Poke

Design and implement a web application for managing take-away orders in a poke shop. Note that, for simplicity, there is no need for handling dates: all orders are placed on the same (unspecified) day.

The application needs to satisfy the following requirements.

Poke bowls can be of **three different sizes**: Regular, Medium, or Large (R, M, L). Each bowl has one **base** and can be ordered by choosing between different amounts of **proteins** and **ingredients**, according to the bowl size:

- Regular bowls include one protein and up to 4 ingredients (minimum 1, optionally repeated).
- Medium bowls include two proteins and up to 4 ingredients (minimum 1, optionally repeated).
- Large bowls include three proteins and up to 6 ingredients (minimum 1, optionally repeated).

Specifically users will have these options:

- Bases are rice, black rice, salad.
- Proteins are tuna, chicken, salmon, tofu.
- Ingredients are avocado, ananas, cashew nuts, kale, mango, peppers, corn, wakame, tomatoes, carrots, salad.

An order may contain **one or more bowls**. Several bowls with the same size, proteins, and ingredients can be selected by simply modifying a suitable counter while ordering: for example, it must be possible to order 3 R bowls with rice, salmon, avocado and kale without entering the same proteins and ingredients three times but just selecting 3 as the number of bowls.

Every day, the shop can make a **maximum number of poke bowls of each size**: respectively, 10 R, 8 M, 6 L. This information must be stored in the database, together with the prices of the poke, detailed later.

Any user (authenticated or not) can **freely browse** the availability of the three sizes of poke bowls, as well the list of bases, proteins and ingredients that can be used.

Authenticated users can make their own order **by creating a list of poke bowls**, specifying their characteristics in an interactive configuration page. All the interactions for bowl configuration must be handled client-side, except for availability checking of the available sizes, which must be done in real time when the order is complete and is being submitted, to avoid letting the user configure a number of bowls which are actually not available.

While the user configures the order, the price is updated in real time at every change (addition/deletion, modification of the proteins/ingredients, or the quantity).

The **total price** is the sum of the costs of all poke bowls in the order. Each bowl size (R, M, L) has a price which is stored in the database: R is 9 €, M is 11 €, L is 14 €.
Each selected *ingredient* in excess of the included quantity (i.e., 4 for R and M bowls, 6 for L bowls) increases by 20% the initial price of the bowl. A 10% discount on the total of the order is applied if more than 4 bowls are included in the order.

When the user submits the order for processing to the shop, the shop checks that enough poke bowls of the requested sizes are still available and confirms the order. Otherwise, a suitable message is shown (e.g., “not enough L poke bowls”), and the user is sent back to the configurator to fix their order.

Authenticated users can always check their list of **past orders**: the list should show a row for each order. The row shows the total number of poke bowls with the total paid price. A detailed view of the order must be shown (and hidden) when the row is clicked, and show the same information that was presented in the configurator.

The organization of these specifications in different screens (and possibly on different routes) is left to the students.

Important notes

- The price has to be computed on the client side and in real time while users proceed in their order. The final price has to be checked server side.
- The order should contain an optional text input where users can specify special requests or allergies.