Ivan Yeung Daniel Yentin (Team Success)

Soft Dev

Storytelling game/website

2022-10-31

Target ship date: November 14, 2022 (2022-11-14)

1. Program Components

- **a.** Flask
  - i. /templates
    1. login.html
       - a. This page contains a form for logging into your account
       - b. button to register.html
    2. register.html
       - a. This page contains a form for registering a new account
       - b. button to login.html
    3. home.html
       - a. This is the landing page that users go to after successfully logging in.
       - b. There will be buttons for creating, adding, and viewing stories
       - c. There will be buttons for each story for the add and view function that will allow users to choose which story they want to add to/view.
    4. create.html
       - a. This page will have 2 forms. One form will be used to give the new story a title and the other form will be where the user can write the content of the story.
    5. add.html
       - a. This page will 1 have a form where the user can enter their addition to the story. On this page, users will also be able to see the last contribution(only the last contribution and no other part of the story) to the story along with the title and publisher of the story at the top.
    6. view.html
       - a. When the user chooses a story they want to view, they will be directed to a page with the title of the story, the author(creator), and the content of the story(latest version)
  - ii. /static

1. login.css
2. register.css
3. home.css
4. create.css
5. add.css
6. view.css

iii. app.py - the python file containing all the python to flask and SQLite3 interactions

1. @app.route("/"):
   a. redirect to /login
2. @app.route("/login"):
   a. renders login.html
3. @app.route("/login/auth"):
   a. login form: username & password
      i. check for existence of username and validity of password
4. @app.route("/register"):
   a. renders register.html
5. @app.route("/register/auth"):
   a. register form: username & password
      i. check for availability of username
      ii. if account is successfully created, information is stored in database
6. @app.route("/home"): then displays buttons to start a new story or add to an existing story
   a. New Story: button
      i. Takes user to the create page
   b. Add to Story: List of buttons
      i. There will be a button for each story that user has not contributed to and the button will take user to the add page
   c. View Story: buttons of all stories that user has contributed to including stories that they have created
      i. Takes user to view page with the information of story they chose(title, content, publisher)
7. @app.route("/logout"): clears the cookie(session) and redirects users to the login page.
8. @app.route("/create"): renders a page with two empty text boxes where the user can add a title and write their story and upload it to the database

9. @app.route("/publish"): Checks the title of the story that the user inputted in the create page. If the story name has not been taken, it inputs the title and body form into the database.
10. @app.route("/add"): renders a page with the title, publisher, last contribution to the story and a text box where user can submit their own addition
11. @app.rout("/save"): Inputs the new content into the story and sets the new contribution as the latest contribution.
12. @app.route("/view"): renders a page with the story that user selected from homepage

b. SQLite3

- i. Used in app.py to store data that users provide as they navigate through the site
- ii. .db file holding the tables specified in #3 Database Organization
- iii. User ids and Story ids are created by incrementing an int starting from 0

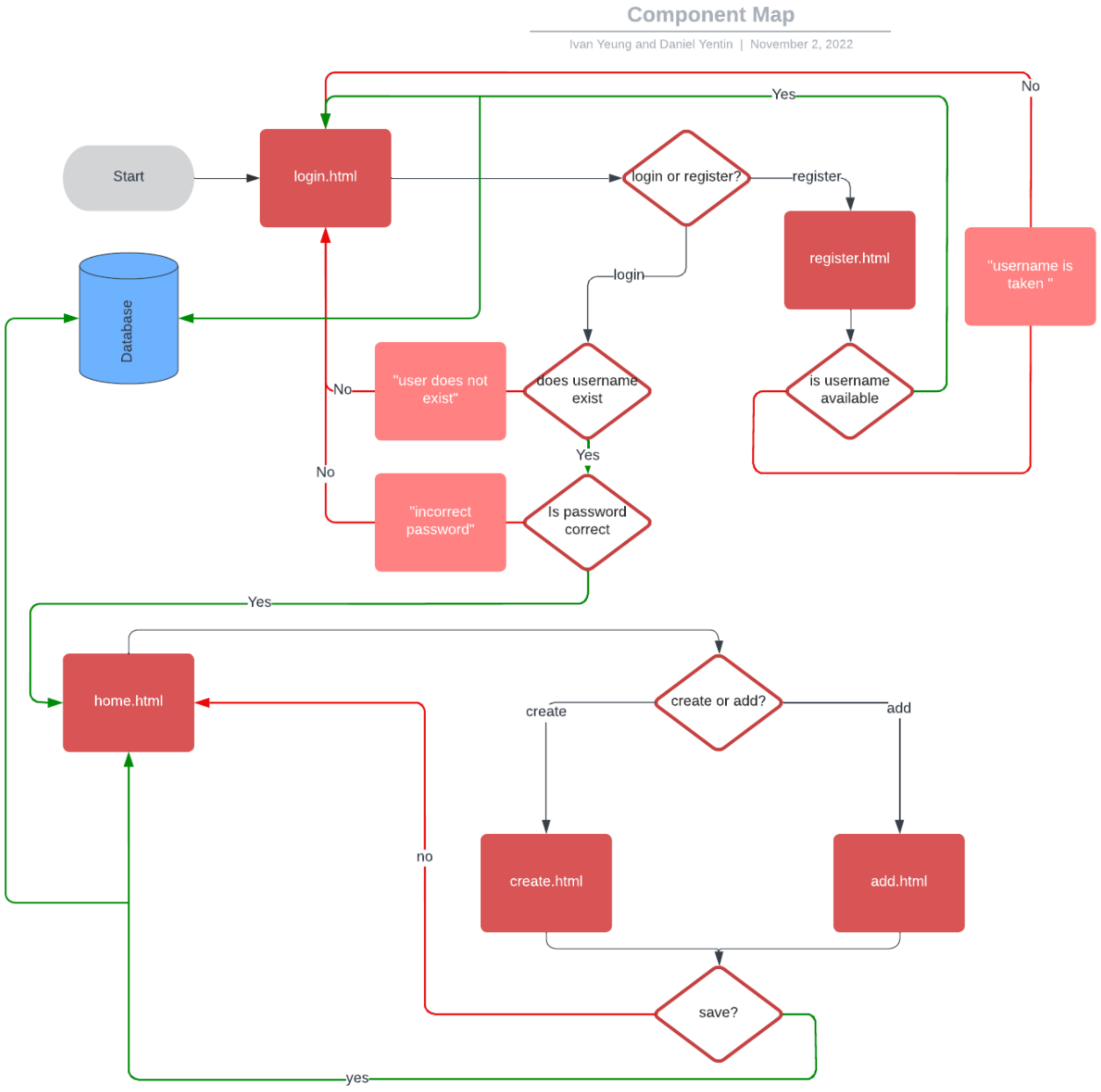
c. Sessions

- i. Session["username"]: Stores the username of the user that is logged in
- ii. Session["error"]: Holds an error(such as incorrect password when logging in) for when the page is rendered
- iii. Session["stay_logged_in"]: information indicating whether the user checks the box when logging in to stay logged in. Checking the box will automatically login the user when they go to the root page by saving their cookie after logging out.
- iv. Session["title"]: Holds the title of the story when a user is creating their story. If their title is invalid, the page will be rendered again with an error without wiping out the work that user had.
- v. Session["body"]: Holds the body of the story when a user is creating their story. If their title is invalid, the page will be rendered again with an error without wiping out the work that user had.

d. Helper functions

- i. P_redirect(url): returns redirect to the route passed into the function and automatically adds the 307 code to the redirect function.
- ii. find_uid(username): returns the uid of the inputted user
- iii. find_sid(title): returns the sid of the inputted story title

2. Component Interactions/Component Map



3. Database Organization

a. stories: | title | body | publisher | sid

b. <sid>: | contributors | contributions |

c. users: | username | password |

d. <uid>: | stories |

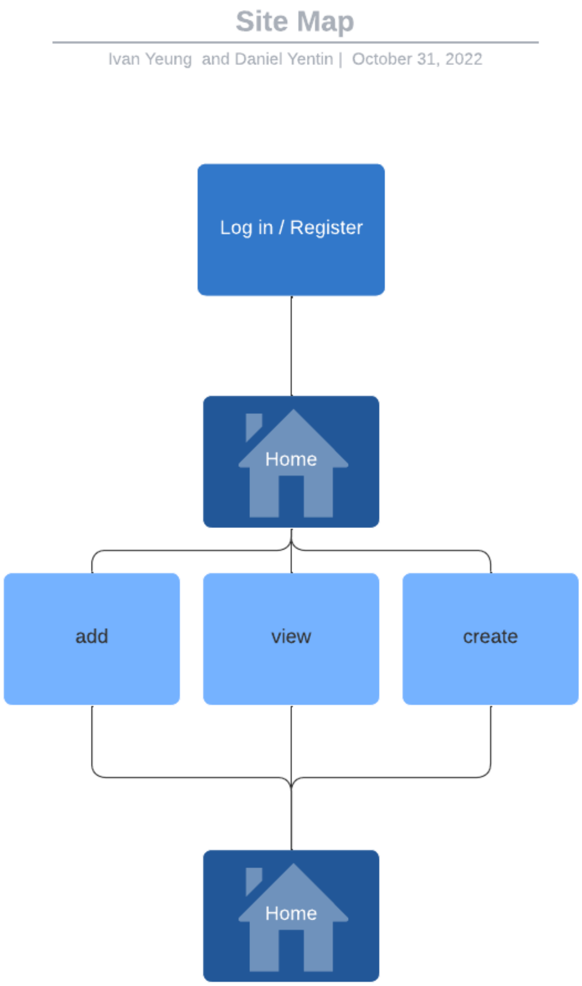
*all fields are of the TEXT type

*<> indicate a variable to be replaced.

* <story> would be replaced by the story's title

* <user> would be replaced by the user's username

4. Site Map



5. Task Breakdown

To be decided once the project starts. We do not know how long each part will take, so if one person finishes one part quicker than the other, they will move on to the next part.

Priority:

1. login.html & register.html
2. /login & /register
3. home.html & create.html
4. /home & /create
5. add.html & view.html
6. /add & /view
7. *.css files