bigpro’s Wiki Design Doc | TARGET SHIP DATE: {2025-11-7}

Legend
Bold=Completed *Italic=In Progress* Normal=Not started/explanation

(PRINT 3 COPIES -- **A PDF OF THE FINALIZED DESIGN DOC MUST BE PUT IN THE REPO UNDER DESIGN.PDF!!!!!!!!!!!!!!!!!!!!!!!!!)**

COMPONENTS

- Sessions
  - **login/logout**
    - **Locally stored user/password list**
    - DATABASE TABLE FOR USER INFO:

USERS		
TEXT	username	PK
TEXT	password	

    - Ability to register: Separate webpage
    - Add to USERS a username and password of their choice, so long as the username isn't taken
  - Use sessions to keep a user logged in and only log them out if they press a logout button
- Webpages
  - Profile (PRIVATE, POST)
    - List of websites the user has created /edited with links
      - Allow editing + creation of pages: have user type in title of webpage, content of webpage, and submit

- User created webpages (editable if logged in, if a non-logged in user presses edit, we redirect them to the sign in page)
  - Entire webpage templates (in HTML) using placeholders/variable
    - Use render_template and variables received through Flask code to input values into a templated webpage (so same format)
  - DATABASE TABLE FOR WEBPAGES

WEBPAGEINFO		
TEXT	title	
TEXT	heading	
TEXT	content	

  - DATABASE TABLE FOR PROFILE AND DISPLAYED WEBPAGES

PROFILE		
TEXT	picture	
TEXT	username	PK

WEBPAGES	
TEXT	username
TEXT	pageTitle
TEXT	pageLink

    - PROFILE AND WEBPAGES OPERATE SIMILARLY TO OUR PREVIOUS ASSIGNMENTS STUDENTS AND COURSES, WE CAN ACCESS THE WEBPAGES SOMEONE HAS CREATED BECAUSE THEIR USERNAME IS LINKED TO THE PAGE :)

    - Navbar
        - Log in / sign up / logout button
            - Log in/Sign up for if the user isn't logged in, allowing them to either create an account or log in
            - This turns into a logout button once logged in
        - Link to a logged in user's profile page
            - Users not logged in redirected
        - Categorize webpages and split them into dropdown tabs?
            - Either this or one of those alphabetical list of webpages that a lot of wikis have
        - We could just link to a different webpage because depending on number of tabs/sections, it could get hard to format
    - General wiki information (PUBLIC, GET)
        - Includes everything the user types for the webpage
            - Heading
            - Page content
            - Author of page (username)
    - Homepage
        - Description of what our wiki is/for, maybe a link to some popular pages, maybe instructions on how to edit webpages??
        - Link to a page of all webpages sorted by alphabetical order?
- Database (SQL)
    - *Search would be cool but is extra*
    - Stores all webpages and data
    - Refer to DATABASE TABLE FOR USER INFO:, DATABASE TABLE FOR WEBPAGES, and DATABASE TABLE FOR PROFILE AND DISPLAYED WEBPAGES for the mock table setup :)
- HTML templates
    - Login page
        - Must take user input
    - Logout page (?)
    - User created wiki page(s) (only need one template)
        - Must take user input
    - Homepage

- Profile page

COMPONENT MAP

The Wiki
└─ **Session** acts like a cookie;is the user signed in? To which account?
└─All of the following functionalities require the use of **SQLITE3**
│and its database tools in some capacity. Thus, they all require
│proper algorithms to pull the requested data
├─ **Users** and user data will be stored in the **user** database, and
│authentication may be handled/checked by its own function called
│upon accessing any page.
│└─**Registration** (possibly account deletion too)also falls under
│a similar umbrella. A form will take two inputs, a username and
│password, and then check whether the username is taken. If not,
│it will add it to the table of registered users and the user
│will be signed in, otherwise it will prompt the user to pick
│another username. This page will not be accessible if the user
│is signed in.
├─ **Navbar**
│ └─ acts as your standard finder tool, formatted similarly
│to a directory, with multiple (likely fixed) webpages [home,
│profile, one or two popular wiki articles] and a search tool.
│Said tool will be its own function that combs through the
│database of **webpages**
├─**The Webpages**
│ └─nothing too complicated here, a standard display function
│can work here, that fishes the standard wiki post HMTL template
│and fills in the appropriate data [heading, body, title,
│optional images etc]. Separate booleans or functions may be
│needed for the profile and home webpages which may have
│different formats.
├**Editing and Creating webpages**
│ └─If a user is logged in, they may create a webpage through
│filling out a series of forms that designates its title and
│information (following a select few inputs like header, body
│etc), where duplicate headers would overwrite the old file,
│allowing editing. Following this, it saves the information to
│the **Webpages** sqlite3 database and refreshes the page. The title

|may also be saved to an attribute on the **user** database,
|**webpages_created** so we can display all created/edited webpages
|on their profile

SITE MAP

```
Home (PUBLIC)
├─ Login (PUBLIC)
|   └─ redirects to Profile
├─ Profile (PRIVATE)
|   └─ includes links to all webpages this user created / edited
├─ Webpage List (PUBLIC)
|   └─ all webpage links sorted by alpha order
|   └─ Create/Edit Webpage (PRIVATE)
|         └─ user inputs webpage title and (new) content for webpage
|         └─ redirects to Login if not logged in
└─ Logout (PRIVATE)
     └─ redirects to Login or Home
```

TASKS AND ROLE ASSIGNMENT

Task	Role	Agent
Database creation (SQL)	Use SQLite3 to initialize tables and values	Carrie
Flask pathways (Python)	Creating routes for all pages, functions for login/logout/profile, and restrictions	Amy
HTML template	Build HTML templates for all webpages (home, webpages, profile, login, etc.)	Carrie
Wiki functions	Implement function for creating, editing, and getting inputs from user; restrict users that are not logged in	Ethan
Navbar	Create and link navbar to webpage links	Owen
Profile page	Display necessary information on profile	Owen
CSS	*Creating a style template for webpages*	*Amy/Carrie*

README.md (github link)

First App by bigpro

Roster: Carrie Ko (Back-end/Front-end), Ethan Saldanha (Back-end), Owen Zeng (Project Manager), Amy Shrestha (Front-end)

Description: This website is a wiki, meaning it's a community run encyclopedia for the greater good. Anyone can access its various webpages containing community posts (like a QAF) by searching for a desired topic through a navbar, however users must be registered to create or edit any webpages. Logged in users will have access to a profile page, containing some personal information along with access to any webpages they've created or edited.

Install guide

1) Clone the repo into a local directory:
    a) git clone https://github.com/Owen-Zeng/bigpro_amy_carrie_ethan_owen.git
2) Enter the app directory:
    a) cd bigpro_amy_carrie_ethan_owen/
3) Install necessary modules:
    a) pip install -r requirements.txt

Launch codes

1) Run the app through Flask:
    a) flask run
2) Open the website at website:
    a) http://127.0.0.1:5000

requirements.txt

```
# Carrie Ko, Ethan Saldanha, Owen Zeng, Amy Shrestha

# bigpro

# SoftDev -- P00

# 2025-10-22

# time spent: 0.1

Flask
```

CONTACT INFO

Carrie Ko
carriek3@nycstudents.net
Github: potcats
Insta: elluwin
Discord: elluwin

Ethan Saldarha
ethans201@nycstudents.net
Github: esalad60
Insta:
Discord:

Owen Zeng (Project Manager)
owen220@nycstudents.net
Github: Owen-Zeng
Insta: owheen.z
Discord: owenidur

Amy Shrestha
Amy Shrestha
Github: zwkir
Insta: xjin.ju
Discord: xjin.ju

Guys pls check the tree next time :(the flag needed to be in the REPO and not in this design doc….. 🙁
Repo:
https://github.com/Owen-Zeng/bigpro_owen_ethan_amy_carrie

TEAM FLAG