

Mandatory exercise #1

Formalism

Course

Frameworks – fall 2016

Task

You must create a full-featured chat application that supports multi user communication. Requirements to the application are elaborated in this document.

Hand-in

The hand-in must include two parts:

- 1) An describing report that explains the thoughts behind design of frontend and backend solutions. The report must be uploaded as PDF document.
- 2) Programmed solution that can be opened and tested in webstorm.

Hand-in must take place, latest **Friday 7. Oktober 2016**

Evaluation criterea

To pass the course his hand-in has to be accepted by an evaluation that will test if frontend and backend technologies has been used as taught in the course. Evaluation will focus on checking if:

- the design of the solution is feasible
- frontend is simpel and flexible to cover all requirements
- instant backend-frontend communication is established
- NodeJS backend application offers a sufficient API to the client
- Mongo backend database contains all data and is organized to improve read performance

It is allowed to work in groups. However, all group members must hand in separate reports and programmed solutions. Hand-in of pure copies among group members is not acceptable.

Chat application functionality

Login process

On the welcome screen users register themselves by typing unique usernames. Then, hereafter the user enters the chat applications main screen.

Chat room selection

On the main screen a user can create a new or select an existing chat room from a global chat room list. Then all messages inside that chat room appears in a messages list that is loaded from the database. The author is noted next to each chat message.

Chat messages

Users have a field in which additional messages can be entered. When a message is entered it will immediately be stored in the database and pushed to all users looking at the same chat room.

Technology

Architecture

The application must consist of a frontend and a backend solution. The frontend is a website that for convenience is build using Pug template engine. The backend is a NodeJS application that uses Express and MongoDB.

Database

Express and MongoDB.

Database

The chat application must support 3 entities:

- Users
- Chat rooms
- Chat messages

Users, chat rooms and chat messages are stored automatically in the backend database.

Process guideline

Steps needed to solve the task would be:

- 1) Design the document based NoSQL database
- 2) Design backend service methods API
- 3) Design frontend user interface
- 4) Document the design in your report
- 5) Program database by using Mongoose
- 6) Program Express routing to implement the backend service methods
- 7) Program instant frontend-backend message connection.
 Use either HTTP-polling or WebSocket technology
- 8) Program frontend web pages
- 9) Document the programmed solutions in your report
- 10) Hand-in your report and programmed solution

During the process you can get supervision from the teacher.