

Mandatory Assignment - Web development in CMS

The deadline for this exercise is **Wednesday, September 18, 08:00**.

For this **mandatory assignment** you should work on **master branch only**.

Preparation

1. Create a new repository on GitHub called **mandatory-cms2**
2. Follow the instructions that GitHub gives you; Create a local repository and add a remote or clone the newly created repository.

Submission

When you submit the exercise in PingPong, before the deadline, you will enter a link to your repository, such as:

<https://github.com/mygithubusername/mandatory-cms2>

The teacher will look in the **master branch**. If any commits are done to the branch after the deadline, the teacher will look at the last commit before the deadline.

Included in this assignment is an oral presentation. A schedule will be available on PingPong.

You will get one of the grades **G**, **VG** or **IG**.

Instructions

In this assignment you will create an Ecommerce site using Cockpit as a backend. Please refer to the previous exercise for more information about Cockpit and how to run it.

Please use Cockpit version 0.9.2.

Implementation

Your application should be an SPA with at least the following pages.

- A main page that displays a list of products with name, price and an image of the product. You need to implement pagination on this page and it should be possible to search for products by name. There should be a checkbox so the user can show only products that are in stock.
- A details page for every product that shows the name of the product, a description, how many units are in stock, the price of the product and a gallery of images. It should be possible to add the product to a shopping cart.

When adding products to the shopping cart it should be possible to select how many items are added.

The details page should also include a list of reviews with ratings. A review should include a title, a body and a rating between 1 and 5.

- A “shopping cart” page that displays a table of the products currently in the shopping cart. You should show the name, quantity and price of the products. You also need to display the total price.

If there are items in the shopping cart there should be a button or a link that takes the user to the “checkout page”.

- A checkout page with a form where the user can input a name and address.

When the form is submitted the shopping cart should be emptied and the user should be redirected to a page that says the order was created (actually creating the order is only required for the grade VG).

In order to do this you will have to create at least two collections in Cockpit. Below is a suggestion.

- Products (name, price, images, amount in stock, description)
- Reviews (title, body, rating)

The shopping cart should be implemented using “localStorage” so it isn’t emptied when the page is reloaded.

Additional requirements for VG

There are some additional for the grade VG. Two of these requirements involve adding new entries to Cockpit from the frontend application. It is possible to use the Cockpit API to add new entries using “/api/collections/save/{collectionname}”:

<https://getcockpit.com/documentation/api/collections>

For the grade VG a “review form” must be included on the product pages. When a review is submitted it should be added to the collection in Cockpit.

For the grade VG an “order” entry must be created in Cockpit when a user submits the form on the “checkout page”.

The “order” collection should include at least

- Name
- Address
- Total price
- A list of products and their respective quantities

In order to create a list of products and quantities, please refer to this part of the Cockpit documentation:

<https://github.com/agentejo/cockpit/wiki/Collection-Field-Types#nested-sets-within-repeaters>

Deployment

Deploying the application is not required but it is important that you are able to show a working application during the oral presentation.

Requirements

- The application should be an SPA written using React, Vue or Angular
- Cockpit should be used as a backend
- The application should implement routing
- The main page should show a list of products including name, price and an image
- The main page should have pagination
- It should be possible to search for products by name and filter to show only products that are in stock
- The product pages should show the name, description, price, quantity in stock and a collection of images
- The product pages should show a list of reviews
- It should be possible to add products to a shopping cart
- The shopping cart should be implemented using localStorage
- The checkout page should include a form where the user can input name and address
- When the user submits the checkout form the shopping cart should be emptied
- Use Collectionlinks in Cockpit to link reviews to products

The following requirements are only for the grade VG

- The product pages should include a form where the user can add new reviews. These reviews should include a title, a body and a rating between 1 and 5

- When the user submits the checkout form a new “order” entry should be added to Cockpit. This entry should include the name and address of the user, the total price and a list of products and their respective quantities.