

Web developer Exam

June 2023

Leospa Beauty Spa



Situation description

You are employed in a web agency that develops dynamic web solutions. You are part of a development team consisting of a design department and a code department. The agency has received an order for a website from the company Leospa Beauty Spa. The layout and the general structure of the site have already been determined and approved by the customer. Your main task will be to code the site's functionality.

Task description

Leospa Beauty Spa wishes to have a more visible profile on the Internet. Therefore, the company wants to have a website built that can be included in their overall marketing strategy.

The purpose of the site is to present the company's brand matters in a clear way, as well as to give users the opportunity to easily and clearly find information. In addition, they also want a new page to present their "services" in a professional way and a page for "features" which should contain any future events that Leospa holds.

Delivery

The site must be delivered as a zip file - as well as on a local or web-based server solution of your choice.

Remember, you must NOT edit your original version of the project after submission – if you want to prepare/practice something for the oral exam, make a copy of the site so that you can take the exam with the original unedited project.

In addition, according to the GDPR legislation, you must give your consent to the examiner and censor being able to review the task solution on your computer in connection with the oral exam.

You will receive further information about delivery when the delivery time approaches.

Technical requirements

- General **design files and fonts** are provided with the assignment.

- As a starting point, you must **follow the design in the setup**, but the customer is receptive for changes in the design, if these can be argued in favor of the customer.
- The site must at least be built with **HTML, CSS and JavaScript** - for example React and preferably with the use of Sass/Less (or similar), TypeScript etc.
- The site must meet the usual standards within **semantics**, correct code, accessibility and search engine optimization/SEO. Also remember favicons/manifest.
- The website must be **responsive** and adapted to relevant screen sizes – including ordinary computer screens (desktop/laptop), tablets, mobile phones, etc.
- The site should contain **error handling** for all relevant errors, so that the user receives an error message which takes into account the site's content.
- There must be **validation** on all forms/input elements.

Web API

The site's data/content can be retrieved via the provided API/backend project.

Setup and startup of the backend/API can be seen from descriptions in the "_INSTALLATION" folder in the root of the delivered backend project.

Where necessary, you must **adjust the size of the images** to the site's layout.

It is important that you **retrieve data (texts, images, etc.) from the provided API**, where these are available in the API.

In the aforementioned folder there are json files for import into the Mongo database + for your local Postman.

There is also an external Postman documentation here: [https://](https://documenter.getpostman.com/view/12464673/TzCQb77e)

documenter.getpostman.com/view/12464673/TzCQb77e

Pictures

You have been given graphics and photos together with the assignment.

Images to be retrieved as part of API data exist with the following endpoints:

- <http://localhost:5029/images/recommendation/paavej.jpg>
- <http://localhost:5029/images/treatment/paavej.jpg>
- <http://localhost:5029/images/team/paavej.jpg>

Port number 5029 assumes that you do not change the setup in the backend.

Report

You must prepare a project report together with the task solution.

The project report must contain at least the following:

- Assessment of your **own efforts** and completion of the assignment process, comments on your schedule/estimate of time etc.
- **Argumentation for choices** you have made during the solution of the task
- If you have **changed/added something to** the "backend"/API project, describe what and why in the report.
- Explanation of **the origin of the various code elements** in the task solution, which you may have not developed themselves (link to the origin of codes for e.g. slider, gallery, "smoothscroll", etc.)
- Specifying url addresses (e.g. to GitHub), usernames and passwords that are necessary for the teacher and censor to see the assignment
- A description of any special points for assessment

Annex to the report:

- Your **schedule**
- Possibly. **photos, screen prints** , etc. which documents your schedule, your work process, etc.

- Other things that you find relevant to attach

The cover of the report must state

- Name of the assignment, your full name and team number/class (webith120)
- URL to the site, if it is online
- Any relevant usernames and passwords

The scope of the report must be **a maximum of 5 A4 pages** excluding appendices.

The project report is submitted as a pdf file, and **is submitted in the root of the finished project/website** - and is named **Report.pdf**

Requirements specification

Here follows an overview of the requirements for the site's content.

A jpg file with a layout that supports these requirements is provided with the assignment.

Elements in the layout that are not mentioned in the text are basically also part of the assignment's requirements (ask if you are in doubt). Where tasks/elements are not described or shown in detail, you are free to choose how you want to solve these elements – but you must follow the design style and consider the needs of the customer/users in your solutions. **You must be able to argue for your free choices.**

General requirements

- The site is built in standard dimensions according to the layout
- The site must be **responsive**

Also read the section "**Technical requirements**" above.

Fixed elements on all sides

The following elements must be included on all pages (Front page, Services, Features, etc.):

- **Logo**
- **Navigation** Menu
- **Footer** with contact information (retrieved from API)

On the admin pages, you adapt navigation etc. so that it matches the purpose of the admin area.

Tasks - site content

"Home"

- **Header**, navbar, etc.
- **Hero** (with link to fancybox/popup/modal etc. with display of Hero's video clip)
 - o Show only the Hero that has **show=true** in the API
- **About** (with link to the "Features" page)
- **Treatments** (procedures/services/treatments)
 - o First **4 random treatments/procedures** (photo only) (create a script yourself that selects 4 random ones from "all treatments") - with link to the "Services" page
 - o Then **3 treatments/procedures** (e.g. the first 3 in the API), with both photo and text, where **the text is truncated (...)**, so that the box is a reasonable size (see the layout) - with a link to the "Services" page .
- Customer feedback/recommendation (slider)
 - o Show **3 latest recommendations** (use the appropriate API call)
 - o Each recommendation **links to a page/overview** with all recommendations (not shown on the layout - follow the design style)
- **Team** (show all the therapists)

- **Book an appointment** (must post into the API)
- **Footer – with contact information data** (not shown on the layout) from the API (footer)

"Services" (treatments/procedures)

- Must be **linked to from** the nav bar/"Service" AND when you click on one of the 7 treatments/procedures (4 random + 3 selected) on the front page.
- **You** must set up this page yourself - remember to follow the design profile.
- Must show **ALL treatments/procedures** (the treatments the place offers)
- Feel free to show some of the text with the option to "Read more" which then shows a pop up/modal, fold-out text etc., where the entire text appears.

"Features"

- Must be **linked to from** the navbar/"Feature" AND when you press the button in the section "About..." on the front page.
- **You must set up and design** this page yourself, and you must show that you can follow the design profile.
- Set up **one or more fictitious SPA/beauty events** on this page.
The content of the events is not important (could be e.g. evenings with lectures, presentation of beauty products, running competitions with prizes... just come up with something) – focus on the design/style.
- Remember it must represent **the design profile**.
- Must **not contain** extracts from the API - use provided/images and some lorem ipsum/dummy text

Admin part

An admin area (possibly protected behind login – see the optional tasks) from which it is possible to manage (**create/correct/delete**) data for min. following:

- **Treatment** (create, delete)
 - **formatting of the "content" text** (bold, line breaks, etc.) *e.g. using an **editor** (CKEditor etc.)*
- **About** (right) –
 - formatting of the "content" text** (bold, line breaks, etc.) *e.g. using an **editor** (CKEditor etc.)*
- **Hero** (create, correct, delete)
 - + allow the administrator to **choose which Hero should have **show=true**** (= the Hero that appears on the front page) - see the API's PATCH request
- **Footer** (right)

The API has the necessary request methods (POST, PUT, DELETE, etc.) – if something is missing or there are errors in the API, please point it out/ask.

Optional assignments

You must choose **a minimum of 2 of the following optional tasks**. However, you are welcome to choose more if other/more important requirements are met or abandoned (check the priority list!):

1. Login

Make **login protection of the administration part** so that only users/users created with e-mail and password can gain access. The following user has been created in the system:

hn@leospabeauty.dk with password: **admin123**

You can create more users yourself with Postman or login.rest (in the backend)

2. Sign up for the newsletter

Users must be able to **sign up and unsubscribe from Leospa's newsletter** - you have free hands to design where and how. There are API endpoints for both opt-in and opt-out.

In the administration part, the administrator must be able to see a list/page with all registrations - for example with the option to delete registrations.

4. Administration of "appointments"

A page in the admin section of **all customers' upcoming appointments/agreements**, which the clinic's staff can use as an overview of the coming days' orders (possibly for printing)

The overview must be divided into date order/date headings, so that it is clear what the agreements/appointments are **for each day**.

Only show appointments from "today" onwards = exclude old appointments/appointments.

And for each appointment, **it must be clearly shown** which treatment has been ordered (e.g. highlighted with different color codes), the customer's contact details, notes, etc.

Expand if necessary with extra functionality, so that the clinic can, for example, enter two dates and see all appointments in the relevant period – or appointments of a single date, etc.

5. Administration of "users"

Only users/users with **admin=true** (see the API's user endpoints) must have access to one page in the admin section, where he/she can create, edit and delete users.

Users CAN, for example, be the therapists/team - but do not have to be. Just think functionality here.

6. Additional validation of "Make an appointment"

Expand validation of date and time, so that you **can only book appointments from the date "tomorrow" onwards**. And no more than **6 months** into the future. And only **during opening hours**

(Monday to Friday 8 am to 4 pm). Either make it impossible to choose otherwise and/or give an error message.

7. Customers can see their own appointments

Users/customers must be able to **see their own appointments** by sending/posting the telephone number (**phone**) and **email**. You have free hands to design where and how.

There are API endpoints to handle such a request.

Nix – of course it doesn't comply with the GDPR rules, but it's good that you're paying attention here :-)

But ignore that and just focus on showing that you can set up the page and the feature.

Additional

If you have solved at least 2 of the optional tasks and met the other requirements of the task, you have free rein to expand with further functions - **but they must be relevant and you must be able to argue for your choices.**

REMEMBER to note in the report which optional tasks you have solved.

Priority list

1. General setup of **responsive design** (remember SEO and accessibility)
2. The site must appear with a high **degree of skill** that matches the design proposal - built with html and css etc.
3. Integration of **text and images extracted from the API**
4. Book an **appointment**
5. **The Services/treatments** page
6. **The Features** page

7. Admin part with create/correct/delete text/images from the API - **see the section on "Admin part"** above, where the requirements are described.

8. Optional tasks

Delivery

The deadline for submission is Saturday 17 June 2023 at 14:45 + time for setting up the API.

Before the deadline, you must hand in a zip file with the following on the shared drive:

- Folder with your finished web project (WITHOUT node_modules!!!)
- Report.pdf – placed in the root of the web project folder
- "Backend"/API if you have changed in this (if you have not changed anything, you should NOT hand over the backend).

Delivered in a folder on VIDonline by agreement when delivery is approaching.

Good luck with the task **yyyy ...**