Advanced Web Applications

Project work: Finding New Friends Site (Tinder clone but with friends)

Aino Räkköläinen

Implemented features

Feature	**Max points**
Basic features (as stated in the previous chapter) with well written documentation	25
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5
User profiles can have images which are shown on the main page and in the chat	3
If match is being found the UI gives option to start chat immediately	2
User can click username and see user profile page where name, register date, (user picture) and user bio is listed	2
Sending timestamp of message is stored and shown in the chat	1

Technology choices:

I decided to use Express for the backend and React for the frontend because I have studied those during this course and taken a Code Camp course about React as well. Additionally, I used Figma a little bit for prototyping the user interfaces of the application even though I didn't implement my prototypes exactly. In the front-end, I used some Bootstrap 5 components and icons because those were easy to use and helped with creating consistent and responsive UI. Additionally, some react libraries like react-virtualized were used.

Installation guidelines:

Project can be installed by downloading the GitHub repository as Zip-folder and unzipping it. It can be opened in VSCode environment or in terminal. In both cases, running the project requires having two terminal windows open: one for server and one for client because I haven't managed to yet make a production version of the project which could be runnable with one command.

In conclusion, following commands needs to be run in terminal for running the project:

Server:

npm run preinstall

npm run dev:server

Client:

npm run install

npm run dev:client

User manual:

Registering and login

When the application is opened, the user gets a choice to either login or register. If the user has no account yet, they can first register and after that, they are redirected to login page if register information was correct and acceptable meaning that the email was not already reserved. The user does not have to fill out username because it is automatically generated based on the first part of the email user used to register.

Main features:

After successful login, the user sees the main page where for the first time, the user is requested to add bio text and profile picture for themself. Below this information box the user can then see other users one at a time and choose to like them by clicking thumbs up icon or dislike them by clicking thumbs down icon. If user hits like, that user is saved to their profile in a list of liked users and if disliked, the information is not currently saved anywhere so the application shows again those who user hasn't liked yet. One future improvement could be that the dislike button actually removes those

users from the user queue. By clicking the name of the shown user, the current user can see the profile page of that user.

Then if the other user likes or has liked the current user back, the UI notifies that match is found and shows a button for opening a chat with that user. Basically, it just redirects to the messages page where the chat component with specific user is opened automatically. The UI has a navigation bar where the user can open edit profile page where they can change email address, bio text or profile picture, or user can also open messages page where they can see the chats with other users or log out from the navigation bar.

Messages:

Messages can be sent between friends which are users that have both liked each other. Each message item shows the user who has sent it and when and the content. Additionally, profile pictures are shown within chat if users have those and if not, default icon is shown. Also in chat view, the user can click the name of the user they are sending messages to and check their profile page.

Sources for my code (also mentioned in the repository):

Client-side:

Creating scrollable component:
https://getbootstrap.com/docs/5.1/components/scrollspy/

How to pass data from child component to parent component in react:
https://www.geeksforgeeks.org/how-to-pass-data-from-one-component-to-other-component-in-reactjs/#approach-2-passing-data-from-child-to-parent-component

styling image in chat is based on this tutorial:
https://www.w3schools.com/howto/howto_css_chat.asp

Accepting only images in input field:
https://stackoverflow.com/questions/3828554/how-to-allow-input-type-file-to-accept-only-image-files

Uploading images to the page:
https://stackoverflow.com/questions/43692479/how-to-upload-an-image-in-react-js

Sending image to the backend: https://www.youtube.com/watch?v=-7w2KtfiMEM

Bootstrap 5 React documentation was used a lot as well.

Server-side:

Sorting lists by date in javaScript: https://bobbyhadz.com/blog/javascript-sort-array-of-objects-by-date-property

Comparing json objects: https://www.freecodecamp.org/news/javascript-comparison-operators-how-to-compare-objects-for-equality-in-js/

Uploading image received from front-end with Multer and Node.js:
https://medium.com/swlh/how-to-upload-image-using-multer-in-node-js-f3aeffb90657

Using regex to query specific string in mongodb

https://sparkbyexamples.com/mongodb/mongodb-check-if-a-field-contains-a-string/