Scuba-Doo Dog Erasers (Karen Shekyan, Gabriel Thompson, Russell Goychayev)
Softdev
P00 -- Half Quick
2022-10-27
time spent: 1.8 hours

QCC

- How can we efficiently store differences between edits (edit history)?
- Is there a better way to store a "list" in SQL?

Components:

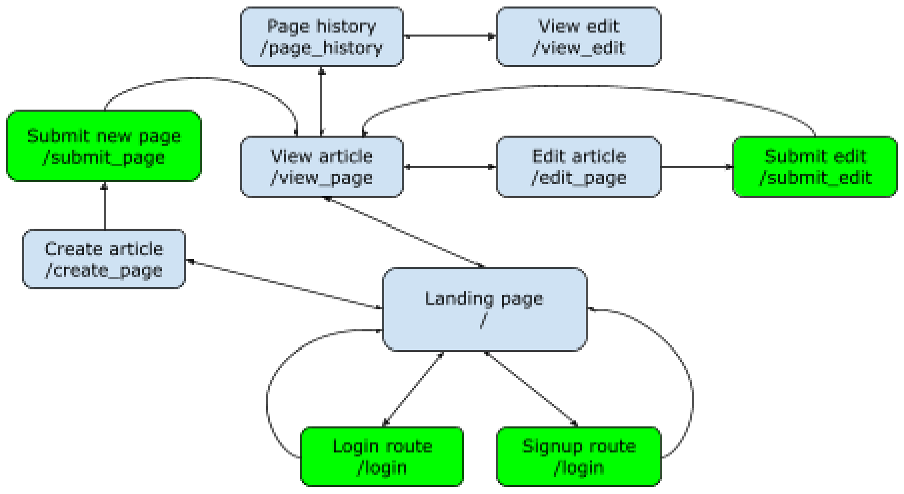
- User accounts
    - User sessions
    - A log-in and log-out system
- A route that allows users to create new articles
- A route that allows users to edit articles
- A SQLite database that stores the contents of articles, the edits made to them, and user account information
    - A hash function to encrypt passwords
- A route to serve the contents of any wiki page
- A landing page with a button allowing you to log-in to the website and sign-up for website
- Templates for each of the pages

Database Organization;

- 3 tables:
    - User info
      `create table user_info(username text, password text);`
        - This table will be used when logging a user in, to verify that their username matches their password
        - `username` is unique, so we can match edits to their creator and there's no namespace conflicts
    - Edits
      `create table edits(id int, username text, revision_content text, timestamp int);`
        - This table will be used to keep track of edits.
        - `id` is unique, and is used to identify edits in conjunction with the pages table.
        - `revision_content` is used to keep a history of the edits
        - `timestamp` is used to keep a history of the edits
    - Pages
      `create table pages(title text, content text, edit_ids text);`
        - `title` is unique
        - `content` is the html of the article
        - `edit_ids` is a comma-separated list of edit ids for all of the edits made to the article. For example, if edits #200, #564, and #1337 are the only three

edits to the article, then the content of the `edit_ids` would be "200,564,1337"

Site Map:



Key:

- The format for each box is the title of the page, followed by its route
- Two-way directional arrows mean that two pages are accessible from one another
- One-way arrows mean that one web page redirects to another
- Blue boxes mean that this route is accessible via GET request
- Green boxes mean that this route is accessible via POST request
- **Important thing to keep in mind:** Every GET-accessible page has a link back to the root of the webpage (sort of like how Wikipedia has the clickable logo in the upper-left)

Description of each page:

- / — The landing page. Contains a logo and the login and signup forms. Contains a search box where you can type in the name of every article, and the server will return the content of the article via /view_page. Contains a "Create Article" button.
- /login — The login route. This is the route that the login form on the landing page sends the information to
- /signup — The signup route. This is the route that the signup form on the landing page sends the information to
- /view_page — Contains the content of any requested page (the title of the page is specified within the URL arguments). Contains links to "Edit page" and "Page History" buttons
- /page_history — Contains a chronological list of every edit on one page. Each edit is a link, which links to the /view_edit route
- /view_edit — Contains the appropriate older version of the appropriate article, given the ID of the edit requested

- /edit_page — Contains a form that allows you to type in the new contents of the article you're editing. When you press the submit button on the form, the information gets sent to…
- /submit_edit — Updates the database with the new article content
- /create_page — Sends the user to a form where they can enter the title of their new page, as well as the starting content of it. When submitted, the info from this form gets sent to…
- /submit_page — Updates the database with the new page submitted by the user

List of templates we'll need:

- landing_page.html
- view_page.html
- page_history.html
- view_edit.html
- edit_page.html
- create_page.html

Assignments of each task to each group member:

- Karen:
  - user_info tables
  - edits tables
  - pages table
- Gabriel:
  - Templates
  - /view_page, /edit_page, /create_page, /view_edit routes
- Russell:
  - /login route
  - Hash function for encrypting passwords
  - Landing page