

Mandatory exercise 2

This is a mandatory* exercise. Make sure you deliver it in Canvas before the deadline. The exercise must also be approved at the lab.

You may ask teachers and students for tips and advice, but in the end, it is **you** who must do the exercise so that you fully understand your solution and code. **At the approval of the exercise, you may be asked questions to verify that you fully understand your solution/code. If it is obvious that you don't understand your solution/code, the exercise will not be approved.**

Helseflora

Create a client for a webstore named *Helseflora* that sells medicinal plants. A server with information about the products (plants) in Helseflora already exists. The information you receive from the server is in JSON-format.

Note: The information you retrieve from the server is in Norwegian. Ask the teacher if you need an explanation of the content in English.

Task 1

The first version of the client should consist of three pages/views (see examples on the next page).

- a) **Category view:** This is the start-page. Here the user should see a list of plant-categories. Get the (JSON) information from: <https://sukkergris.no/plantcategories/>. When the user clicks on a category, he/she should see a list of all the plants in that category (see *Plant view* below)
- b) **Plant view:** The plants in a category should be shown with a picture, name, description and price. Get the plants (JSON) in a category from <https://sukkergris.no/plants/?category=hudplager>. The path to the images is <http://sukkergris.no/plantimages/small/> E.g., this is an example of an image: <http://sukkergris.no/plantimages/small/P1000903.png> It should be possible to go back to the category view from the plant view. When the user clicks on a plant in the list, he/she should see more details about the selected plant (see *Plant detail view* below)
- c) **Plant detail view:** The plant details should be shown with a bigger picture of the plant, the name, category, description, height, growth zone, fertilizer mix, planting advice and the price. Get the details of a plant (JSON) with a specific id from <https://sukkergris.no/plant/?id=5>. The path to the images is <http://sukkergris.no/plantimages/large/>. Note that the large images are in JPG-format instead of PNG. It should be possible to go back to the plant view from the plant detail view.

Show the views in different webpages. To go to another page, you can use `window.location.href = url`. If you want to navigate back from a page, you can use the back-button in the browser. If you want to create your own back-button, you can use `window.history.back()`;

Use *localStorage* to transfer information between the pages.

*Only task 1 is mandatory

Style the *layout* of the user interface so that it is similar to the examples on this page. You are free to select your own colors, but the layout must be similar:



a) Category view



b) Plants view



c) Plant detail view

Task 2 (not mandatory)

In the second version of the client, the user should be able to add a plant to a shopping-cart in the plant detail view. The user should also be able to go to the shopping cart by clicking on a cart-button that are shown in all three views (in task 1). Inside the shopping cart, the user should be able to do the following:

- Change the amount/number of plants
- Delete a plant from the cart.
- Empty the cart (delete all plants from the cart)

The total price of the plants should be displayed in the shopping-cart. The information in the shopping-cart should still be there when the user closes the app (browser) and opens the app again.

Store the shopping-cart in *localStorage*.



Plant detail view



Shopping cart view

Note. You don't need to send any data (e.g., an order with shopping-cart data) to the server.