

## Project work

### Description and technologies:

For this project I created a forum using Node.js (v.14.21.2) and Express, MongoDB and React. Forum allows login and registration, posting, commenting, and voting. Only logged in users can vote and users can also edit their own posts and comments and change vote. Other users' profiles can be viewed, and own profile bio can be edited. The forum has home page where all posts (title, user, last edit, votes) are listed, and each post can be opened and then comments, and post (all previous and content) are listed.

The frontend is built using React framework and the client runs on port 3000 (homepage: <http://localhost:3000/home>). React Routes are used to allow user to navigate in the frontend. There is also a 404 page for non-existent Routes. User authentication is done with JWT that is saved as a httpOnly-cookie. User information is then fetched using this cookie and saved as a React state, so login survives page refreshes. Posts, and other data related to them (e.g., votes and users), also use React states. But again, data is fetched, this time from url-parameters, so for example the page showing the full post and comments will survive a refresh. Materialize is used for styling and responsiveness. The app is usable for both mobile and desktop.

The backend server is built using Node.js and Express framework and runs on port 1234. The backend takes care of showing, adding, and editing of content and the authorization. If some actions are restricted to logged in users, then the calls use authorization. This is done using JWTs that are sent as httpOnly-cookies with requests. MongoDB is used as database and mongoose is used with the backend. The database runs on port 27017/codedb.

### Installation guideline and user manual:

Using this application requires Node version 14.

The application can be installed and run with the following commands:

```
npm install
NODE_ENV=development npm run dev:server
npm run dev:client
```

Additionally, MongoDB needs to be run at [mongodb://localhost:27017/codedb](http://localhost:27017/codedb)

Now home page is usable at <http://localhost:3000/home>

**List of features:**

<b>Feature</b>	<b>Points/Max points</b>
<b>Basic features (as stated in the previous chapter) with well written documentation</b>	<b>25/25</b>
<b>Users can edit their own comments/posts</b>	<b>4/4</b>
<b>Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other</b>	<b>5/5</b>
<b>Vote (up or down) posts and comments (only one vote per user)</b>	<b>3/3</b>
<b>User can click username and see user profile page where name, register date, and user bio is listed</b>	<b>2/2</b>
<b>Last edited timestamp is stored and shown with posts/comments</b>	<b>2/2</b>
Code is written and commented in English. Variables are named in an explanatory way, so it should be readable.	
<b>Total</b>	<b>41/50</b>