Semester Project 2

Goal

To create an e-commerce website that has both customer-facing and admin sections. Both sections should be responsive and the website will be populated by a Strapi API supplied by Noroff.

Brief

You are to build an e-commerce website. You can choose the theme of your website. It should follow the site architecture described below.

Design your website using your favourite tool. You will need to find a suitable logo. If you decide to create a logo yourself, do not spend too much time on it.

You must apply all that you have learned in your studies so far. The site must have a good user experience and UI design, following today's trends and design patterns.

Build a frontend with home, product list, product detail and cart pages.

Build admin pages to create, update and delete products.

The website must be responsive on all devices.

Building a checkout and payment system is not a part of the project.

Level 1 is required.

Level 2 is optional.

Strapi API

The Strapi project can be found here.

Install the npm dependencies and then run npm run develop:

npm install npm run develop

You can add, edit and delete products in the API and change the Home single type content, but do not add any new properties or endpoints to the API.

If you do want to add custom properties, endpoints or additional functionality you can create your own API which must be publicly hosted. Please see the Level 2 section.

User credentials

- email: admin@admin.com
- username: admin
- password: Pass1234

Level 1 (required)

Customer-facing pages

Home page

The home page must include:

- A hero banner with an image that is uploaded to Strapi. You can find this in the Home single type in the provided Strapi project.
- A list of featured products. On Strapi products can be marked as 'featured'. When a product is marked as 'featured' it should be displayed on the homepage. You can find the products in the Products collection type.

Products page

The products page must include:

- A list of all products added to Strapi. Each product must display its title, price and image. The product should link to its products detail page.
- A search text box. When searching (filtering), only the products that include the searched text in their title or description should be listed.

Product details page

This page is reached by a user clicking on a product on the product list page. The product details page must include:

- title
- description
- image
- price
- an add to cart button. This will toggle the product in and out of a cart array stored in local storage.

Cart/Basket page

The cart/basket page must display a list of all products added to the cart. Load the items that have been added to local storage and display them on the page. If the cart is empty display a message indicating this.

Each product in the cart must display:

- title
- price
- a link to the product view page
- image

After the list of products, display the total price of all the products in the cart.

Note: the cart page is not a checkout page. No payments or user details are required to be taken.

Admin section

The admin section (apart from the log in form) must only be accessible to logged in admin users and must include the following features.

Login/Logout

Create an admin login form that allows administrator users to login. Use local storage to keep the user logged in.

When logged in, display a logout button in the layout that logs the user out. Logging out should not clear the cart.

Add/edit products

Create form(s) that allow products to be added and edited. The form must allow the user to toggle whether a product is featured.

Product images

For adding/editing product images use either of these 2 methods:

- Use a file upload field to upload images to Strapi, or
- Use a text input that allows a URL to be entered. This allows an image from an external URL to be used as the product image.

You can find the fields for each in Strapi. Use only one to display a product image.

Delete existing product

Allow products to be deleted. Before a product is deleted you must display a confirmation dialog. The product should only be deleted if the user confirms.

Level 2 (optional)

Create your own API for the site. You can use any backend stack (e.g. Strapi, Wordpress REST API, Firebase, etc) but it MUST be publicly hosted on a server.

The API must include all the functionality in the provided Strapi API and you can add any additional functionality.

It MUST be publicly hosted and accessible, you cannot submit a custom API with your frontend code.

We have created a one-click Strapi deployment to Heroku for this project. All you need is a Heroku account and a Cloudinary account, both are free.

FAQ

- You can use CSS libraries like Bootstrap.
- Use either Sass or Styled Components for your styles. Using BEM is optional but using proper class names is important.
- Use vanilla (regular) JavaScript for the project and split your code up using modules (imports/exports).
- You can use small JS libraries to perform tasks such as formatting dates with Moment.js

Marking criteria

- All functionality in Level 1 should be implemented.
- The design should be coherent and provide a good user experience.
- All the customer-facing and admin pages must be fully responsive.
- Use appropriate names for Sass classes and folders.

- All code should be properly formatted and arranged with sensible variable and function names.
- Use modules (imports/exports) to organise your code.

Rules

- Copying and sharing of any code will result in your project being failed.

Report

You don’t have to write an academic-style report, just discuss the project in the headings provided in the template.

500 to 1000 words in total.

The report template can be found here.

Submission

- Create a folder called your-name-semester-project-2, e.g. mary-smith-semester-project-2.
- Add your project files and report to the folder. DO NOT submit the Strapi API files or custom API code if you created one. If you have a node_modules folder in your project make sure to exclude it from your submission.
- Zip the folder and submit the zip file.

NOTATER

Referanse : https://www.ipcc.ch/site/assets/uploads/sites/3/2019/10/img-placeholder.png

Bildet er tatt av <a href="https://pixabay.com/no/users/clker-free-vector-images-3736/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=36730">Clker-Free-Vector-Images</a> fra <a href="https://pixabay.com/no/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=36730">Pixabay</a>

Bildet er tatt av <a href="https://pixabay.com/no/users/elisariva-1348268/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=1991839">ElisaRiva</a> fra <a href="https://pixabay.com/no/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=1991839">Pixabay</a>