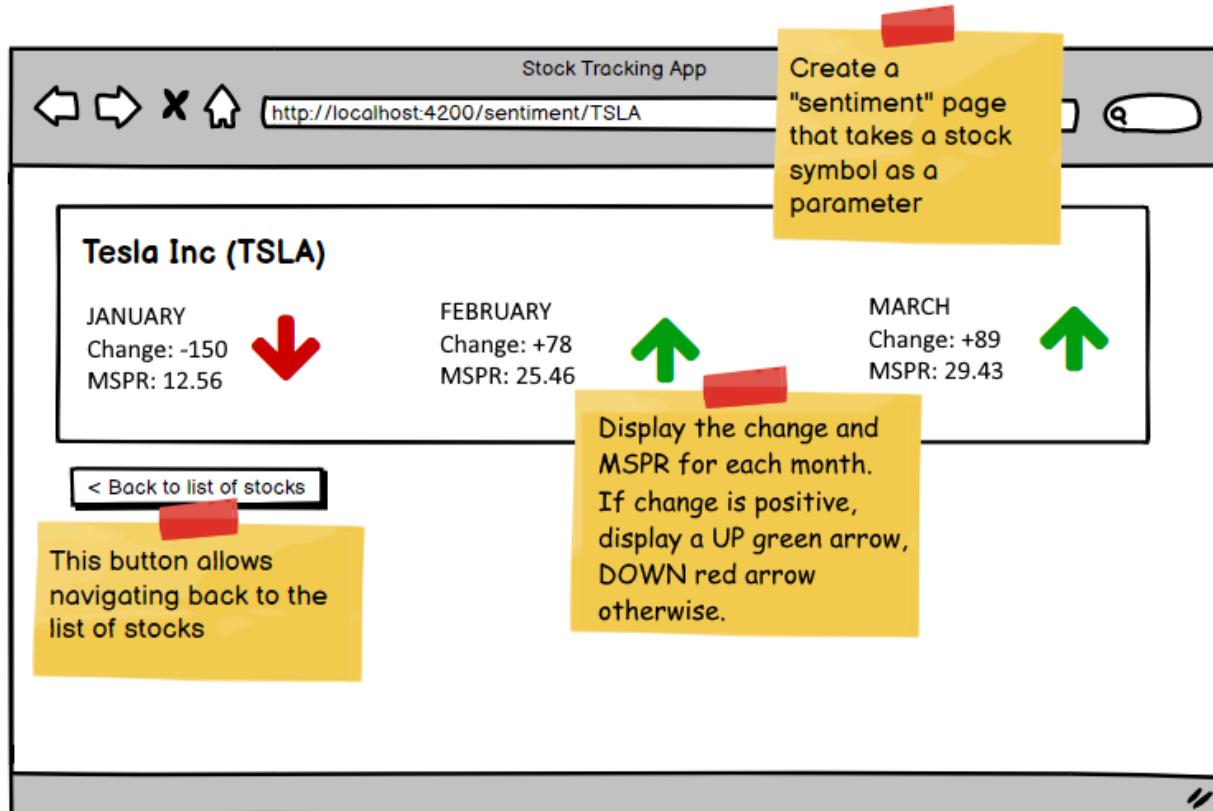
## STEP #2 - Display current quote data for all stocks that have been added



### Extra instructions:

- Use Finnhub API to get current stock quote data: <https://finnhub.io/docs/api/quote> and <https://finnhub.io/docs/api/symbol-search> to get the company name. You can borrow the following key to use the API: bu4f8kn48v6uehqj3cqg
- You can use Unicode arrows from this list or use your own images. Up to you: <https://unicode-table.com/en/sets/arrow-symbols/#heavy-arrows>
- The list of displayed stocks should be restored from local storage when the application is loaded in the browser.
- The "X" button (or link) should have an HTML attribute ID equal to **remove{stock}** (for instance, **removeTSLA** for TSLA, and **removeGOOG** for GOOG) and the sentiment details button should have an ID equal to **sentiment{stock}** (for instance, **sentimentTSLA** for TSLA, and **sentimentGOOG** for GOOG)

## STEP #3 - Add an insider sentiment page



### Extra instructions:

- Use the component router to add a `/sentiment/:symbol` URL that displays the sentiment information for the last 3 months. For each month, display a green arrow if the change was positive and a red arrow if negative. Also, display both the values of **change** and **mspr**. (<https://finnhub.io/docs/api/insider-sentiment>)
- Add a button on that page to navigate back to the main page (with an HTML attribute ID equal to **backBtn**)