**CT30A3204 Advanced Web Applications**

**Responsible teacher: Erno Vanhala**

Documentation for project work

Joona Pellinen

5.3.2023

**Technology choices**

Features

This application has different features as listed below:

* The app can be used without authentication, but non-authenticated users can’t post or comment, they can only view them
* The user can register to the app, and then login to authenticate
* After the user is authenticated, user can post new posts to the website, and comment to existing posts

Frontend

Coding language used in the project is JavaScript. I chose React as frontend framework, because it became familiar to me earlier in the course. React provided easy handling of the data, especially with useState hooks. With the hooks I could keep user data saved during the session, and for example add username to the users post as shown below:

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

For UI library I chose Bootstrap because I used it on my own web project that I made last year. It was very familiar to me and I liked it more than Materialize. From bootstrap I used components: Card, Container, Button, Form, InputGroup, Modal, Col, Row, Nav and Navbar. With Bootstrap I used grids to design the system to be responsive and the application automatically scales with the used device.

Backend and Database

Backend was implemented with Node.js with Express framework because express was familiar from earlier course work. Database used is MongoDB.

Accessibility testing

* Input fields, such as posting a new post to the website, work with “Enter” key on a keyboard. The user can also move through input fields with “Tab”
* Closing the comment field works with pressing the “Escape” key when the comment field is open
* The website works with screen readers. This was tested with Google Chrome Screen Reader extension
* The website doesn’t work with voice commands. This was tested with multiple different Google Chrome voice recognition extensions

**Installation guidelines**

Welcome to the installation guidelines for this project!

NOTE! THIS PROJECT WAS DONE WITH NODE VERSION 18.12.1

First get the source code from my github behind this link and copy it to your own computer in a folder of your choosing:

<https://github.com/joonaaleks/project.git>

Open the folder in which the code is located on a code editor, for example Visual Studio Code and go to the root folder “project” on your code editors terminal. After the terminal command line is in the root folder, install concurrently with command:

**npm install concurrently**

Your dependencies in root folders package.json should look like this

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Then go to “server” folder in your code editors terminal and install following dependencies with npm:

**npm install bcryptjs**

**npm install cookie-parser**

**npm install cors**

**npm install debug**

**npm install dotenv**

**npm install express**

**npm install express-validator**

**npm install jsonwebtoken**

**npm install mongodb**

**npm install mongoose**

**npm install morgan**

**npm install multer**

**npm install nodemon**

Your dependencies in server folders package.json should look something like this with nodemon added

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

After you have installed necessary dependencies in the “server” folder, go back to the root folder and then go to “client” folder in your terminals command line. Then install following dependencies in the “client” folder with npm:

**npm i @testing-library/jest-dom**

**npm i @testing-library/react**

**npm i @testing-library/user-event**

**npm install bootstrap**

**npm install buffer**

**npm install http-proxy-middleware**

**npm install react**

**npm install react-bootstrap**

**npm install react-dom**

**npm install react-router-dom**

**npm install react-scripts**

**npm install web-vitals**

**npm install react-i18next i18next –save**

**npm install i18next-http-backend i18next-browser-languagedetector --save**

Your dependencies in client folders package.json should look like this

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Before starting the application, you should also make connection to a MongoDB database with this connection string:

**mongodb://127.0.0.1:27017/project**

After installing necessary dependencies and connecting to the database, you should be able to run following command in the root folder to start the application:

**npm start**

**User manual**

Welcome to the user manual of my Advanced Web Applications project work. In this manual you will be guided on how to use the application.

Let’s start with the front page of the application:

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

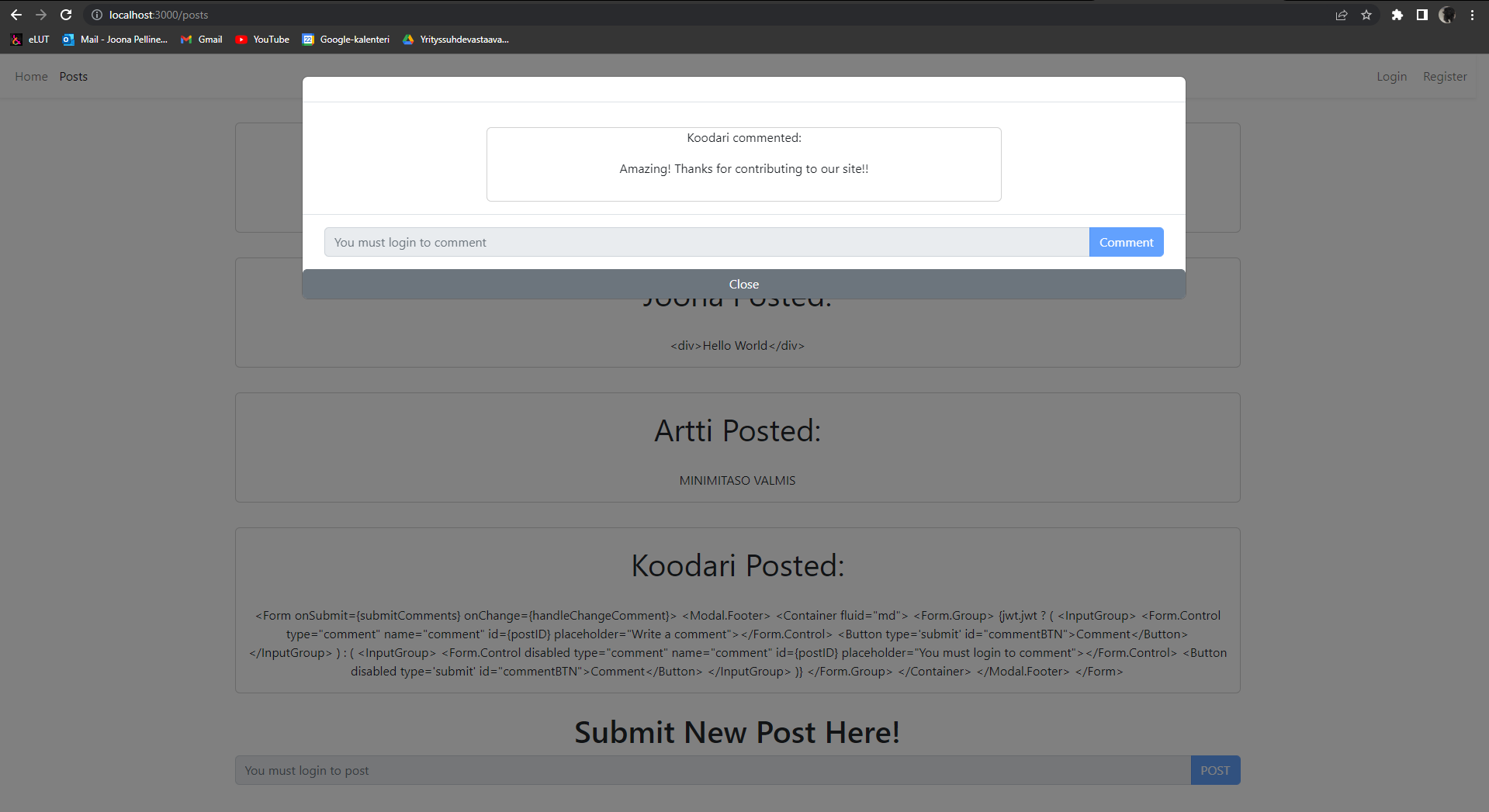
In the front page you can’t do much, but you can access the navigation bar on the top of the page. On the navigation bar you can access posts page behing the “Posts” link, register page behind the “Register” link and login behind the “Login” link. You are probalby not a registered user yet, but you can still view the posts page and see other users posts on the page. Without authentication you are not able to post or comment.

**NOTE: You must wait a moment for the posts to load!**

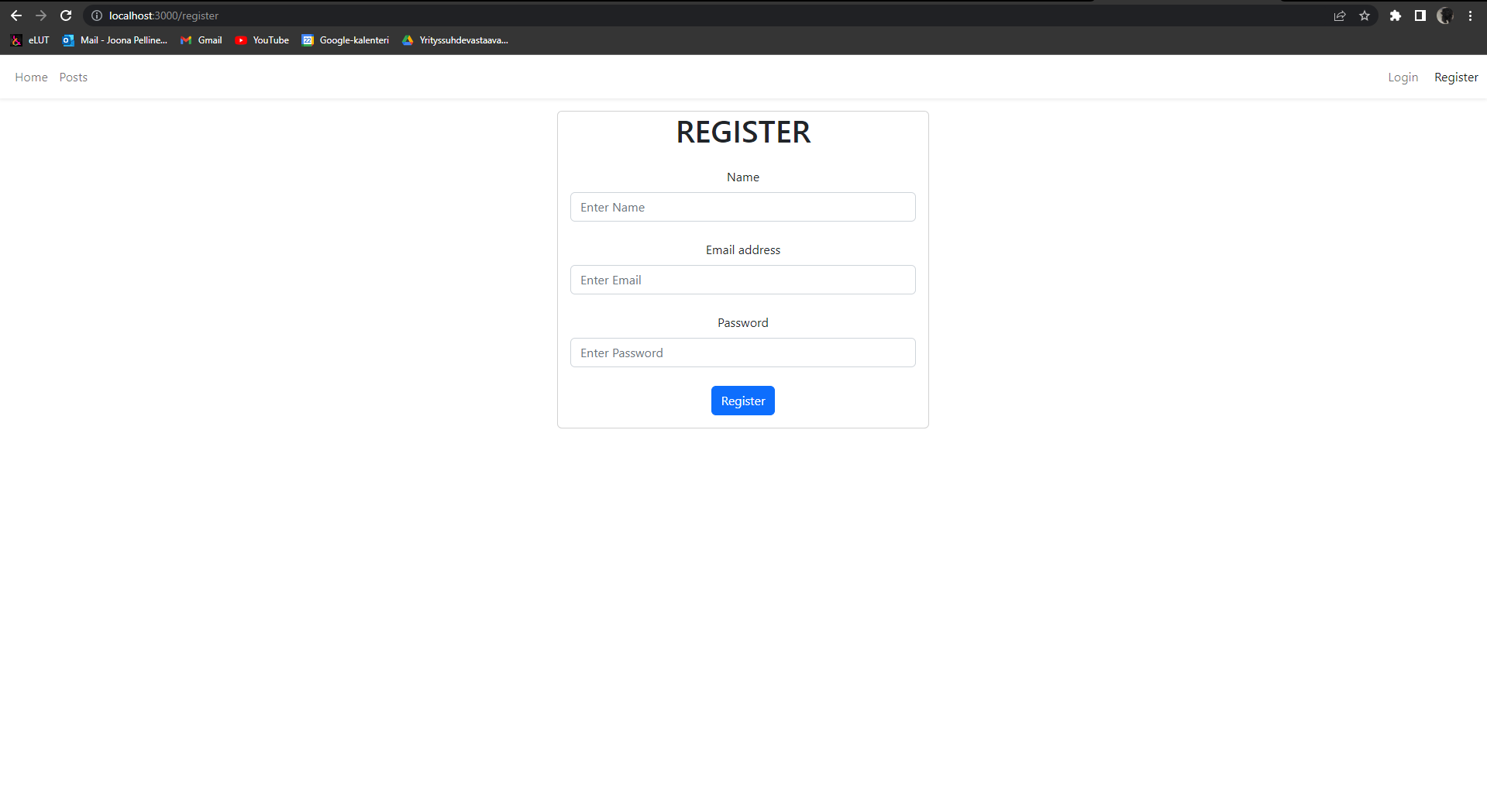
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

You can click a post and it opens the comments for that post:

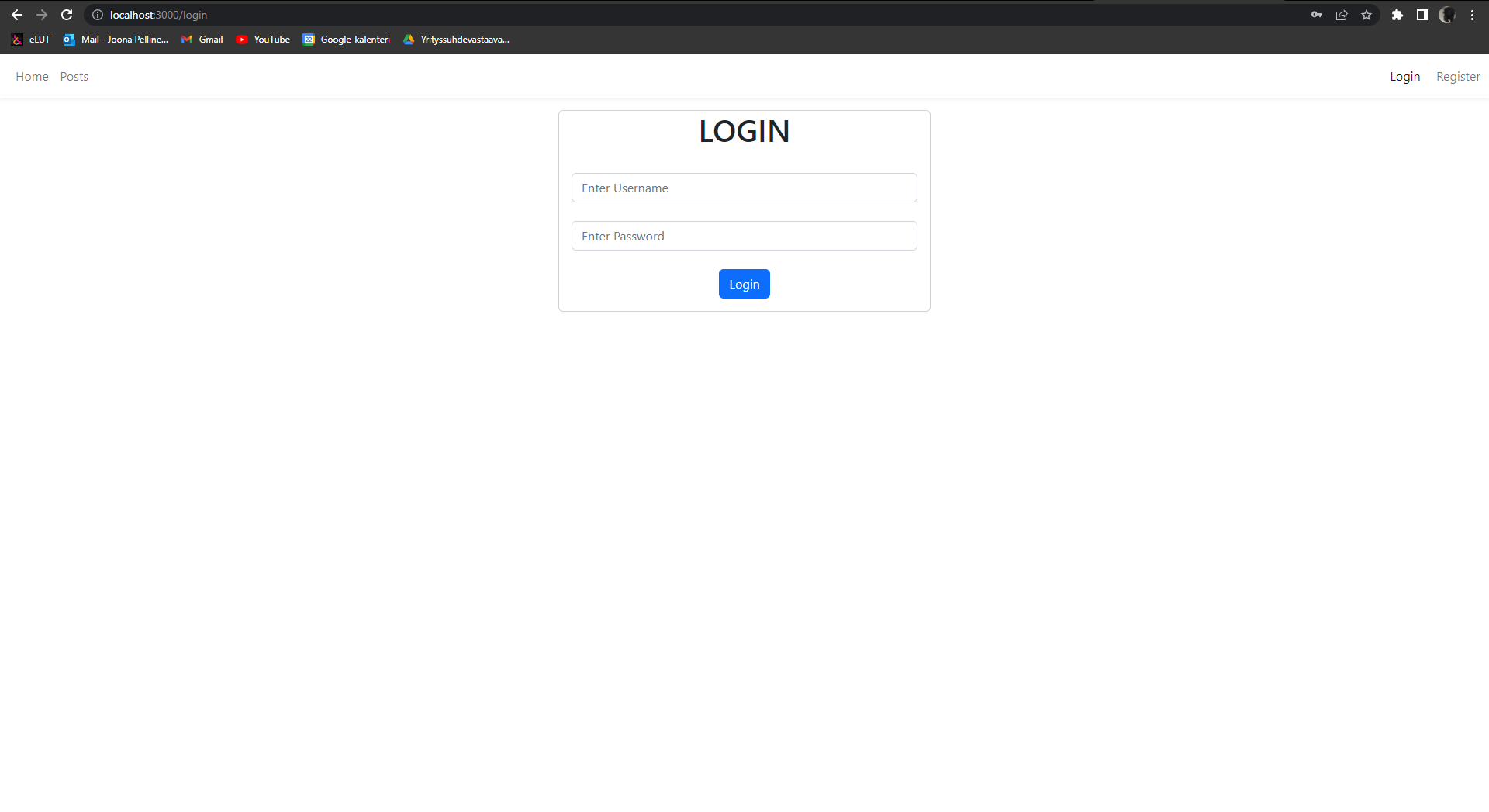


Let’s register to the website! Press the link “Register” to access registering page.



In the input fields please submit your username with an minimum length of 3 characters, email address in a correct email form (Example: [name@mail.com](mailto:name@mail.com)) and password with an minimum length of 5.

After registering, you can login by giving your username and password in the login page found behind the “Login” link.



After succesfully logging in, please do not refresh the page as it clears the authentication for some reason. The application still functions normally if you don’t refresh the page! When authenticated you can post and comment on the “Posts” page!

**List of features**

|  |  |
| --- | --- |
| **Implemented feature** | **Proposed points** |
| Mandatory requirements | 25 |
| Utilization of frontside framework. Used framework is React. | 5 |
| Translation of the whole UI in two or more languages | 2 |
| Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application? | 3 |
|  |  |
|  |  |
| Total proposed points: | 35 |