Angular
Level 2 Certification
mini-project

angulartraining.com

This document is online at
https://bit.ly/at-cert-l2-cocktails

GOAL: Build a small application that displays cocktail recipes

You can see a video demo of the expected app here.

For this project, we want you to start from the code:
https://www.angulartraining.com/ng-cert-cocktails.zip
This will make it easier to start coding as the setup is already done and the API is included.

Once your code is completed, we recommend using Vercel or Netlify to host the project and publish a built version of your code.

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

Important rules and notes - please read carefully:

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

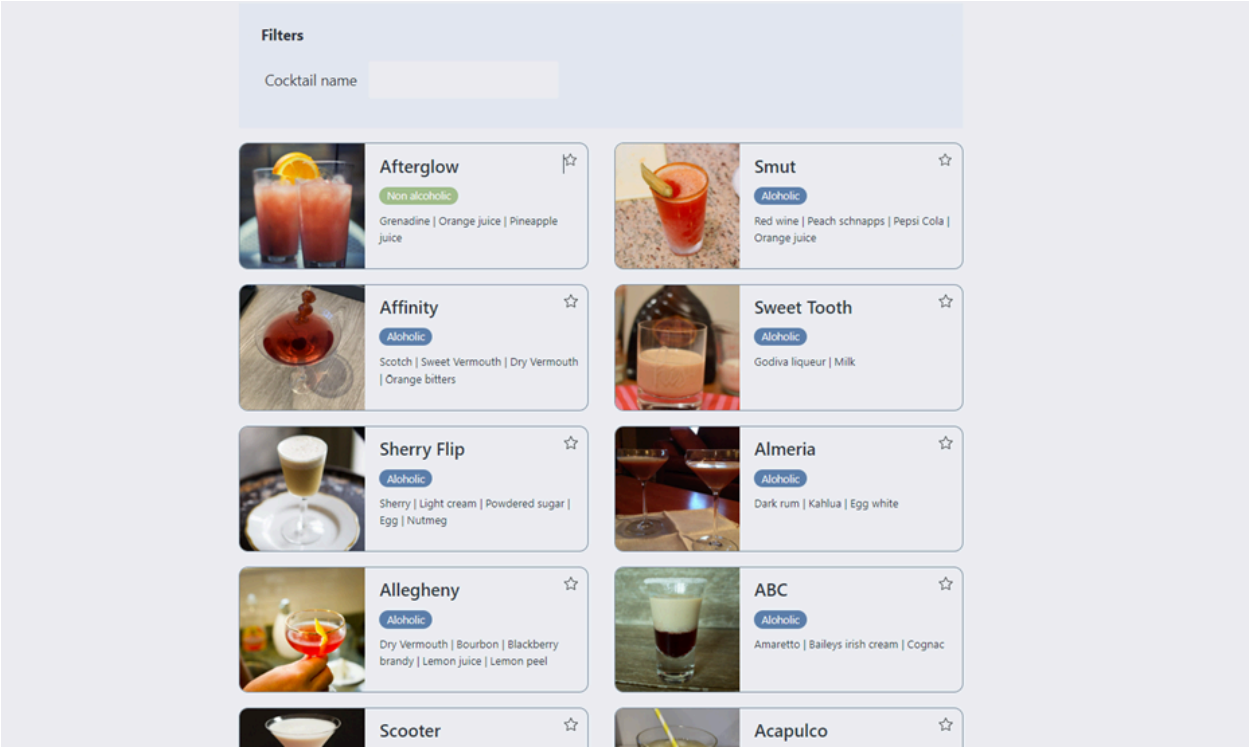
Code **quality and best practices matter**. Please do not use the **any** type in Typescript or have a single component in your application. Your code **MUST** use proper types for all variables, methods, parameters, etc. Failure to do so will automatically disqualify your submission.

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

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

If you have any doubts or questions, send an email to Alain at contact@angulartraining.com

STEP #1 - Implement the ability to list all cocktails



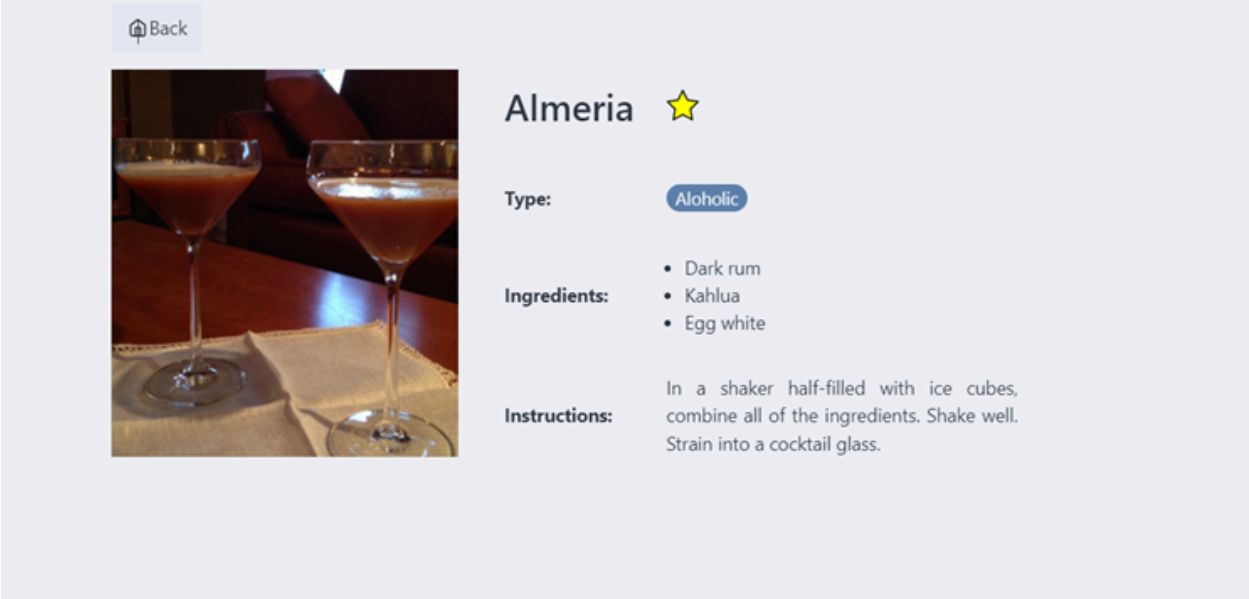
- Cocktails must be retrieved from the API included in the project and accessible at the **/cocktails** endpoint.
- The following properties from the API call should be displayed: name, isAlcoholic, imageUrl and ingredients.
- A text input should allow the user to filter the cocktail list. For instance, typing “ABC” in the input would filter the list and only show the cocktail called ABC.
- 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.

- To prepare for step #2, a star icon can be displayed for each job item with the **CSS class ‘icon-star’** and an ID of **“star-{cocktailId}”**

STEP #2 - Manage favorite cocktails

- Use the CSS class **'active'** to manage the star icon for a job selected as a favorite.
- The click on the star should be managed in order to add or remove the selected cocktail in a favorite list. Clicking on the star for a cocktail already favorited removes the cocktail from the favorites list. The star acts as a toggle button.
- Favorites **must be persisted in the browser** so a page refresh would not lose the favorites.

STEP #3 - Display the details of a selected cocktail



- A simple click on the cocktail title should redirect to cocktail details. Use the **Angular router** to implement this.
- Cocktail details must be retrieved from the API included in the project and accessible at the **/cocktails/:cocktailId** endpoint.
- On the details page, all the following properties should be displayed: name, isAlcoholic, imageUrl, ingredients and instructions.
- The user should be able to go back to the cocktail list using the back button at the top. Use this video as a reference if needed.
- The favorite toggle star button must be functional on this page as well.