LUT-kauppakorkeakoulu

Advanced web applications

Project: Tinder Clone

25.02.2024

Veeti Engblom

Tinder Clone

Tinder clone made with MERN-STACK (MongoDB, Express.js, React and Node.js). In the app you can match, chat, and find common activities with your matches.

Technology Stack

Frontend

Framework: React.js
Libraries:
React-toastify: Used for custom toasts.
React-router-dom: Implemented for page navigation.
React-tinder-card: Utilized for swiping users.
React-cookie: Used for storing cookies from the server.
Mui/Material-UI: Employed for buttons and the navigation bar.
CSS and Media Queries: Utilized for styling and responsive design.

Backend

Libraries:
Database: MongoDB
ODM (Object-Document Mapper): Mongoose
Server Framework: Express.js
Authentication: JWT (JSON Web Tokens)
Password Hashing: Bcrypt
Validation: Express-validator
Unique ID Generation: Uuid
Environment Variables Management: Dotenv

Installation Guidelines

To install the application, check out the readme file for detailed instructions: https://github.com/veetiengblom/TinderClone

Technology Choices

Frontend Technologies

I chose React as frontend framework due to its popularity and its features. For the project I used multiple libraries for variety of reasons. React-toastify was used to easily create a custom toast. For routing I used React-router-dom that enables seamless navigation between different pages. React-tinder-card was used for swiping users left and right. React-cookie simplifies the process of storing and retrieving cooking from the server. Material UI was used for its pre-designed responsive UI components. CSS and Media Queries for styling and responsiveness enabling custom styling for the application.

Backend Technologies

I chose MongoDB as the database because it was though on the course. It also seemed like a solid choice for its flexibility and scalability. In the course we also used mongoose so I decided to implement it as well. It provided a solution for modeling the application data with the MongoDB database.
Node.js was chosen because it enables the use of JavaScript on the server side. This allowed me to use JavaScript in both frontend and backend. Express.js used for server routing and hosting.
For authentication I used JSON web tokens and for password hashing brcypt. To create unique user identifier I used Uuid. Express-validator for validating email and password. Dotenv vas used for secure management for environment variables.

Improvements

There is lot of room for improvement with this project. Proper testing is needed. There might be some bugs left in the code that did not occur in the simple testing I did. There is also a need for better way to add the activities to database.

Features

Feature	Description	Points
Basic features	Like/dislike, update profile, chat, JWT authorization, only authenticated users can post, responsive, documentation	25
Framework	React	5
Swipe	Can swipe left and right to dislike/like	2
Toast	Option to start chat after a match is found	2
Profile pictures	Profile pictures are shown in main page and chat	3
Timestamp	Messages show timestamp when they are sent	2
Activity features	Matches can find activities by swiping on selected categories. After a match is found it is displayed in the chat	6
Total points		45