Feature - Display of Previous Versions

- Programmer: Elizabeth Gieske
- Date: 04/17/2018

Problem Definition

Necessary portion for the archive Database system. Edits of the markdown files aren't currently documented in any fashion and there is no way of reverting or tracking file changes. Users have requested the feature for having the data of a markdown file change to be tracked in a SQL database. This allows for many future feature additions such as reverts, displaying to the user the changes and so on.

There is no way for users to view past version of pages. Even if the old versions are stored, without a user interface, there is no way for the user to interact with the previous pages of the wiki.

Requirement

Previous versions of wiki pages need to be displayed in an organized and user friendly way.

User should be able to select which version to revert back to.

User stories

1. As a user, I want to select which previous version of a wiki page so I can view that version.
2. As a user, I want to be able to view information about previous edit so I can know who made the edit and when that user did.

Design

User story 1, 2

Interface

- Template name: history.html
- Input: the parameter is a dictionary. Key is the page name. Value is an array of HistoryPage(s)

- Action: all information on pages are displayed in an accordion with nested forms
- Output: user selection of page name and commit id (separated by comma)

Detailed design

- To trigger access the History template, user can select the "History" tab along the wiki header
- History page contains traditional header and layout like every other wiki page
- Page creates an accordion and creates cards by looping through the key. Each card has collapsible content made up of the array of commits from the value of the dictionary

Implementation

- Template uses bootstrap to implement the accordion and cards
- Template uses Jinja2 to loop over dictionary and arrays

Interface

- API name: modified core.py
- Function name: searchHistory()
- Input: N/A
- Action: Pulls and formats data from Postgresql database
- Output: returns dictionary of formatted page names and commit information

Detailed design

- Function call on ArchiveDatabaseConnection to a cursor to grab necessary data
- Uses the cursor to execute 2 SELECT statements to grab all page names and commit information
- creates HisoryPage(s) from commit information
  - Key = page name
  - Value = array of History Pages

Implementation

- 'from wiki.web.ArchiveDatabaseConnection import ArchiveDatabaseConnection' - to interact with database
- utilizes HistoryPage class - to store commit information cleanly
  - class stores page name, commit id, timestamp, and user
- runs requests through router.py to render template and redirect user
  - occurs within history()

User story 1 & 2

The Python code for implementation can be found at https://github.com/tiemonl/CSC440WikiWiki.

Tests

- This test uses 'psycopg2' to check the number of items that should be present in the dictionary generated by function 'searchHistory'.
- The tests for the added features are included in the test features folder. This test is can be found from https://github.com/tiemonl/CSC440WikiWiki.
- In this test case, a dictionary of history pages is created from searchHistory and the length is asserted to be the same as the number of distinct pages in the database.

Checking the requirements

- User story1 - Given that this is a UI feature we must test this manually.
- User story2 - Test results shows that the user is properly being displayed all the data requested.

Limitations of tests

1. This test checks only if the dictionary is properly being constructed. We must test the rest of the UI features mannually

User documentation

Users can use the Website to interface with the UI in the History page. There, they can interact as specified earlier.