

Project work

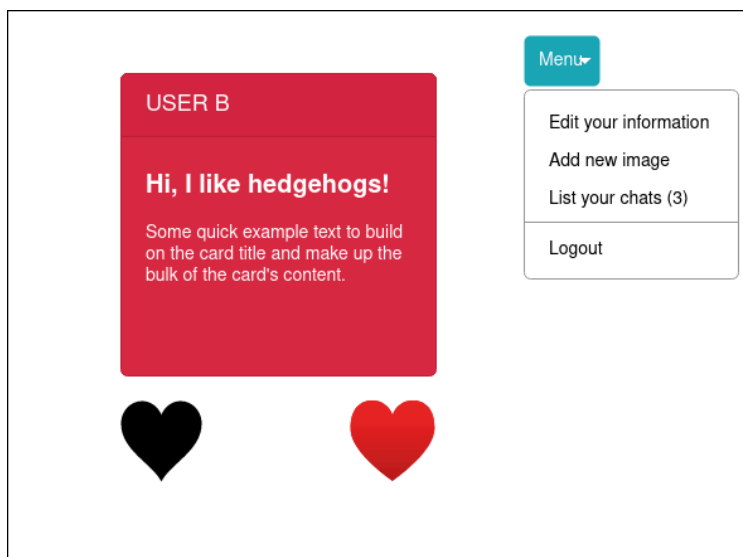
Last update 14.12.2023

Introduction

Your task is to implement a system that lets users register, login and like other users. If user A likes user B AND user B likes user A, they can start sending messages to each other. So in the essence your task is to implement a Tinder clone! Non-authenticated users cannot see anything. Data has to be saved to a database and usage of content management systems (CMS, e.g. WordPress or Drupal) is not allowed. Documentation has to be written. These basic features give 25 points from the project.

Besides the basic features you can implement as many additional features as you wish to gather more points and thus higher grades from the course. The maximum number of points one can get is 50.

The project is individual work and it could have views something like these:



A login and registration form. It includes fields for 'Email address' (with 'email@example.com' as a placeholder) and 'Password'. There is a 'Remember me' checkbox and a 'Sign in' button. Below these fields, there are links for 'New around here? Sign up' and 'Forgot password?'.

A chat interface. On the left is a list of users: 'SpamBot' (14), 'User D' (2), and 'James T Kirk' (1). On the right is a chat window titled 'Seven of Nine' with a close button. The chat window shows a conversation: 'Hi seven, how are you?' followed by 'Hi! I am fine. Are you looking for a friend to have some serious programming?' and 'Oh YES!'. Below this is the text 'If I share you my repo? Would you like to join?' and 'Resistance is futile'. At the bottom of the chat window is a 'Send' button.

Mandatory requirements

- Implementation of backend with Node.js
 - You can of course use Express, Meteor or any other additional framework
 - Java, PHP, Perl, Python, Ruby or other language solutions are **not allowed** in this course
- Utilization of database
 - MongoDB, MariaDB or any other you see fit
- Authentication
 - Users have to have an option to register and login
 - You can use JWT or session based authorization
 - Only authenticated users can post, comment (or vote)
- Features
 - Authenticated users can:
 - See other users and like / dislike them
 - Update their profile (images are not mandatory)
 - Chat with users that have liked both ways
 - Non-authenticated users can register
- Responsive design
 - The app needs to be usable with mobile devices and desktop browsers
 - Use of Materialize or Bootstrap is recommended
- Documentation
 - There needs to be documentation describing the technology choices, installation guidelines and user manual
 - Also list the features you have implemented and the number of points you are aiming at

Possible features

Feature	Max points
Basic features (as stated in the previous chapter) with well written documentation	25
The application does not work	-100
No documentation	-100
If some parts are missing (such as no database or authentication does not work) points are reduced	0 – -25
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5

One can swipe to left or right to dislike or like the profile	2
Use of a pager when there is more than 10 chats available available	2
Login with Facebook, Google, X or other accounts (use Passport.js)	2
Admin account with rights to edit all the users and comments and delete content/users (if a user is removed, all its chats should be removed too)	3
Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application?	3
Provide a search that can filter out only those messages that have the searched keyword	3
If match is being found the UI gives option to start chat immediately	2
User profiles can have images which are shown on the main page and in the chat	3
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
Translation of the whole UI in two or more languages	2
Create (unit) tests and automate some testing for example with https://www.cypress.io/ (at least 10 cases have to be implemented)	5
Inappropriate content, including hate speech related memes and other trash	-100
Code is not written and commented in English	-10
Code is not commented at all	-10
Code is not commented properly	-5
<Your own feature: describe why should it be accepted>	n

Notes

- Front and backend do not need to be under the same project/port/whatever, but they can be different implementations
 - If more than 60k ports are used, your project will not be graded

Returning

Return your code to CodeGrade via link in Moodle.