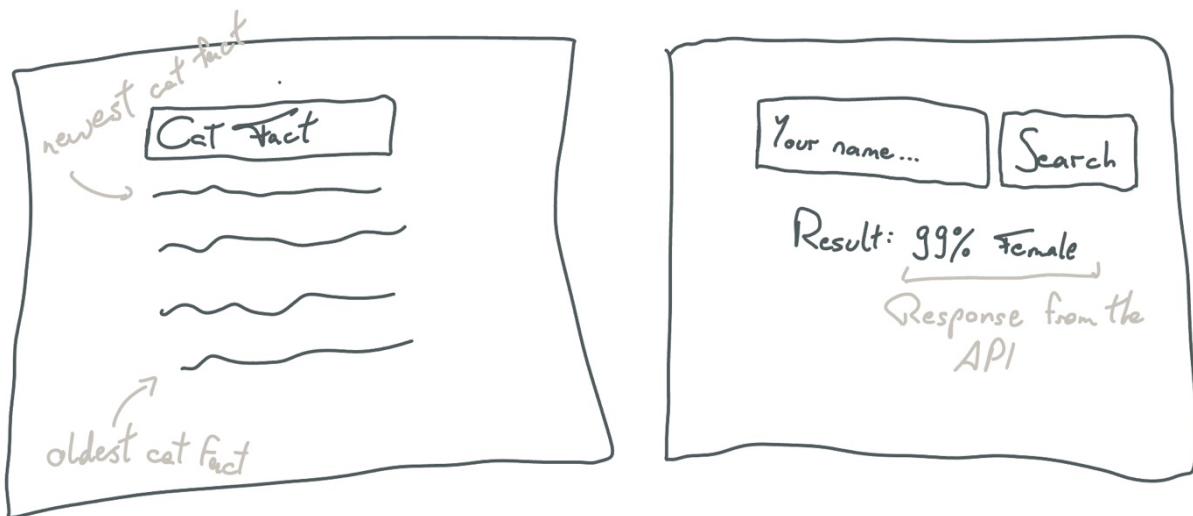


PROBLEM SET 3

This graded problem set (8 points total) is due Sunday, February 12, 2023.

3.1 - JAVASCRIPT (3PT.)

For this task, you will need to create two simple webpages. You are free to use third-party CSS libraries or frameworks such as Bootstrap, Tailwind, paper CSS, etc. to make the styling process easier.



- a) The first webpage gives the user interesting facts about cats. To do this, use JavaScript to access the REST service API of this website:

<https://catfact.ninja/fact>

Any time the user presses on the “Cat Fact” button, a new cat fact should be fetched and prepended to the existing list of cat facts. As in, once a new fact has been received, it should appear above all the other cat facts.

Make sure that each fact is put inside a separate html tag (i.e., don’t just change the text of a single div element).

- b) Given a GET request parameter called “name”, this REST service API predicts the gender of the given name:

<https://api.genderize.io/?name=...>

A text input field and submit button should allow the user to find out, if a name given by them is predicted to be male or female of gender. Note that the webpage should not reload.

3.2 - SPRING BOOT (3PT.)

Clone the Spring Boot project that has been pushed to <https://github.com/DauphineWeb/> during class. Get it up and running and test the different routes.

- Currently, bad requests (code 400) and internal server errors (code 500) are not displayed nicely. Create a static website for both that inform the user on what end something went wrong.
- The room filter in “/rooms” must now include an additional optional parameter “startsWith”. If set, it should only include rooms in the response that start with the requested parameter value.
- Room names must be unique. Add additional constraints such that a user cannot add a room with the same name of an already existing room.
- Add a new route “/deleteRoom” that lets the user delete a room for a given ID. Use the appropriate `@DeleteMapping` to do this. Add a corresponding view for the user where he can select what rooms to delete.
- Change the Thymeleaf html as follows:
 - if rooms are present, render the html table as before.
 - if no rooms are present, do not render the table and instead display the message, that there are no rooms.You may want to check the [if-unless](#) conditional from Thymeleaf.

3.3 - PROJECT SETUP (2PT.)

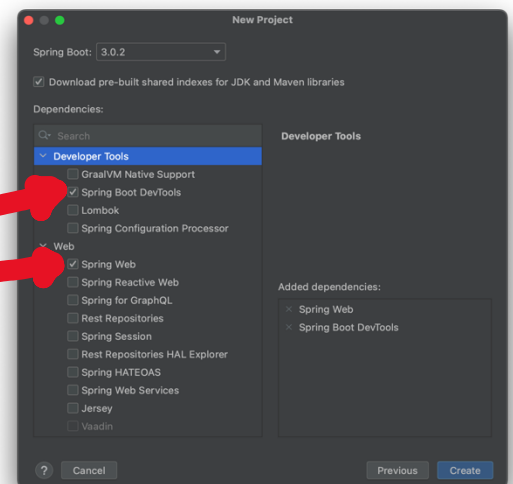
In this problem set, all group members will receive the same number of points. I will not check for individual contribution, rather just for the end result.

- The owner of your GitHub project should create, commit and push a base project of your webapp.

If you have chosen to use [Spring Boot](#) for your project, make sure to include Spring Boot DevTools and Spring Web from the list of available dependencies. You may also add others if you already know that you will need them.

- Add an index page presenting your website. It should present your service and give an idea to the user, what they are able to expect.

- Add a [favicon](#) to your project. You can use your logo from the last problem set if you believe it fits well.



From this point, you are not required to limit your project’s progress to the requirements set in the problem sets. You are free to develop and tinker on your project as much as you like, as long as the baseline requirement set with every problem set is met.

As always, please send your solutions for 3.1 and 3.2 to felix.fritz@dauphine.psl.eu.