

# **Stack application**

## **Project documentation**

# Installation

Clone the repository from the GitHub <https://github.com/ltoikon/awa-fullstack-project> and open the project root directory in the terminal.

Install Node.js dependencies with the following command:

```
npm install
```

To ensure you have compatible Node.js version you may use these commands to install:

```
nvm install 16
```

and to choose the version:

```
nvm use 16
```

Start backend and frontend with a command:

```
npm start
```

If you want to start those separately you can use commands:

```
npm client
```

and

```
npm server
```

The application will be accessible on browser at <http://localhost:3000>

# User manual

## Register and Login

Press Register link in top bar to register new user. Only requirement for registration is that username have to be unique. Successful registration will redirect user to Login view. Login link is available also in the top bar

To login, user will need to provide username and password. After successful login, application will redirect to the frontpage.

## New post

On the top bar New post link opens page with the New post -form. After filling and submitting a new post, the post will appear on the frontpage.

## Frontpage

Application's frontpage works as post listing. It will show all posts which are done so far. Clicking anywhere on a post will open the post on Thread view.

## Thread

In this view user sees the post, all comments to that post and at bottom of comment list is textfield to submit new comment to the post. After submitting, the new comment will appear immediately to the comment list.

## Notice

The application is not yet a complete product. In the sense that it doesn't have all the features that an application should have to be regarded as acceptably complete. Missing features contain, but are not limited to, error handling and a Log out -function.

# Technology choices

## Back-end

### Database

- Mongoose – Accessing the database
- MongoDB – NoSQL database

### Server

- Node.js – Server environment for Javascript
- Express – Creating API connections

### Authorization and authentication

- bcrypt – Hashing passwords
- JSON Web Token (JWT) – Authenticating user

## Front-end

- React – Front-end framework
- Materialize CSS – Styling

## Points proposal based on features

Feature	Requested points
Basic features (as stated in requirements document)	25
Utilization of React framework	5
Creation timestamp is stored and shown with posts/comments	2
<b>Total points</b>	32