Developer Test

You are developing a simple to-do application to demonstrate your web programming skills. The application should display a personalised to-do list with ability to add and complete items. Main requirement for technology is use of Symfony 2 as a core of your implementation. Feel free to show your other web skills including CSS and JavaScript and whatever other tools and technologies you are best at. It is completely optional as this is a test and not a final product.

Functional Requirements

The application is to be used by a number of users, hence some sort of user identification need to take place, ideally Google OAuth authentication process. We would recommend using one of the many OAuth Symfony bundles available to demonstrate your understanding of Symfony, Composer and third-party bundles, combined with configuration of the security components of Symfony.

Users should be able to see a list of their outstanding to-do items. As an optional useful bonus, allow users to see all or recently completed items (i.e. last 5 completed items).

Allow users to add new to-do items as well as the ability to mark them off as complete.

Each to-do item should have a status flag (complete/active) and a short description (255 chars will do). If you are feeling adventurous, add an optional due date field.

Administrator of your application should have an ability to see all to-do items submitted. A simple secure admin section listing all to-dos will need to be implemented. No need to implement a full backend, just a separate section secured with any form of authentication. This should be styled using Bootstrap to show your abilities to theme forms and use external libraries. Anything goes here, a shared login/password via HTTP basic auth, login form or OAuth, anything is acceptable as long as your chosen method restricts public access.

In case of invalid data or application error user should be presented with appropriate error message. No need to cover absolutely all possibilities but core edge-cases should be handled.

Technical Requirements

Since its a test SQLite seem like the most appropriate data connector but feel free to use MySQL, XML, JSON or whatever else. ORM of choice is Doctrine for us but you can use raw DBAL or standard php functions. We do not expect you to use data entities but feel free to do so if it is easier for you.

You can use any third party bundles and libraries.

Adhering to Symfony2 coding specs and having a project that deploys without issues is the bare minimum expectations.

Deliverables

Please provide a public GitHub URL with suitable deployment instructions, plus appropriate Nginx configuration.