POPEYES: Lauren Lee, Ian Jiang, Vivian Teo
softdev
p00 : Story Creation
2022-10-31
time spent: 28 hours
target ship date: 2022-11-15

Scenario One: Your team has been contracted to create a collaborative storytelling game/website.

- Users will have to register to use the site.
- Logged-in users can either start a new story or add to an existing story.
- When adding to a story,
    - Users are shown only the latest update to the story, not the whole thing.
    - A user can then add some amount of text to the story.
- Once a user has added to a story, they cannot add to it again.
- When creating a new story,
    - Users get to start with any amount of text and give the story a title.
    - Logged in users will be able to read any story they have contributed to on their homepage (the landing page after login).

Scenario 1:

Components:

- __init__.py – define routes of pages
- login.html – user can login or register
- db.py – python file for writing SQL and organizing the databases
- users.db - table storing user information
- edits.db - table storing story information
- home.html - users can choose to create a new story or create new stories on this page.
- stories.html - template that’s able to hold different topics/stories

- edit.html - users can edit old stories on this page

How each component relates to each other:

- __init__.py - host the web pages locally.
- login.html - use db_users.py to check if user has an account stored in database
- home.html - displays all the stories a user has contributed to, and has a table of contents of all the stories hyperlinked. There’s also a form for creating new stories.
- db.py – adds/extracts information into/from users.db and edits.db
- edits.db – contains info on stories user edits after db.py is run
- users.db - have individual user information like their logins that will be checked when the user logs in or registers.
- edit.html - send information to db.py to be put in edits.db and users.db (list of stories a certain user contributes to)
- stories.html - get information from db.py to display the stories

Database Organization:

- users.db: user id | username | password | list of story ids user has contributed to |
- edits.db: story id | story title | content | latest change |

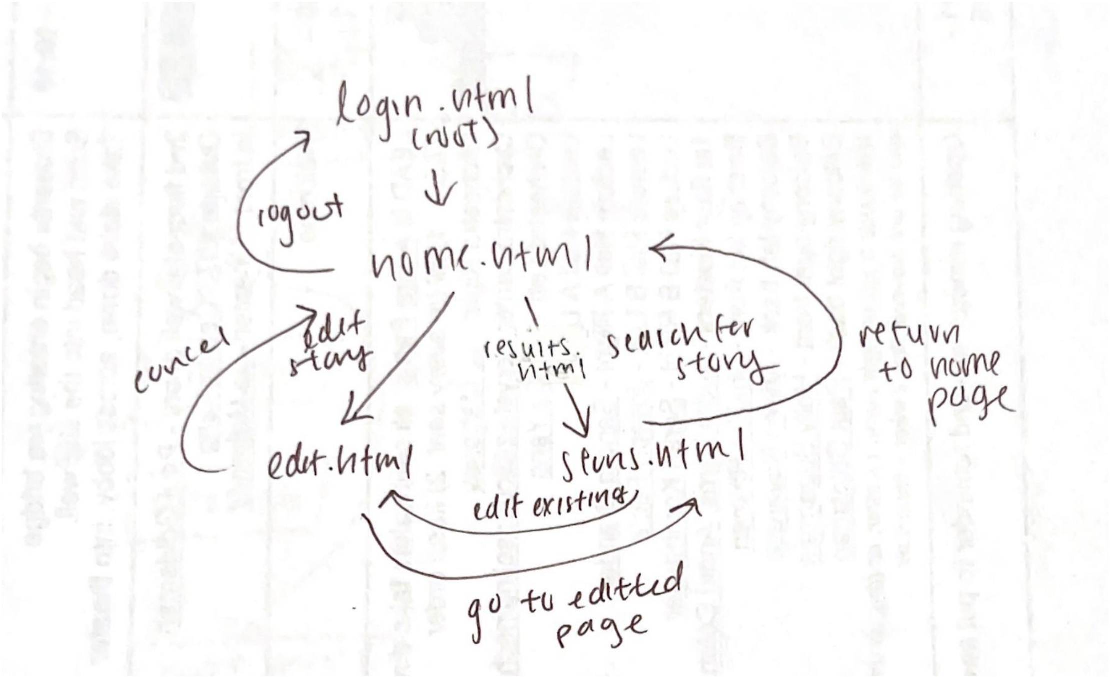
Site map for front end:

- /
  - login page – login.html
  - Form submission for logging in or registering
  - Registering creates new data in the table users.db
- /auth
  - When a user tries to log in and clicks the submit button, the /auth route is taken where we check if the login information is correct; if it is, then the user is taken to home.html. If not, the user is asked to

try again with the correct account information in login.html

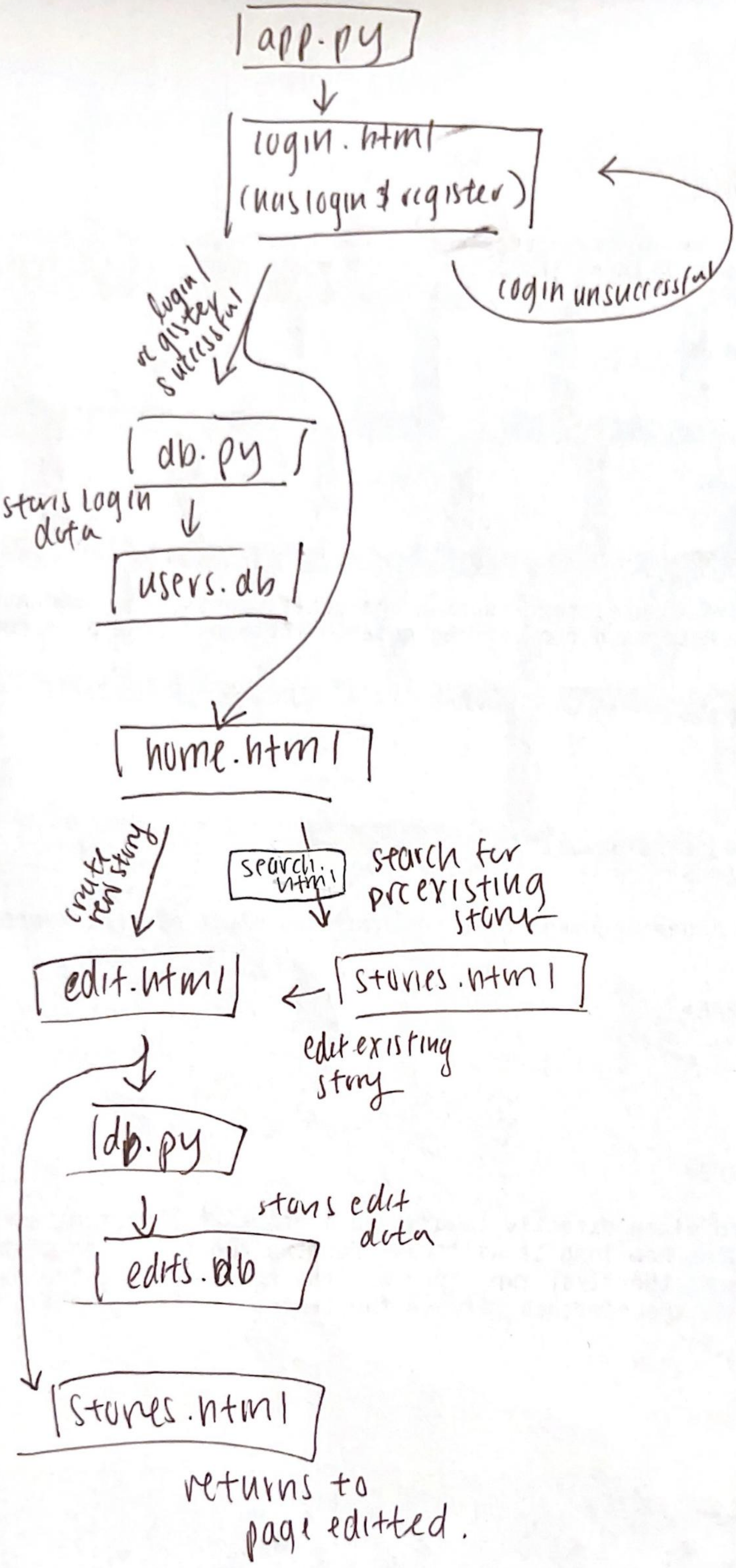
- /register
    - When a user tries to register an account and presses the submit button, the /register route is taken, and we check if the username is available. If it is, then an account is made and the respective information is stored in users.db. If the username already exists, then a message tells the user that the account already exists, and the user stays in login.html.
- /home
    - Once logged in, the home page is displayed (home.html)
    - Home page consists of all the stories the user has contributed to, scroll-based format
    - Able to search for a story via a form; submitting form takes user to the /results route
    - Option to create new story (button), which takes user to the /edit route
- /stories/[story id]
    - Page that displays the content and title of a story once the user clicks on a hyperlink in home.html
    - Page where user can edit an already existing story. If a user has already contributed to a story, they cannot contribute again (checked by looking at the list of story ids the user has contributed to in users.db.The story id is added to the list in users.db if not already in. User cannot delete content, only add.

- /edit
    - Route that's taken after a user submits an edit in the /stories route, allowing data to be put into users.db and edits.db, and the user is eventually taken back to home.html



Breakdown of tasks:

- Lauren: edits.html, stories.html, storing data into edits.db
- Ian: __init__.py, db.py
- Vivian: login.html, home.html, storing data into users.db


app.py
login.html
(has login & register)
login unsuccessful
login/register successful
db.py
stores login data
users.db
home.html
create new story
search.html
search for preexisting story
edit.html
stories.html
edit existing story
db.py
stores edit data
edits.db
stories.html
returns to page editted.