



C1 - Stack MERN

C-COD-121

MERN pool rush

M.E.R.N. => mern M(ongoDB) E(xpress) R(eact) N(ode.js)



MERN pool rush

repository name: rush-mern
repository rights: ramassage-tek
language: node, express, mongodb, react



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).

FOREWORD

Welcome to the MERN rush, ending the MERN pool!

This rush will use every aspect of what you have learned during the MERN pool, and will let you combine Express/Node with REACT and MongoDB.

In order to quick start the environnement of your project, you will use the instruction on this website : <https://alligator.io/react/mern-stack-intro/>

It lets you to have a front and a back end operational with MERN Stack quickly.

Your project must implement at least the basic functionalities described below!

As rushes give you more freedom than exercises, once you have the basics working, don't hesitate to add functionalities. These functionalities will be awarded bonus points.



You are strongly advised to use Git for version control. This is a good habit to have. If you don't, we won't be very pleased !!



WHAT YOU HAVE TO DO

You will develop a microblogging web application named MicroBloggoss that allows users to post short messages (140 characters maximum).

In order to post a message, a user needs to be registered.

In order to view his flow of messages, a user needs to be logged in.

+ RESTRICTIONS



On the frontend side, as you will be using REACT, we want your application to be a Single Page Application.

In other words, the parts of the view modified by an action must be reloaded. You will probably have a central “container” that will change as you browse the site.

As a consequence, every feature is accessible on one page.



Bootstrap/Foundation is allowed (to look nice without working too much on it) but use it wisely. If we like what we see, you know what's coming your way.

PHASE 1 - MEMBERS (5 PTS)

You will create a simple user system.

A user is composed of a `username`, an `email` and a `password`.

You must implement the following features:

- Registration
- Connection
- Disconnection
- Member list
- View the current user profile page
- Editing the user information



PHASE 2 - FLOW TIMELINE (5 PTS)

Upon connection:

- The user will be able to post short messages (140 characters maximum).
- Each user will see displayed the message flow (messages they have posted) in real time and without reloading the page.
- The user will be able to edit and delete his messages.



All this on the same page.

PHASE 3 - FOLLOW (5 PTS)

You must ensure that it's possible to register to the message flow of other members.
Therefore, you will modify the page in order to display all the messages flow a user has subscribed to.
Near each message, the author must be displayed.
In addition, the user should be able to manage the user that follows him and those he is following.
This includes deregistering a user message flow or blocking access to new messages to a user.



All this on the same page.



PHASE 4 - TAG (5 PTS)



Restrictions: No style attribute in the HTML, No JavaScript code in the HTML.

You must implement a hashtag system.



A hashtag is a word of the format #myhashtag written in a message. In this example, the hashtag is myhashtag.

By clicking on the hashtag, you should see a list of all messages that have a reference to it. In addition, the hashtag can be used to search all related messages.



All this on the same page.

BONUS - DO YOUR BEST!

As usual the sky is the limit, but here is an incomplete list of possible bonuses:

- Implement an avatar system
- Ability to register to your app through Facebook/Twitter API
- Content sharing
- A “remember me” option
- Admin space

We love surprises! (Good ones obviously)