

# PROBLEM SET 4

This graded problem set (9 points total) is due Sunday, March 19, 2023.

## 4.1 - DJANGO (5PT.)

Your task is to create a simple, community-driven website for submitting snippets of texts, i.e., for people to share [jokes](#) or their favorite [horror stories](#) (or other text-based content).

For instructions on how to complete some of the steps listed below, feel free to have a look at the README below, as well as the outstanding documentation on the Django page. <https://github.com/DauphineWeb/ProgrammationWeb2023/tree/main/03-08%20Django>

1. Create a project.
2. Create a single app within your project that handles all the traffic.
3. Introduce two routes, one index route `' '` for the homepage, and one `'add/'` route that lets users add new jokes (or horror stories, or others).
4. Both sites should link to each other. Users should be able to get from `'/'` to `'add/'` and vice versa.
5. For now, user entries can be saved in a global array on the server.
6. The submit page should contain a text-area for the content, as well as a text-field for the user to enter their name. Note that:
  - a. The main content must be between 10 and 120 characters long.
  - b. The name must not exceed 20 characters.
  - c. Once the request reaches the server, names should be capitalized (i.e., `'maria'` -> `'Maria'`)
  - d. *Optional:* [block profanity](#) and vulgar expressions.
7. After their submission, the user should be redirected to the homepage and see their submission **highlighted** (i.e., through some colored background or border), next to all other submissions.
8. Next to the main content and the user's name, each entry saved on the server should also have an id attached to it.
9. Add a URL `'entry/<id>'`. It should return a JSON object of the user entry based on the id. I.e., `'entry/1'` should show the first user submission and so on.
10. Add a way on your homepage to share a link to an entry. It doesn't matter if that link only returns the JSON object.

## 4.2 - GROUP PROJECT (4PT.)

In your groups, discuss what some of the different parts that make up your application could be.

Set up your application. Add at least **four** routes. This could be the homepage, the login / signup page, a page that the user sees and a settings page for the user, to name a few examples.

Link these pages together (no login required yet). Populate these pages with mock content, as in, add enough example data to get a feel of what the web application could feel like.

If you need help or more concrete guidelines, do not hesitate to contact me.

As always, please send your solutions for 4.1, either as compressed .zip file or as a link to your GitHub repository, to [felix.fritz@dauphine.psl.eu](mailto:felix.fritz@dauphine.psl.eu).

For exercise 4.2, I will check by your GitHub repository after the submission date.