Project of the second evaluation of Web Development in a Server Environment: Messaging application

Introduction

The project consists of making a web application that allows users to exchange messages. It is similar to the usual internal messaging systems in companies or a web mail system.

Optionally, an application can be developed with an approach more similar to Telegram or WhatsApp.

The application will be developed using Symfony, Doctrine, MariaDB/MySQL and TWIG (unless it is done as a single page application).

Basic functionality

- Access: Users access the application with a username and password that are stored in a database, using Symfony's user system.
- Send: Users of the application can send messages to other users (the recipient is chosen by name)
- Inbox: Users will be able to see a list or table with the messages that have been sent to them, differentiating between read and unread messages. In the inbox the complete messages are not shown, only the subject or the beginning of the message.
- Messages: You will also be able to consult each message individually. Here, the complete content of the message is shown and there is also the opportunity to answer.

Extensions

Each student/group will choose from among the following which extensions to incorporate into their application. Each extension can obtain a maximum grade of one point.

A1. Selfregister

- Realistic process for users to register on the website.
- To get the full score, you have to include an email with an activation link.

A2. Password recovery

- Realistic process for users to recover their access password or generate a new one.
- If the solution is to send a new key (or the current one), without having to perform any further operations, only half the score will be obtained.

A3. Messages to multiple recipients

- The application allows entering more than one recipient in the message sending form.

A4. Encrypted user key

- User passwords are stored encrypted, using Symfony's user system.

A5. User avatar

- Users will be able to upload an image to use as an avatar.
- It has to have some consequence in the application. For example, that the avatar appears in the messages that a user sends, or that it can be seen when accessing their profile.
- Users have to be able to modify their avatar.

A6. User profile

- Users will be able to enter and modify a series of data about themselves, such as age, city of residence or hobbies.
- Other users need to be able to access this information.

A7. Friendship

- Two users can establish a friendship relationship with each other.
- One of them starts the process by sending a request, which the other has to accept or reject.
- It has to have some consequence in the application. For example, that you can only write to friends or that a user's profile is only visible to their friends.

A8. Groups

- Option 1: Develop a group system similar to Telegram or WhatsApp.

- Option 2: Allow the creation of groups that are collections of usernames. When you send a message to one of those groups, it is sent to each of the users.

A9. Administration area

- Develop an administration area only visible to some users (according to their role).
- In this area you have to implement tasks typical of an administrator. For example, block or unblock users.

A10. Attached files

- Allow attaching files to a message.

A11. Images

- The app allows inserting images as part of a message.
- For this extension to be considered correct, the images must be seen directly when viewing the message, not as attachments or following a link.

A12. Testing

- Incorporate acceptance testing using Symfony's testing component.
- The note is proportional to the number of sections tested.

A13. Outbox

- The user will be able to see a table with their sent messages, similar to the inbox. Messages will be sorted by date sent, most recent first.
- In the table you have to see the recipient of the message. If there are several, it can be indicated by putting "Various".
- From the table you can access the detail of each message, as in the inbox. In the detail of the message, the complete list of recipients must appear, indicating which of them have read it.
- If the application does not allow sending to more than one user, only half of the score can be obtained in this extension.

A14. Presentation

- Public presentation of the application in the classroom.
- Includes description of the development process and demo.
- The mark is proportional to the number of sections made.

A15. Use of AJAX.

The application uses AJAX for some of the controller(s). The score will be proportional to the functionality performed.

A16. Single page application.

The application is implemented as a single page application integrating with the Symfony user system.

Things to turn in

- Script to create the database. It must contain sample data.
- Full application code.
- A PDF document of a maximum of 30 pages with:
    - Specification of requirements. It must include a table indicating which extensions have been made.
    - Scheme E/R.
    - Logical model of the database.
    - Database diagram obtained with SQL Developer.
    - Screen map.
    - Summary table of the controllers that are part of the application.
    - User manual, including a description of the data loaded in the database (users, passwords, groups...)
    - For each extension, the corresponding section will be included in the documentation.
    - The work is delivered through the virtual classroom.
    - It can be done in pairs. Each member will have to upload the work separately. In addition to the delivery in the virtual classroom, the work must be presented to the teacher in the classroom. The date and time of this presentation will be informed through the virtual classroom.
    - The works uploaded to the virtual classroom that have not been presented in the classroom will not be corrected.
    - The work must be uploaded to the virtual classroom before presentation to the teacher.
    - For practices carried out in pairs, the grade will be obtained by multiplying the grade of the work by 0.8.
    - IMPORTANT: Remember to delete the data from your email account.

Grading

Basic functionality (4 points)

- PDF documentation and sample data (excluding parts only relevant to extensions): 30%
- Documentation and code quality: 10%
- Bug control: 20%
- Functionality: 40%

Extensions (1 point each a)

- Documentation in PDF and sample data: 20%
- Documentation and code quality: 10%
- Error control: 20%
- Functionality: 50%

IMPORTANT:

- **If the A16 extension is not carried out, it will not be possible to obtain a grade higher than 9.**
- **If both A3 and A8 extensions are not carried out, it will not be possible to obtain a grade higher than 7.**

- Extensions

EXTENSION	IMPLEMENTED (Y/N)
A1	
A2	
A3	
A4	
A5	
A6	
A7	
A8	
A9	
A10	
A11	
A12	
A13	
A14	
A15	
A16	