Team threeCoffeePeanuts: Jesse Xie, Yaying Liang Li, Ryan Wang

SoftDev

P00 -- Half Quick

2021-10-25

Program Components:

- Flask
    - Flask will host our code onto a localhost, in which it will display html code from our templates folder onto various pages
    - The html code will be formatted differently and will show data from our data folder, which will contain .db files which will be created through inputs on the website and sqlite3 as tables
- Python
    - Various python modules will be used accordingly in order to utilize flask and sqlite3. In other words, python will serve as the middleware, exchanging data between flask and sqlite3 to reconstruct a working website
- SQLite3
    - Sqlite3 will create data tables in a db file which will be stored in a data folder
    - The data will be obtained from inputs in forms on the page generated by flask

Database organization:

- All of our db files will be stored in a data folder
- We will store the usernames and passwords as a single table created in a db file called users.db (username TEXT, password TEXT)
- We will store all of the changes made to page in a db file called log.db
    - This db file will have a master table that contains all of the changes made by users (username TEXT, time TEXT, comment TEXT)
    - Other tables will be made that contain the changes made to individual pages
    - The data will be used in the logs page that is generated by logs.html in the templates folder
- We will store of a website's text in a db file called websites.db
    - This db file will contain various tables that are named after the title of the pages
        - These tables will contain data that concerns the formatting of the text (text TEXT, size INTEGER, formatting TEXT, …)
            - Images and hyperlinks to be decided
    - The data will be used in pages that are generated by website.html in the templates folder

Features:

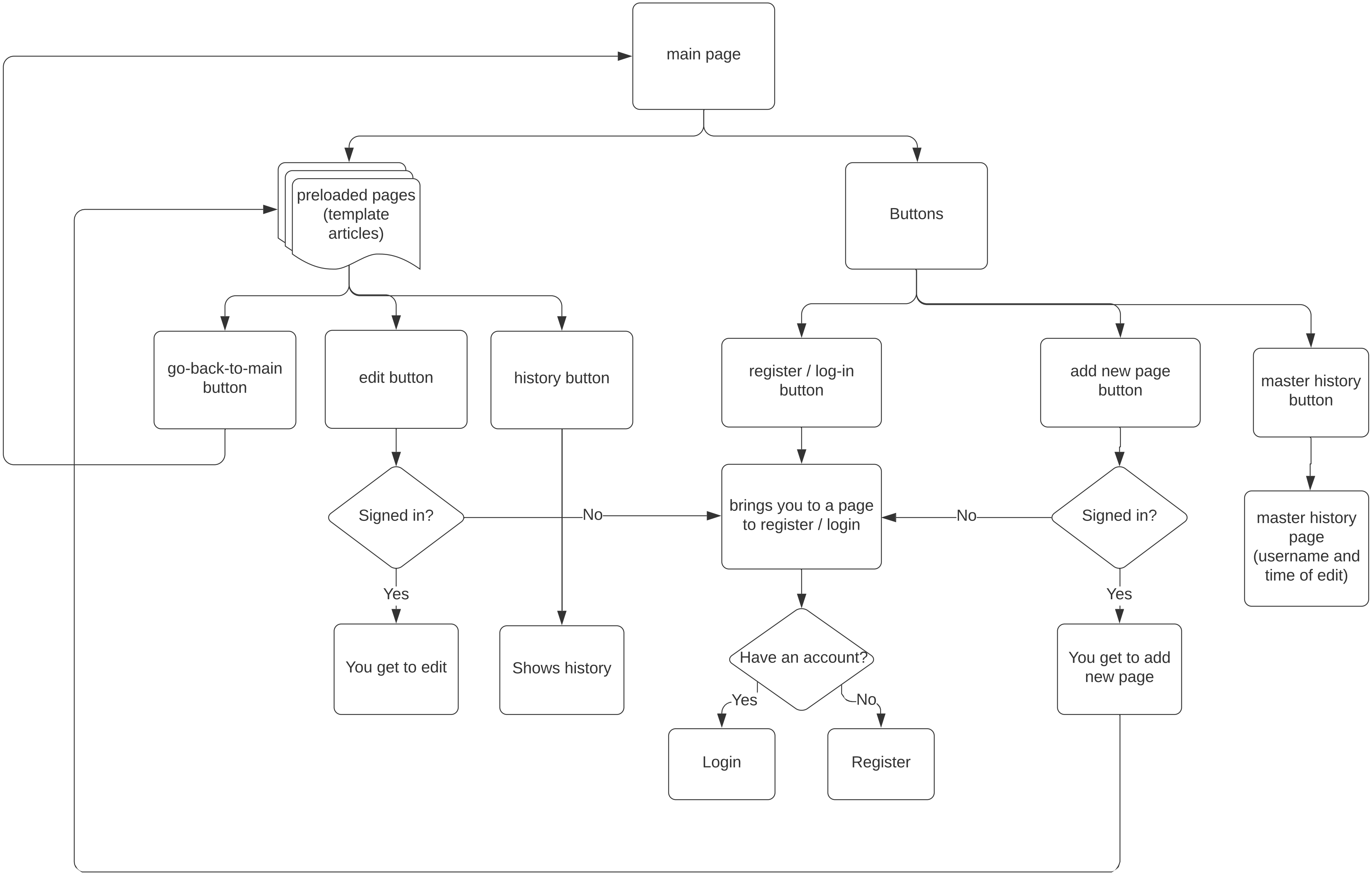
- User lands on site!!!
- There is main page with all the links to the other pages (articles); register button; log-in button; edit button (can only edit if you are logged in); log of edits (keep track of user + time they made edit) per page
    - Html formatting of all pages similar
    - Edit button: if logged in, take to edit page; if not, take to log in page

Checklist:

- Main page [Yaying]
- Preloaded pages [Yaying]
- Buttons
    - Register button [Jesse]
    - Login button [Jesse]
    - Add new page button [Yaying]
    - Edit button + page [Jesse]
    - Go-back-to-main-page button [Ryan]
    - History button (master log and per page) [Yaying]
- Sessions (keep track of who has account/who is logged in) [Ryan]
- Databases [Ryan]
- Logs [Ryan]
- Put data from db files onto template pages [Jesse]

Our Wiki Plan

Three Coffee Peanuts: Ryan Wang, Yaying Liang Li, Jesse Xie


main page
preloaded pages (template articles)
Buttons
go-back-to-main button
edit button
history button
register / log-in button
add new page button
master history button
Signed in?
No
Yes
You get to edit
Shows history
brings you to a page to register / login
No
Signed in?
Yes
master history page (username and time of edit)
Have an account?
Yes
No
Login
Register
You get to add new page