Work test for Back-end developer

The task

Write a simple blogging app.

- Some views to include
    - The blog post list
    - The detail view of the blog post
- The app should be able to accept comments for blog posts
    - Comments should be anonymous, no need for users
    - This should be done asynchronously
    - The comment should appear immediately after posting (using js)
- Some admin capabilities [must be able to log in (please use the Symfony security bundle)]
    - Can add/edit title of blog posts
    - Can add/edit content of blog posts
    - Can delete/hide comments
- Search function, so that the user can search in blog posts (both title and content)

There is no need to include paginators (but is allowed) because the review will not include many posts.

Technology

- Do not use pre-built blogging libraries, we want to know how you solve tasks on your own
- Symfony or if you feel more comfortable with Laravel, you can use that but considering you will be working with Symfony then we prefer answers which use Symfony
- MySQL or any relational database of your choosing but then it has to be outlined in your answer as to why you chose that
- For the front-end use a CSS framework, there is no need to write your own CSS (but is not prohibited)
- You do not have to use a front-end library like react or angular. Simple http posts/get will suffice

Delivery

- Sources for building and executing
    - Preferably in a private GIT repository, please give access to it
    - Otherwise the project can be zipped and sent
- Do not zip/push large files/folders e.g. vendor, node_modules etc.
    - Those should be generated/installed when installing the project
    - Only when really necessary and cannot be generated when installing
- The project should be able to run locally
- Some sort of explanation for your solution
    - e.g. Why you chose some library and what problem does it solve
        - You do not have to explain all libraries just the ones you decided to add
    - How to deploy/install your application
    - Can be informal

Expectations

- The goal is to judge your ability to design a system covering the outlined requirements

- You should not spend too much time on the appearance of the front-end, we expect a working solution
- The system does not have to be perfect but should give an idea of how you approach problems
  - No need for perfect error handling