CS 203, Emmanuelle Renauld
Assignment 10 (Final lab): a blog

This lab is your final lab. You have more than two weeks to complete it. Must be submitted before December 4th at midnight. This lab is worth 13 points.

Today we will create a new page; a blog. Read the whole assignment before starting coding! You have mandatory aspects (worth 7 points) and optional aspects (worth 6 points) to implement. You must choose a total of 6 points out of the optional items. If you do more than 6, you will receive a maximum score of 6. At submission time, please tell me which optional items you have coded.

Additional instruction 1. To make the code easier to review, please:

- Put all Javascript code in a separate page.
- You may include separate PHP pages if it makes the code lighter.

Additional instruction 2. For every PHP, JavaScript, or CSS file, include comments explaining why you wrote the code this way and what each major block does. Generic comments will not be accepted. If you reuse code from internet, make sure you understand what it does.

Additional instruction 3. If you reuse code from internet, add a reference (in comments), or this could be considered plagiarism.

Additional instruction 4. Add a GitHub commit for each new element that you add or each fix that you manage.

Additional instruction 5. As much as possible, keep the code MVC-like (model-view-controller). Separate data handling (model) and presentation (view).

Part 0. Setting up your GitHub branch

At the beginning of every lab, you should set up your GitHub repo. Follow instructions in lab 0.

1. Fix my comments on your previous pull request.
2. IF YOU HAVE NOT RECEIVED A COMMENT SAYING YOUR PULL REQUEST IS GOOD FROM LAST LAB, COME SEE ME. I MAY NOT HAVE RECEIVED YOUR LAB.
3. If you have received an ok, merge your previous pull request online.
4. On your computer, go to your master branch and pull the changes.
5. You may delete last week's branch
6. Create a new branch lab10 and go on it.

Note. From now on, you do not need an active GitHub page anymore. You may deactivate it if you wish. I will now see your final website using Osiris.

But you will still need to do a pull request: it is easier for me to give comments there. Else, I would have to load your files from Osiris, save them as a PDF, comment on the PDF, which is much less efficient.

Mandatory aspects (7 pts)

1. 1 pt: Create a **new page** for your blog and add it in your navigation menu. Set it like your other pages, with a menu and a footer. Add a "hero" section explaining the purpose of your blog in an attractive way. The blog must contain interesting text, blog-like. More than one sentence per blog item! The page must look nice. See examples below, or see optional items 1a) and 1b) below.
   https://www.w3schools.com/howto/howto_css_blog_layout.asp
   Using grids:
   https://www.w3schools.com/w3css/tryit.asp?filename=tryw3css_templates_blog&stacked=h

2. 1 pt: Add a main section for the blog posts, and an **aside section** listing all posts. The list items must be hyperlinks pointing to the blog item. Hint: to point to a section, use its ID. <a href="#id">. Hint 2: For a smooth effect, you could try the scroll-behaviour smooth attribute: https://www.w3schools.com/howto/howto_css_smooth_scroll.asp#section1

3. 1 pt: The blog posts (i.e. the raw text) must be saved in a **json file** in your directory (in /srv/http/home/username). For instance, each blog item could be saved as a date, a title and a list of paragraphs. You are free to organize your file as you wish! The page must

load all posts from the json file and display them on the page with PHP. You will probably need a for loop. Example below. The “blog1”, “blog2” keys act as id to identify each post.

```
{
    "blog1": {
        "date": "2020-01-20",
        "title": "First day on my trip!",
        "paragraphs": ["Today was my first day travelling
          alone! It was nice.", "I discovered many
          impressive things!"]
    },
    "blog2": {
        "date": "2020-01-22",
        "title": "Discovering my AirBnB!",
        "paragraphs": ["I finally got to lay down a bit
          today!"]
    }
}
```

4. 1 pt: Add a **login button** in the top right corner. This button allows you to log in. The password must again be ‘CS203’, stored as a hash.

5. 1 pts: If you are logged in, then when you view your blog page, new elements become visible: A **delete button** beside each blog item. Clicking on it will display a warning with “are you sure?” and then delete the element on the page (javascript) and the json file (javascript or php). *Hint. Add the delete button inside a form, and use both action= and onsubmit=.*

6. 1 pts: If you are logged in, then when you view your blog page, new elements become visible: A “**Add new post**” button at the top of the page. It opens a new page with a form to add a new post in your json file. Don’t forget to sanitize the user’s input (ex, htmlentities)! Hint: discover <textarea> element!
   https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/textarea

7. 1 pt: At **submission** time: the pull request is done on GitHub and the website works on Osiris.

Optional items (6 pts)

When options a), b), c) are shown, it means that they are mutually exclusive. Ex: it is not feasible to use both choices 3a) and 3b).

Suggestions and hints below are ideas of ways to implement the assignment item, but you don't have to follow them if you have better implementation ideas.

1. Option a) 1 pt. Use **bootstrap** to make your website look nicer.
    a) In the head section, add a link to the bootstrap CSS stylesheet:

```
<link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
```

    b) Use examples online to discover bootstrap classes that could be interesting for you. Do NOT copy-paste these examples! Just get inspired! Make your own style! Click "View source" or "Inspect" to discover what was used.
        a. Bootstrap's own blog example: https://getbootstrap.com/docs/4.0/examples/blog/
        b. Emmanuelle's example: https://osiris.ubishops.ca/home/erenauld/blog_bootstrap.html

1. Option b) 1 pt. Use **tailwind** to make your website look nicer.
    a) In the head section, add a script to add tailwind:

```
<script src="https://cdn.tailwindcss.com"></script>
```

    b) Use examples online to discover tailwind classes that could be interesting for you. Do NOT copy-paste these examples! Just get inspired! Make your own style! Click "View source" or "Inspect" to discover what was used.
        a. Tailwind's documentation : https ://tailwindcss.com/plus/ui-blocks/marketing/sections/blog-sections
        b. Emmanuelle's example : https ://osiris.ubishops.ca/home/erenauld/blog_tailwind.html

2. 1 pt. Use **flexbox** to style your navigation menu. Make sure it adapts correctly to different screen sizes. See flexbox examples here:
    a) https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Flexible_box_layout/Use_cases
    b) https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_website
    c) https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_image_gallery
    d) https://flexboxfroggy.com/

3. Option a) 1 pt. For each post, instead of displaying the whole text, show only a few words or a few lines and add an "**Read more**" button. Suggestion: When clicking on "add more", it could call a new php page, ex, show_post.php, which receives either the full content, or the id of that post to go read it in the json file.

3. Option b) 1 pt. For each post, make the content **collapsible**: we only see a part of the content, and when clicking on it, it expands to show the full content. See example here: https://www.w3schools.com/howto/tryit.asp?filename=tryhow_js_collapsible

4. Add a button to switch between **light and dark themes**. Hint: it could call a JavaScript function that will change which CSS stylesheet you use. Give an ID to your link (`<link id="theme-link" rel="stylesheet"` …) and use JavaScript to access that element and change its href attribute.

5. When adding a new post, add an **auto-save feature**, which will save your content in local storage. You can either
    a. save draft when user click on a “save draft” button
    b. save draft every few seconds

```
<script>
setInterval(() => {
    // … save draft
}, 5000); // every 5 seconds
</script>
```

    c. save draft 2 seconds after user stops typing new characters

```
let saveTimer;
function autoSave() {
  clearTimeout(saveTimer);
  saveTimer = setTimeout(() => {
      // …save draft
  }, 2000); // save 2 seconds after typing stops
}
Input_element.addEventListener("input", autoSave);
```

6. 1 pt. When logged in, add an **edit button** beside each post, that will allow you to edit the text. For instance:
    a. It could send you to a new form to send a post, where element are pre-formatted with the post title and text.
    b. Or when clicking on the “edit” button, you could add contenteditable="true" attribute to the element with javascript. This allows you to change content directly in the browser. Then you could add a “save” button that would tell JavaScript to read the content and save it. An incomplete code would look like:

```
<script>
function save() {
  post_id = ……;
  const content = document.getElementById("edited_div").innerHTML;
  // make sure the content is safe html! Don’t be hacked!

  fetch("save.php", {
    method: "POST",
    body: "content=" + encodeURIComponent(content)
  }).then(r => r.text()).then(…..);
}
</script>
```

7. Add a **comment section** under each post or at the bottom of the page. Comments can be saved in the same json file or in another one. Comments can be sent anonymously or not, with the date or not.

The following options all impact the posts that should be shown. If you choose more than one options here, make sure your options don't break each other!

8. 1 pt. In the aside section, add **filtering options**. It can be buttons or checkboxes. Clicking on them will call a JavaScript function that will show or hide blog posts that don't fit the filtering option. Ex: Posts posted later than some date. Posts on a given topic. Suggestion: Add classes to your posts with significant keywords.

9. 1 pt. In the aside section, add a **sorting option**. Users can choose to sort posts by date or by title.
    a. Suggestion 1: this could reload the same PHP page with a GET value set to say in which order to place posts (between loading the json file and displaying everything in a for loop). To sort dates, if they are in the form Y-m-d, then `sort($arr)` should work.
    b. Suggestion 2: you can reorder elements dynamically in JavaScript. Get all elements and choose the new order. You can use element methods .removeChild () and .insertBefore() to change the order one post at the time, or change the children order at once: `container.replaceChildren(...sortedPosts);`

10. 1 pt. In the aside section, add a **search bar** to search for keywords. Suggestion: use JavaScript to loop on posts and look for the keywork in both the title and the text. Then choose whether to display the element or not.

11. 1 pt. Add an option for the **number of posts** that should be displayed, with previous and next buttons to show other posts. Suggestion: create Javascript variables `current_page` and `items_per_page`. Choose dynamically which posts to show. Then either loop on posts to choose which ones to show, or use `const postsToShow = allPosts.slice(start, end);` and `container.replaceChildren(...postsToShow);`.

The following options concern animations. If you would prefer to add animations in another page than your blog, please explain to me where I should look for them (in your submission email or in your Moodle submission).

12. Explore the **@keyframe** option. DO NOT copy-paste the code. Change the variable name. Add comments to show me you understand the code. Change the images being used or the style.
    a. Add a logo or a picture at the top of the page or in the hero section, which will move or wiggle and hovered AND
    b. When arriving on the page, make a short animation changing the color of the hero section or of the main body

    https://www.w3schools.com/howto/howto_css_shake_image.asp
    AND
    https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1

13. Use one of the **canvas** animations suggested in the website below. DO NOT copy-paste the code. Change the variable name. Add comments to show me you understand the code. Change the images being used or the style.
    https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations

14.

Grading grid:

Item	Value (/ 13)		
	Mandatory : 7 pts		
1	1	new page	New page blog contains enough content? Looks nice? Contains a menu, footer, hero sections?
2	1	aside section	The aside section is on the right? It lists all blog posts with hyperlinks?
3	1	json file	The blog posts are loaded from a json file? The page in MCV-like?
4	1	login button	The log in button allows you to log in correctly?
5	1	delete button	When logged in, the delete button appears? It works?
6	1	add new post	When logged in, the add new button appears? It works?
7	1	submission	The pull request is done and the web site is live on Osiris?
	Optional : 6 pts		
1a) 1b)	1	bootstrap OR tailwind	The blog looks nice? (Used bootstrap OR Used tailwind)
2	1	flexbox	Flexbox for the nav menu? Looks nice when changing the screen size?
3a) 3b)	1	Read more OR collapsible	Make content shorter: Add a read more button OR Make content collapsible
4	1	light and dark themes	The button to change theme between light and dark works.
5	1	auto-save feature	The text is saved upon refresh.
6	1	edit button	The edit button works and allows editing the text correctly.
7	1	comment section	The comment section allows adding comments that are correctly stored in memory.
8	1	filtering options	Filtering options in the aside section work?
9	1	sorting option	Sorting option works correctly (by date or title)?
10	1	search bar	The search bar searches correctly through posts?
11	1	number of posts	The correct number of posts is shown at the time?
12	1	@keyframe	An image wiggles AND a background changes color?
13	1	canvas	Created a canvas animation? It was unique and personalized?