Fullstack Assignment - Messaging system

OVERVIEW

write a simple, responsive, web app in client & server that handles messages between users.

Client side can be one of the following: angular 8+, react (Hooks, redux, saga), VUE + VUEX..

Server side can be one of the following: node.js (express or any other) / Python (Django / Flask). (If you choose to use Django, please provide back office admin user/password).

We recommend to use a 3rd party libraries backend & Frontend (material, bootstrap repo, grid libraries, and, etc)

Backend

The server should have a list of API, by user Id which are written below

1. Write message
2. Get all messages
3. Delete message

Message Object

4. Sender (owner)
5. Receiver
6. Message
7. Subject
8. Creation date

Database is not Mandatory

Frontend

The UI should support the server functionality, and contains two pages:

1. **Compose Email Page -** ability to write message, which should contain the following
    a. Subject and message box fields
    b. Sender id input field
    c. Receiver id input field
    d. Send message button

    The behavior of the page should be similar to sending email functionality, validation, and clear the input with a success message after the message was sent and stored on the server.

2. **Mange Emails Page -** should contain the following
    a. User id input field (upon changing this field the presented screen should be update)
    b. Get all messages - display them in tabs sent and received
    c. Delete message button (for each message) + prompt message

Important notes for this task

- Try to be clean and write a “production level” code.
- Database is not mandatory
- Handle errors

GOALS

1. Show us your coding skills - We want to see your strengths, don’t be afraid to go wild, add features and customize the design.
2. Readability - Your code should be readable and self-explanatory with minimum comments. Remove any unused code, console logs, and unneeded files. Use logical project structure and code separation.

Bonus Task

Use an authentication method to set up the request for the logged in user. so if you will ask for the GET messages API you will only get the messages for the current logged in user.

HOW TO SUBMIT

1. Push your code to your git repository and make it public (we recommend GitHub in order to use GitHub Pages with ease), **name the repo as firstName-lastName-dateOfStart (John-Doe-01-12-2018)**.
2. Deploy the compiled app to your server or GitHub pages/Heroku/etc - **MUST**

Send us the git repository link and a link to the deployed app (2 links)

Good luck, Herolo Team