

Static Restaurant Site

Description

During this iteration you will build a static website, meaning that the content, images etc does not change and there's no integration with the database. Your assignment in the coming days will then be to continue working on it, fetching all the courses from the API, making it dynamic.

So in short: **Design and build a site listing the courses for the restaurant**

See all courses, and all their data in [the database](#)

In this iteration you shouldn't display all courses, but choose 3-4, and use those as you design. And keep in mind that the rest of the courses should be added later. Make room for these additional courses, and keep in mind that you don't know exactly how many you'll have.

The front page, listing the courses is the only thing you need to build. You do **not** need to build subpages.

Remember to take into account (and present visually) that the courses can be

- Sold out
- Have a discount price
- Containing alcohol
- Allergens

Create your design so that all of these differences can be shown.

Hint: Make a class for each possibility that you add to the course

UI requirements:

Use CSS grid and optionally flexbox to layout the site.

It should be possible to click a course and read more (through a modal or something expanding).

Categories

The list of courses should be sorted into categories, you don't have to add all the courses in all the categories, as long as you have the basic skeleton for categories and courses.

Course details

How you show the details is up to you, but you should as a minimum show the following:

- Name
- Image
- Short Description
- Price
- Sold out status
- Discount price
- Vegetarian status

It's up to you to decide how you do this, in the case of whether the product is suitable for vegetarians, you can show the information for all products, or just for those that are suitable.

Hints and good ideas

Remember to make your design ready for the dynamic part, make the courses as similar as possible and make a mental note of the differences.

Use as many relevant classes as possible, a soldOut class for products that are... sold out, a vegetarian class for products that are suitable for vegetarians etc.

Add comments for your self, for instance, if there's an element that should be removed in some circumstances.

Delivery

This iteration is not a hand-in, but rather something you will continue working on after the hand-in.

Have it ready by Thursday morning.

Ressources

[the database](#)

[A zip file containing the course-images](#)

Feedback

We'll see a few, randomly chosen, in class Thursday morning