Project Exam 1

Goal

To put into practice the skills learned over your first year of studies.

@2021 Lasse Hægland using Marp

Brief

You have been tasked with creating a blog site. You can choose the design and topics covered of the blog, but it should have at least the following pages:

- Home page
- About page
- List of blog posts
- Blog post specific pages
- Contact page.

Home Page

The home page should have a ‘Latest Posts’ section which uses a carousel (slider) for users to click to view more posts.

For example, by default the user can see four posts, then they can click an arrow on the right to view the next four posts, and click it again to view the next four posts.

The user can also click back to view results they had previously seen.

Blog Page

The blog posts page should show the first 10 blogs, and the user should click to view more results which then show underneath the first 10 blogs.

Blog Specific Page

The content of the blog specific page should be dynamically built using a query string parameter based on whatever link the user clicked.

The title of the blog specific page should change based on the blog that has been clicked on e.g. “My Blog | An Article I Wrote”.

If images on the blog post page are clicked, a modal should appear giving the user a bigger view of that image. Clicking outside the image should hide the modal.

Contact page.

Create a contact us page, there should be 4 textboxes on this page.

- Name (Should be more than 5 characters long)
- Email address (Must be a valid email address)
- Subject (Should be more than 15 characters long)
- Message content (Should be more than 25 characters long)

Please use JavaScript for validation, show error messages if the values in the textboxes do not meet the requirements.

WordPress

The content for your website will be stored on a WordPress installation used as a Headless CMS.

It’s important to note that we are only using WordPress to provide an API and add content for the blog.

You should not submit a link to a WordPress site, but build your website using HTML, CSS and JavaScript and making a call to the WordPress REST API to fetch the data.

The project has two aspects:

- API from your WordPress installation
- Your website built with HTML, CSS and JavaScript

You will need to add at least 12 blogs for your website. You can use lorem ipsum for paragraphs if you need, but headings, images etc. should all make sense.

> Note that this is an exam, and therefore tutor support will be limited as per the study plan.

Level 1 Process

1. Decide on the theme for the blog you're going to make
2. Create a prototype of the website
3. Install WordPress on your web host and add the blogs on the admin panel.
4. Create a GitHub repo and deploy to Netlify

   > Note: By following the GitHub Classroom link, a repository will be created for you to use.

5. Build your website using HTML, CSS and JavaScript making a call to the WordPress REST API to fetch your data.

Level 1 Process, cont.

6. Install Hotjar on your website.
7. Ask users to test your website, and adjust based on their feedback and any insights from Hotjar.
8. Write a report documenting your project.
9. Submit your report as a PDF and a link to both your Netlify deployment and your GitHub repo.

Level 2 Process (optional)

1. You can try adding a sort, filter, or search to the blog posts page allowing users to find the blog post more easily that they’re looking for.
2. Post the data from the contact form to WordPress so you have the details saved.
3. Allow users to submit comments on a blog post, and post this data to WordPress

PLAGIARISM WARNING

Do not use any code in your project that you haven't written. You can't copy-paste from websites like W3Schools, StackOverflow or other resources online.

You can use resources to get ideas for how to solve problems (and these must be credited in your report), but you can't simply copy code and paste it into your assignment.

Plagiarism will result in failing the assignment and you could be reported to the national reporting system and stopped from any tertiary studies for 2 years.

For more information, you can read about plagiarism here.

Checklist

✅ Mobile responsive and looks good on all screen sizes (not just one mobile screen and one desktop screen). Meta viewport in the head of the document.

✅ The HTML is neat and semantic, and the CSS is concise and styles aren't duplicated in media queries.

✅ Each page has a unique title, one unique h1, and meta description.

✅ Images are below 200kb and have alt text.

Checklist, cont.

✅ The site looks good and there's a class in the navigation telling the user which page they're on

✅ Text lines are kept short. Here's an example of how you can do that using max-width https://codepen.io/noroff-education/pen/VwYpweQ

✅ The colours have good contrast, the text is easy to read and the site is easy for user's to navigate.

✅ The report includes planning, and covers why decisions were made and how the process was to create site.

Marking guide (aka rubric)

Appealing design (30%): Masters relevant tools, techniques and expressions for developing dynamic web solutions in accordance with guidelines for universal design

Technically efficient (40%): Can apply vocational knowledge to practical problems within the field of study with focus on the development of dynamic web solutions

WCAG guidelines, content management and SEO (20%): Masters relevant tools, techniques and expressions for developing dynamic web solutions in accordance with guidelines for universal design

Report (10%): Can study and document his/her own project and identify issues and what measures need to be implemented

Dates

Delivery: **Sunday May 30th at 23:59**

Guidance: **Every Wednesday 0900-1500**

As soon as possible (preferably by 1st Guidance): Please provide link to github-repo (or github user name) and a link to the Netlify where your final blog is to live, which gets the posts from the WP site (proof-of-concept).

Links

- Installing Multiple WordPress Instances (see variant 2)
- Wordpress REST API Handbook
  - Pagination
  - Reference: Posts
  - Reference: Comments