Jake Liu, Bogdan Sotnikov, Joyce Lin, Naomi Kurian
Lost and Found
P00: Half-Quick
11/9/25
TARGET SHIP DATE: 2025-11-10

SCENARIO Σ

Program Components
Landing page - search bar, logout button. navbar
Navbar - help user navigate stories
Search bar - search for stories
Login page - log in to access editing and profiles
Registration page - creates a new user
User base - store users
User - store our user's information (username, password, pfp, path to profile page) or basically each row in the user base
Story base - store stories
Story page - displays story information from specified story
Profile pages - display user information
Logout page - log out of session
Edit Page - allows user to edit story contents

METHODS:
- **fetch_data(table, rowid, data, criteria)**: Accesses corresponding table and pulls the requested data
    - Will comprise IF statements corresponding to each value and submit a query using those select statements → fetchall() will be ran afterwards
    - If criteria is 'none' then this section will be ignored
- **create_user(username, password)**: Adds an 'empty' user to the userbase with the provided username and passwords
    - Randomly selects a profile picture
- **update_story(s_rowid, editor_id, title, content)**: Either creates a new story with provided info or adds on to an existing story (appends new content onto old content)
- **update_pfp(pfp, u_rowid)**: Updates a user's profile picture based on input from profile page
- **update_password(pw, username)**: Updates a user's password based on input from login page (forgot password feature)

Component Relationships

Landing page
- Only visible for logged in users, if not authenticated (no rowid present in session), send straight to login page
  - Can be done by running authenticate()
- Contains *search bar*
  - Utilizes *Story base* (through a case-insensitive search) to pull up an article
- Contains a list of created stories that pulls from the *story base*
  - Displays nothing if empty
- *Navbar* will contain an 'publish new story' button (leads to *author page*), logout button and profile button

Login Page
- Asks for username/password and matches it to what is contained in the *user base*
  - Takes the username and compares it to existing *users*
    - If username does not exist **OR** if username exists but the password does not match the one attached to the corresponding *User*, output 'username or password incorrect'
    - If username exists AND password matches, sends the user back to the *landing page* and adds the username to the session
- Contains a button to create an account (sends to registration page)
  - Once account is created, adds the new *user* to the *user base* and signs them in (sending them to *landing page* and adding the corresponding **ROWID** to the session)
- A "forgot password" button
  - User is prompted for their username
  - Shows a math problem
  - If the user solves the problem, they can change their password

Registration Page
- Looks basically like the login page except there is an extra confirm password form response box.
  - If submit is pressed and it does not pass the comparison check, the page will display a "passwords do not match" message
  - If submit is pressed and the username is the same as another one (case sensitive), the page will display a "username taken" message
  - If submission passes all checks, user will be sent back to the login page and the new user will be added to the user base
- Will include a button (back button or return to login button) that sends the user back to the login page

Profile Page

- We could also cut this part out potentially by using a default format for the path and plugging the rowid in at the end (rowids are unique identifiers)
- Displays pfp, 'hi, [username]!', associated *stories* , a back button (leads to *landing page*), and a logout button (leads to *logout page*) —> both buttons located in a *navbar*
  - *Stories* will be pulled from the *Story base* by checking if *user* is present as an editor (stored in Story base as string of rowids) or author for any of the stories, and the link will be included (redirect() with the story's path)
  - Two separate lists for authored stories and edited stories
- A button or something that allows you to choose from a curated list of profile pictures (while the current one is randomly selected when the account is first created)

Story Page
- Only visible for logged in users
- Displays the title of the article at the very top, most recent entry below (unless already edited), and then the editors and authors at the very bottom (structure may change based on whatever's aesthetically pleasing)
- *Navbar* contains a *search bar*, a logout button (leading to *logout page*), a back button (leading to *landing page*), profile button, and an edit button (goes to *edit page*) that is only visible if the user hasn't edited yet.

Edit Page
- Through the identity of current story (using the rowid of the story's path to select from story base), send the most recent content of the *story* (last_entry) that is being edited to be displayed in edit page
  - Have the edit page display form response box below the last entry
- Page will contain a submit button
  - New section will be stored in *Story base* (last_entry)
  - Will send user to corresponding story with changes published and the user can now see the entirety of the story
    - Once you go back, the *user* currently in session will be saved to the editor string of the corresponding *story* (cannot edit again once added to this string)
- In the *navbar*, a back button will exist that'll send you back to the article you were originally on (changed are unsaved and no further actions will be taken)
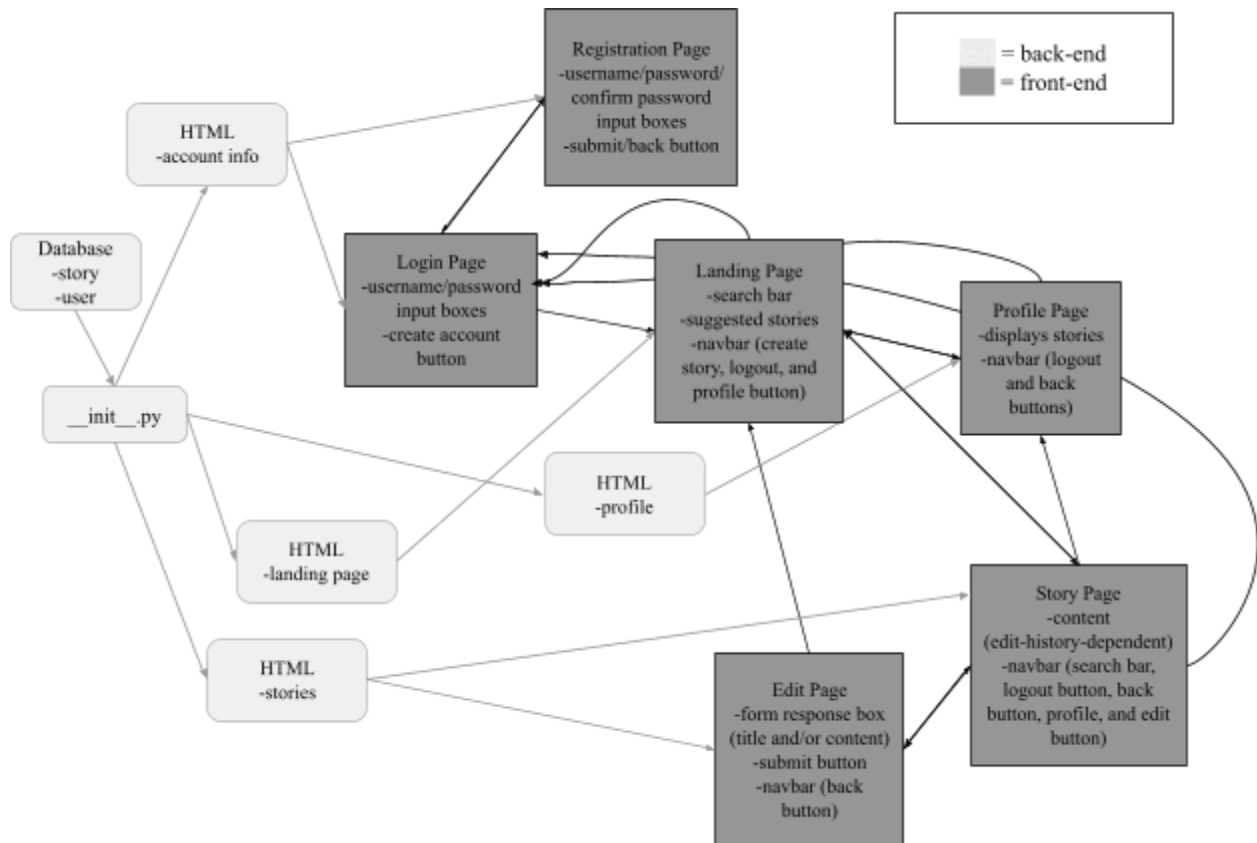
Author Page
- Works similarly to edit page except there are two empty form response boxes (one for title and one for content) → this should be made clear on the page
- Submission (using the submit button) will send you to the newly published story
  - The new story will then be added to the story base and path will be created with the first submission being saved as both content and last_entry
  - Editors will be empty while author will save the authoring user's rowid

- In the *navbar*, only a back button will exist that'll send you back to the landing page (changed are unsaved)

Removed Features
- Login page
- "Next page" button
- Letter-by-letter search
- Random story selection

Component Map

Database Organization
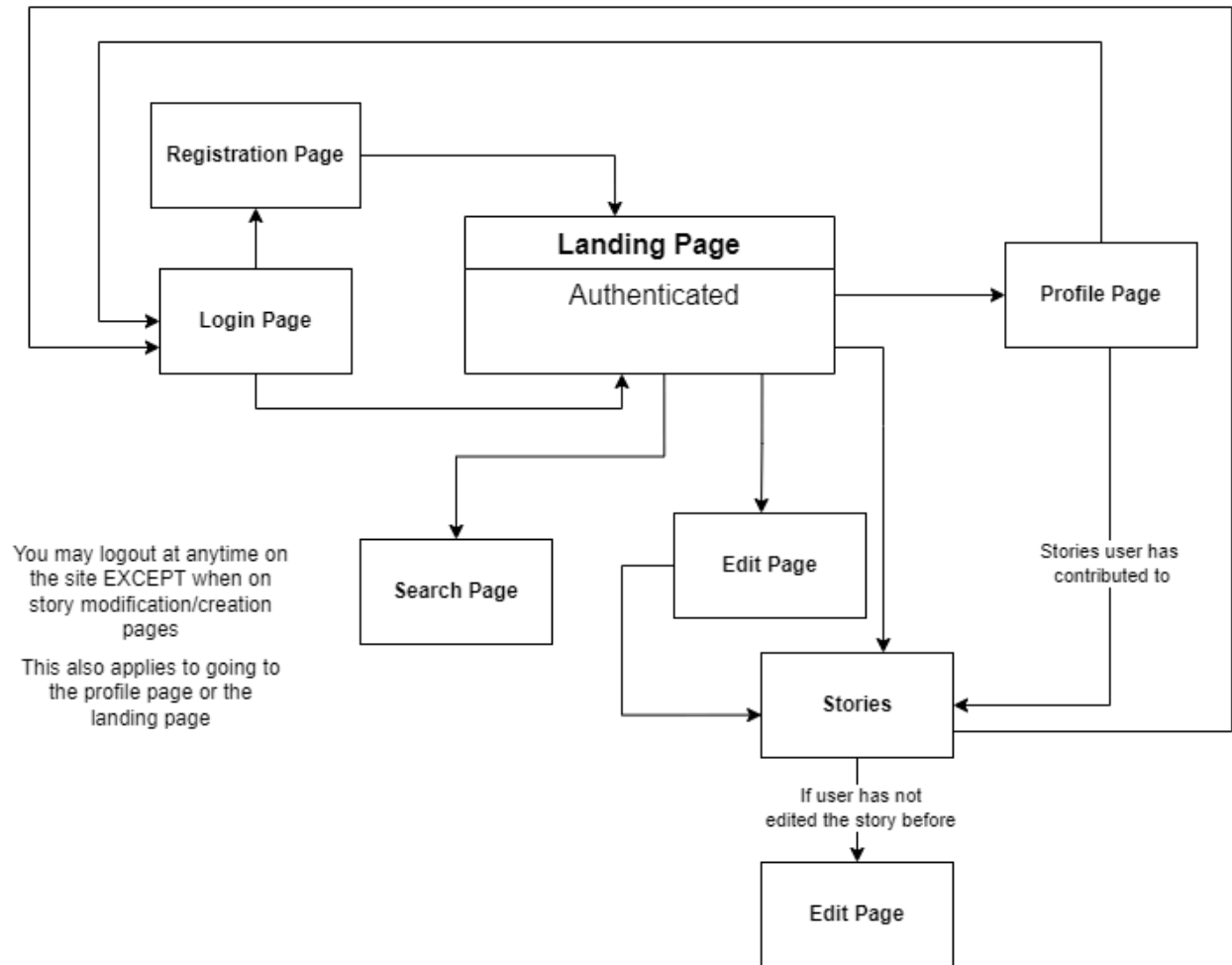sqlite tables will contain the following information:
***path will store "/[USER/STORY]/rowid"*
Default SQLite PK is used (rowid)

USER BASE		
TEXT	username	
TEXT	password	
TEXT	pfp	
TEXT	path	
TEXT	contributions	Contains story rowid
INTEGER	times_cont	

STORY BASE		
TEXT	path	
TEXT	title	
TEXT	content	
TEXT	last_entry	
TEXT	editors	Contains user rowid
INTEGER	author	Contains user rowid

Site Map

You may logout at anytime on the site EXCEPT when on story modification/creation pages

This also applies to going to the profile page or the landing page

Task Breakdown

Team member	Tasks	Expected Delivery
Jake Liu	☑ Story Page ☑ Edit Page	- 11/5 - 11/6
Bogdan Sotnikov	☑ Landing Page	- 11/8
Joyce Lin	☑ Profile Page ☑ Flask Routing (__init__.py)	- 10/30 - 10/29
Naomi Kurian	☑ Login Page ☑ Registration Page	- 10/30 - 10/30