DAILY KOS

Harper Sandford - Senior QA Homework Assignment

- Please describe your test process for an update to the login software for Daily Kos.

  Your answer might be a combination of manual test instructions, plans for automated tests, and/or use of external tools. Your answer will be evaluated for clarity, completeness, and insight, though we don't expect you to accurately guess any of the custom tweaks we've made to the system over the years.

  Daily Kos uses the Ruby Devise gem to manage user accounts, with extensions such as devise-encryptable, devise-lockable, and devise-security. Each user also can be managed via a privileged admin area.

  Assume that we are upgrading our login to include the following new features:
  - The ability for administrators to force-expire a password older than a variable specific date and walk the user through selection of a new password
  - 8 character minimum for new passwords
  - The ability to lock a user's account for 2 hours after 5 failed attempts

  Assume there are no existing automated tests from the front end for any aspects of the user account.

  Make assumptions about how things work and what features are enabled as needed, and what the configurations are other than what is written; if you make assumptions, clearly call them out.

  There is no need to be repetitive, but it's appropriate to explain choices you make for a particular approach or why you chose automation or manual testing for a particular task. We are looking more to understand how you approach the problem and what considerations you use rather than expecting any particular answer.

If you want a place to safely experiment with the current feature where you can create users and write data without bothering anyone, you are welcome to use our public beta server at https://www.dailykosbeta.com

- Please choose **one** of the following below. You do not need to complete all three. You can select any of one the following:

    1. Find two bugs on the dailykos.com website. Write bug reports for us with as much information as you think we'll need to reproduce the bug.

    2. Visit https://www.dailykos.com/groups/Good%20News, this is an example of one of our Groups pages. Analyze this page as if you were going to create a testing plan with a mix of manual testing and automated integration and acceptance tests. Sketch out a test plan highlighting what would be tested at each testing stage, and be prepared to explain why you chose those options in the group interview. For extra credit, write a simple acceptance test in the language of your choice around one or two elements on this page. We don't need to see the code work, but we do want to see how you approach test creation, organization, and structure.

    3. Examine the structure of this imaginary database. There are three tables: user, group, and user_group .

        - The user table contains the following columns: user_id, first_name, last_name, email, join_date
        - The group table contains the following columns: group_id, group_name, created_at
        - The user_group table contains the following columns: user_id, group_id

        For the purposes of this question, assume a user may only belong to a single group, but a group may have unlimited members

        Write three SQL queries which would complete these tasks:

        - generate a list of all users who joined in the last month in the Cat Lovers group
        - generate a list of all users with an @yahoo.com email address who are in the Unicorn Conspiracy Theories group
        - generate a list of all groups created before 1-1-2016 with zero members

        Provide these queries in this document, as a link to a repo, or as plain text, whatever works best for you.