Project description MM-206 – autumn 2022

In the project you are going to create a new version of the application you created in exercise 2 (Helseflora). The new application will have more functionality than the old version and it will use another server.

You will find the documentation for the server in *HelseFloraAPI_documentation.pdf.* Be aware that you must read the server documentation to get a full understanding of the different tasks in the project. Each project group is assigned a **group key** that gives the group access to the server (see the documentation for the server).

NB! The group must write a small project report with a summary of the functionality that has been implemented in the application – and known limitations/bugs.
Each member in the project group must also write a page about his/her contribution in the project. I.e., what each member has worked with in the project.

Evaluation

The project is evaluated based on:

- How much functionality has been implemented successfully? This also includes:
    - Does the application give sensible feedback to the user if something goes wrong (handling of errors)?
    - Is the application foolproof? I.e., the application should not break/crash because of invalid input or user interaction.
- Is the code divided into manageable and logical parts?
- Is the code rational and logical?
- Does the application make good use of the functionality on the server?
- Is the code easy to read, with sensible naming, formatting, and correct indentation, (no: innrykk)?
- Does the user interface have a good and functional design? Is responsive design used?
- Total impression.

You don't have to follow the order of the tasks below but be aware that some tasks require that you have done previous tasks.

Part 1 – Browsing

Task 1

Create the *startup-page* ("home-page") of the client-app. Requirements:

- The page should show all the plant categories. When you click on a category, a page with a list of all the plants in that category should be shown (see below for requirements for the *plantlist-page*).
- The page should have a search field and search-button. All the plants with a name that contains the search-word should be shown on the *plantlist-page*.
- The page should have a "Go to shoppingcart"-button. When you click on the button, a *shoppingcart-page* should be shown. This is done in part 3.

Task 2

Create the *plantlist-page*. Requirements:

- The page should show a list of plants based on the input/search from the *startup-page*. Each item in the list should show the plant with a picture, name, and price.
- If the input from the start-page results in no plants in the list, the page should show a message with "Your search did not match any plants".
- When you click on one of the plants in the list, a page with details about the plant should be shown (see below for requirements for the *detail-page*).
- The page should have a "Go to shoppingcart"-button. When you click on the button, a *shoppingcart-page* should be shown. This is done in part 3.
- It should be possible to go back to the *startup-page* by clicking a "home"-button/text/image.

Task 3

Create the *detail-page*. Requirements:

- The page should show details about the selected plant:
    - Show the plant with a bigger picture, the name, category, description and height.
    - If the plant has a discount, that should be clearly shown with the discount in percent (e.g., with a text overlaying the picture).
    - The number of plants in stock. If the plant isn't in stock, this should be clearly shown with the estimated delivery time.
    - If the customer rating for the plant is available, it should be shown as a number of stars (1 – 5).
- The page should have a "Buy this item"-button. When you click on the button, the plant should be added to a shoppingcart. This is done in part 3.
- The page should have a "Go to shoppingcart"-button. When you click on the button, a *shoppingcart-page* should be shown. This is done in part 3.
- It should be possible to go back to the *startup-page* by clicking a "home"-button/text/image.

Part 2 – Administration

We are going to create a client for administration of Helseflora. The UI in this part doesn't need to be adapted for small screens.

Task 4

Create an *admin login-page*: Requirements:

- It should be possible to log in as administrator. The credentials for the administrator are:
    - Username: **carlvonlinne@helseflora.no**
    - password: **busybee_taxonomy**

- After successful login, these buttons should be shown:
    - Plants
    - Orders
    - Users
    - User comments/reviews

  The buttons correspond to different pages you are going to create later. When you click on the first button, the page in Task 5 should be shown.

Task 5

Create an *admin plant-page* (or several pages) where it is possible to list, add and change the information about a plant. It should also be possible to delete a plant. Be aware that the server doesn't allow you to change or delete plants that you haven't added yourself.

Part 3 – Shopping and ordering

We continue with the client-application for the customers/users in part 1.

Task 6

Create a *shoppingcart-page*. Requirements:

- The page should show a list with all the plants in the shoppingcart. Each plant should be shown with the ID-number, name, number of items, price per plant and total price for that plant (number of items * price). You don't need to show a picture of the plant.
- If a plant is not in stock, the expected shipping date should be shown for the plant
- The page should also show the total price for all the items in the shoppingcart.
- It should be possible to change the number of items of a plant (initially it should be set to 1).
- It should be possible to delete a plant in the shoppingcart by clicking a "Delete"-button
- It should be possible to delete all the plants in the cart by clicking an "Empty shoppingcart"-button.
- It should be possible to go back to the *startup-page* by clicking a "home"-button/text/image.
- The page should have a "Proceed to checkout"-button. When you click on the button, a *checkout-page* should be shown (see below for requirements for the *checkout-page*).
- If you close the application (close the web browser), and then open it again, the items in the shoppingcart should still be there. I.e., store the data for the shoppingcart in localStorage.

Task 7

Create a *checkout-page* (or several pages). Requirements:

- The checkout-page is divided into three parts/steps:
  1. Type in customer information:
     - Name
     - Email
     - Address (street address, city, zip code, country)
     - Phone

  2. Select shipping method. Retrieve the different shipping methods and prices from the server.

  3. Show the total price including shipping together with a "Place order"-button. When you click on the button, an *order confirmation-page* should be shown (see below for requirements for the *order confirmation-page*).

     (In this version of the application, you can ignore any handling of payment)

- It should be possible to go back to the *startup-page* by clicking a "home"-button/text/image.

Task 8

Create an *order confirmation-page*. Requirements:

- Send the order to the server, I.e., the information in the shopping cart, shipping and customer information should be sent to the server.
- The “answer” from the server will be an order confirmation. Display the confirmation on the page, including:
    - Order number
    - Customer information (name, email, phone, and delivery address)
    - A list with the ordered plants. Each plant is shown with ID-number, name, number of items, price per plant and total price for that plant.
    - Shipping method
    - Estimated shipping date
    - Total price for the order – including shipping.
- It should be possible to go back to the *startup-page* by clicking a “home”-button/text/image.

Task 9

Go back to the client for administration and create an *Orders-page* where it is possible to list and delete orders. The page should be shown when you click on the second button mentioned in task 4.

Part 4 – Users and comments/plant reviews

Task 10

Add a “Create User”-button on the *startup-page*. When you click on the button, an *add user-page* should be shown where you can supply user-information. The information should be the same as the customer information in task 7 – in addition to a username, password, and picture.

Task 11

Add a “Login”-button on the *startup-page*. When you click on the button, a *login-page* should be shown where it is possible to login with username and password. When the user is logged in, the thumb picture of the user should be displayed in all other pages, e.g., in the upper right corner of the pages (but not in the administrator-application).

Task 12

Some of the plants on the server are only accessible for logged-in users. Change the *plantlist-page* and the *plant details-page* in Task 2 and 3, so that these plants are shown – together with the other plants – when you are logged in.

Task 13

Change the *checkout-page* in Task 7 so that the customer information is automatically filled in when you are logged in.

Task 14

Change the *plant details-page* in Task 3 so that it is possible to add a comment (review) for that plant – in addition to a rating (1 to 5 stars) – when you are logged in.

Task 15

Add a “Show comments”-button in the *plant details-page*. When you click on the button, a *plant comment-page* should be shown that lists up the comments and ratings for the selected plant. The button and the *plant comment-page* should also be available for users that are not logged in.

Task 16

When you click on the user-picture (in any of the pages), a *user settings-page* should be shown where it is possible to change the user information – and delete the current (logged-in) user. The page should also have a logout-button which logs the user out (deletes the authorization token in localStorage/sessionStorage). Show the startup-page after logout.

Part 6 – Extra functionality

Add more functionality to the web shop, suggestions:

- Add a button to the plant details-page. When you click on the button, a page that contains a guide for planting and fertilizing will be shown, for example:
  - Show the growth zone with a description
  - Show the fertilizer mix as a diagram, e.g., a pie-diagram
  - Show the height of the plant graphically, e.g., as a relative height next to a picture of a human.
- Use the *extra* information for plants to store the number or orders or total sale sum for a plant. Show this as statistics on the administrator page.
- Make it possible for users to list and delete his/her own comments.
- Use the yr.no API (the Norwegian weather service) and show graphically when it is favorable (if possible, at all) to plant a plant the next week – based on the minimum planting temperatures for day and night for a plant.
- Create a message board or Q&A site where it is possible for the users to write and publish messages or questions and answers. If you want to be a little “sinister”, you can add rating of users (see MeowMeowBeenz in the documentation for the server).