Project

General Comments

- Single person project
- Due on 6th of April

- There are no limits on the number/types of npm packages you use. However, make sure name and explain the use of the packages in the report and your video to avoid losing marks.

Goal: Channel-Based Chat Tool

- Design and implement a chat tool that allows users to
  - create channels
  - view all channel
  - select a channel and to post messages in that channel
  - post replies to existing messages

- The user interface must be implemented in Reactjs and all text data must be stored in a mysql database.

Part 1 - Basic System (40%)

- Design and implement the database tables
  - Expose the mysql database using a separate nodejs server

- Design and implement the user interface using Reactjs
  - Your application should use react-dom-router and provide at least
    - Landing page that offers a brief description/intro into your system
    - Page that allows users to see/create/select channel and see/create new messages or replies

Part 2 - Adding User Accounts (20%)

- Extend part 1 to enable users of your system to have accounts and enforce that only registered users can use your system.

- An account should consist of at least:
  - ID
  - Password
  - Name that is displayed in messages/replies/channels

  You can add additional elements e.g. avatar/image for the user

- Make sure that users can create an account & that users must always first sign in before using your system
- Create one special account for a system administrator (can be hardcoded) that has the power to remove users, channels, posts and replies.

Part 3 - Nested Replies & Ratings (10%)

- Expand/change your database and your Reactjs app to
  - Allow your users to reply to replies
  - Visualize the nested replies
- Allow users to approve or disapprove a post or reply
  - Use a thumbs-up/down with counter

Part 4 - Search Feature (10%)

- Allow your users to search the data in your system to answer questions like
  - content that contain specific strings e.g. list all content that contains the string “arrow function”.
  - content created by a specific user
  - User with the most/least posts
  - User with the highest/lowest ranking of messages/replies
  - …

What to submit

- docker-compose.yml file (and if needed additional dockerfile(s))
- All .js, .css files you created or modified e.g. App.js
- package.json files
- Design-Report: 1-page report describing your architecture/design decisions of your database and the react application
- Test-Report: 1-page test report demonstrating how you tested your system
- A link to a 10 minute video of you presenting your system and demonstrating the features of your system (worth 10% of grade)

Evaluation

- What parts have been implemented (fully/partially/incomplete)?
- Do they work?

- Part 1 = 40%
- Part 2 = 20%
- Part 3 = 10%
- Part 4 = 10%
- Design report = 5%
- Test report = 5%
- Video = 10%