

## Software Development

### Project 1: Django

1. Choose a name for your team.
2. Choose a project manager. The manager will help coordinate meetings and discuss how to split the job with team members.
3. Choose a topic for your website. Try to choose something interesting for all the team members.
4. Your Django project should have at least 2 apps with at least 3 views in each application.
5. Your project should have the following features:
  - a. Cookies:
    - i. Set up SameSite lax cookies.
    - ii. Session Expiration
    - iii. You are going to save the data in the cookies.
  - b. One form with Django Forms
  - c. The same form without using Django Forms
    - i. You have to replicate manually the same behavior as the Django form has (for example, the error sent when something is wrong on the form)
  - d. For both forms:
    - i. I expect the same result with both forms. Please insert the form data into cookies.
    - ii. Use JSON.
    - iii. For the data you will save, I expect a list of dictionaries.
  - e. Create a view to see the list of data coming from the forms (this is called a list view). You must add an option to delete one dictionary from the data list (delete button). Once we click to delete one dictionary, we should have the page refreshed to see the new list updated. Also, on this page, you can add 2 create buttons that will redirect to each form.
  - f. Another option will be to update one dictionary from the data list (edit button). This will redirect to the form with the data prefilled for the dictionary chosen. Once the form is submitted, it will update specifically this dictionary (not creating another dictionary in the list, nor modifying another dictionary; I want the one I choose on the edit button). Obviously, I should not have to mention it, but once the edit has been submitted, we should be redirected to the list view.
  - g. A search form to be able to search something into your data and retrieve the result according to the search input.
  - h. Images on some views. Add a button next to the picture to be able to download it.
  - i. Links to be able to navigate from page to page.
  - j. A home web page.
  - k. Menu in the base template. It does not have to be sophisticated; links to other web pages are okay. No javascript. You will need to highlight the link to the current page we are on. If the current page does not have a link on the menu (in

a big application, you will not put all the links in the menu), you will need to highlight the parent link of the current page.

- I. CSS (Optional, if you know it, to give a nice look to your website).
6. Create a detailed log to track the team members' work. Log your tasks indicating the date you developed those tasks. Also, include the decisions made by the group after each time you have a meeting.
7. Create a document explaining the structure of your website (optional: add a diagram of the structure). Submit a digital and a hard copy of this document.
8. Projects should be available on GitHub. Please create branches. So I can see the logs and check the team members' activity. The project manager should share the project repository with the teacher.

You cannot use JavaScript, bootstrap (or any alternative), or front-end libraries for this project. Just pure HTML, CSS (if you know it, it is not required), and that's it on the front end. I do not want to see you importing JavaScript or CSS libraries. It's the same thing on the backend, only Django, no other Django package.