Instructions

To make this exercise, you will need the Node.js runtime environment, which can be downloaded from https://nodejs.org/en/ and the Node Packet Manager (npm), which is included MongoDB database, which you can download from https://www.mongodb.com/download-center?jmp=nav#community or create an account on the free Mongo Atlas service https://www.mongodb.com/cloud/ atlas? jmp = download-center. (I created an account so the mail is [vanja.dimkovska@student.um.si](mailto:vanja.dimkovska@student.um.si) and the password is – mongodbatlas. I’m giving you this information so you can make the exercise as me.) First, install the Node.js environment and make sure that you can run the node and npm executables from the console. terminal. Node.js is also a web server, so you won't need a wamp environment for this exercise. For the exercise, select the folder where you will create your application. When you install the Mongo database, it will run in unprotected mode by default, and will require you to create an x: \ data \ db folder on the disk on which you installed it. Use the express frame and express-generator to help you build the skeleton of the application. Use a mongoose generator to make it easier to build models, controllers and routes, and handelbars, to create views. You can also use Bootstrap and other libraries for better and easier visualization.

Create a website that will allow logged-in users to ask and answer questions. First, you create a registration page where the user enters basic information such as username, email, and password. After successful registration, the user can log in and ask his question. The question has a title and a more detailed description, as well as any tags (sport, education etc.) regarding the content of the question. Also keep the date and time the question was asked. The user should be able to see the list of all the questions he asked, and see the answers to each question. The user can choose an individual answer as the accepted answer to his question (pin the question). The registered user can also answer the questions already asked by other users. Each user (even the unregistered one) can see the list of all questions asked, which should be sorted by the date of the question from the newest to the oldest. When displaying an individual question, the answers should be sorted by time, but if the user has already chosen the one that he thinks is correct (has pinned the question), it should be written immediately below the question. When printing questions and answers, also print the time of asking the question or publishing the answer and the username of the user who asked it. Also, allow users to delete their question or answer.

Allow users to rate responses (upvote, downvote). When displaying answers, it also displays the grade of the answer (number of likes and dislikes).

Enable question search by question tags.

Make a special section hot questions where the questions will be sorted according to the activity of the users in that question. As an activity, you measure the number of views of an individual question, according to a unit of time (5 minutes, one hour, ...).

Provide user profiles where you add a profile picture to users and information about their activity (number of questions, answers, and number of selected answers). When displaying the username, display a thumbnail profile image for questions / answers.

Allow comments on questions and answers to questions.