



NodeJS

PROJECT: MEMBERS ONLY

Introduction

In this project you'll be building an exclusive clubhouse where your members can write anonymous posts. Inside the clubhouse, members can see who the author of a post is, but outside they can only see the story and wonder who wrote it.

This will be a chance for you to use the authentication skills we learned in the last project, you also be practicing your database skills so buckle up!

Assignment

1. Begin by thinking about how to set up the database models you are going to need to accomplish your goal. You will need users with full-names (first and last), usernames (you can use email for this) passwords and membership-status. Users should be able to create messages that have a title, a timestamp and some text. Your database should keep track of who created each message.
2. Setup your database on Mongo and generate or otherwise create your project skeleton, including the models you designed in the last step.
3. Start with a sign-up form so you can get some users into your DB! Don't forget to sanitize and validate the form fields and secure the passwords with `bcrypt`. You should add a `confirmPassword` field to your sign-up form and then validate it using a custom validator. Read how to do that [here](#).
4. When users sign up, they should not be automatically given membership status! What fun is a private club if just anyone can join? Add a page where members can "join the club" by entering a secret passcode. If they enter the passcode correctly then update their membership status.
5. Create a login-form using passport.js like we did in the last assignment.
6. When a user is logged in give them a link to "Create a new message" (but only show it if they're logged in!). Create the new-message form.
7. Display all member messages on the home page, but only show the author and date of the messages to other club-members.
8. Add an optional field to the user model called Admin and then add the ability to delete messages, but only, allow users who have `admin == true` to see the delete button and delete messages. You'll need to add a



allow users who have `admin == true` to see the delete-button and delete messages. You'll need to add a way to actually mark a user as an 'admin' so either add another secret pass-code page, or just put an "is admin" checkbox on the sign-up form.

9. By this point, anyone who comes to the site should be able to see a list of all messages, with the author's name hidden. Users should be able to sign-up and create messages, but ONLY users that are members should be able to see the author of each message. Finally, you should have an Admin user that is able to see everything and also has the ability to delete messages. Obviously this is a simple and silly little app, but

the things you are practicing (creating and authenticating users and giving users different abilities and permissions) are things that will be *very* useful to you!

10. When you're satisfied with your work, deploy your project to heroku and share it below!

Solutions:

Add Solution

NodeJS: (Members Only)

5 ♥	AtifCppProgXXI	View Code	Live Preview	
2 ♥	Daxas	View Code	Live Preview	
2 ♥	Pavel Nikolaev	View Code	Live Preview	
1 ♥	shivam@16122000	View Code	Live Preview	
1 ♥	lprzybyslawski	View Code	Live Preview	
1 ♥	wickettc	View Code	Live Preview	
1 ♥	DoviMaj	View Code	Live Preview	
1 ♥	daggersilver	View Code	Live Preview	
1 ♥	tatianatothe	View Code	Live Preview	
0 ♥	Dhillman1989	View Code	Live Preview	

Showing 10 most liked submissions - [View full list of solutions](#)

[View Course](#)

[Login to track progress](#)

[Next Lesson](#)

 [Improve this lesson on GitHub](#)

Have a question?

Chat with our friendly Odin community in our Discord chatrooms!

[Open Discord](#)

Are you interested in accelerating your web development learning experience?

[Get started](#)



5-6 months



Job Guarantee



1-on-1 Mentorship





About
FAQ
Blog

Success Stories
Contribute
Terms of Use

