Project Documentation

Technology Choices

MERN stack was used to make this project as I'm most familiar with it.

Frontend: React.js and styling was done with bootstrap and css.

Backend: Node.js with Express.js

Database: MongoDB was used as it was easy to use and good enough for this kind of project.

Authentication: JSON Web Tokens (JWT) was used to authenticate users.

Installation guidelines:

1. Clone the Repository: git clone 'https://github.com/JoonaManninen/dating_app_project.git'

2. Install Dependencies:

Frontend: cd .\my-tinder\ and then npm install

Backend: cd .\server\ and then npm install

3. Run the Application:

Frontend: npm start on .\my-tinder\ folder

Backend: npm start on .\server \ folder

also you need to have MongoDB installed.

User manual

You need to have new database made with MongoDB. This can be done changing the database location name in the app.js in the server folder line 17. You can choose whatever database name you like to insert in this: 'mongodb://127.0.0.1:27017/YOURDATABASENAME'. Then you can launch the project and create accounts using the register page.

Features in the application for users are:

Edit Profile: Users can edit their profile bio.

Like/Dislike Users: Users can like or dislike other users and potentially match with them.

Chatting: Users can chat with matched users.

List of features:

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
Use of a pager when there is more than 10 chats available available	2
User can click username and see user profile page where name, register date, (user picture) and user bio is listed	2
Last edited timestamp is stored and shown within chat	2
Total:	36