19/01/2018

Note Taker: Ruochen

<u>Topics Discussed</u>

The navigation bar should contain the following elements

1. Home
2. Ask [Form submitted though a POST, and will redirect to signup/login page if not logged in -> this requires sessions]
3. Sign-up [For now, only *email* and *password* are required, all other database fields are written as -1 or null]
4. Login

The following are the *three* initial databases required, and their appropriate fields

1. Database 1 – The user database
    a. Username
    b. E-mail
    c. Password
    d. Score
    e. User_id
2. Database 2 – The question database
    a. Question_id
    b. Question Title (String)
    c. Question Body (String)
    d. User_id (of person who asked question)
    e. Date (when person asked question)
    f. Num_answers -> the number of answers for a given question
3. Database 3 – The answers database
    a. Question_id (id of the question being answered)
    b. Date (of this particular answer)
    c. User_id (of user who answered)
    d. User_id (of person who asked question)
    e. Answer (the answer given for this particular question)
    f. Answer score (how many up/downvotes the answer has)

The following is a breakdown of each page:

Home/Landing page:

Should include a list of questions, perhaps the top 10 or more questions with the most answers (see database 2 for this field). Each entry in this list should contain the name of the user who posted the question, the date when the question was asked, and question title itself (perhaps with a limited number of characters when displayed) and the number of answers for the question.

Login Page:

At the login page, there should be a simple interface with entries for both Email and Password. Behind the scenes, javascript will perform basic validation if the client requests it. At this point, there should only be a hard character limit, and a built-in html check that the email is valid. Once the login information is submitted, the user is either logged in (and redirected to the home page) or attempted an invalid login – at which point the user will redirect to the same page with an error message

Signup Page:

The signup page should mimic the format of the login page to start, and later it will be modified to look like its own entity. On this page, the user will enter an email and password for the account creation process. Behind the scenes, Database 1 will be queried to see if an account already exists with this email… If so, redirect to the same page with an appropriate error message, else redirect to a blank page with an *Account Created Successfully* message. Later, the user will be redirected to the home page with this same message.

Ask Page:

There should be two text boxes, one for the question title, and one for the question body. To keep things simple in the initial design, the *question title* will be limited to **150 chars**, and the *question body* will be limited to **1000 chars**. To start, the submission of these entries should only make sure that the question contains text, and then sent to *database 2* for storage with all appropriate database fields filled.

Sessions (old php nomenclature)

- There should be a way to have a session (from php, must find node.js equivalent) be active across all webpages in order to hold persistent user data

Task allocation

- Tom and Nizar are both creating their own html/css versions of the webpage without embedding any javascript yet. The design will start as one html file per webpage, but it will soon be refactored to have each section of the site in its own html file, loaded dynamically as needed.
- Saleh + Abishand → will work on understanding how to set up, read from, and write to databases using Node.js, express, and MySQL (and try to test populate a html file with some data in this way)
- Alex + Jason → See Saleh and Abishand's task above
- Anthony + Ruochen → Will have to write up the User Requirements/User Stories and the subtasks/requirements within them. This was made clear by the TA during the tutorial session. Also → work on databases to raise familiarity.

Next Meeting

The next meeting will take place either Tuesday @ 5:30 PM or Friday @ 10:00 – 1:00 PM.