Team Greedy Toads

Helena Williams, Anya Zorin, Arib Chowdhury, Benjamin Gallai

p0 Design Document

Components

Templates / Frontend

The HTML files that make up the front end and have the forms into which users input the data that will then be retrieved by the python program. The pages are specified in more detail in the site map.

Flask

Flask is the python website framework that takes input from the templates and puts it in the database, as well as retrieves data from the database. It also redirects the pages as well as keeps a session of the user that is logged in.

Database

The database stores the data for the website. It is explained in detail in Database Structure.

Component Interaction

Creating New Stories

Users can create new stories. To do so, they will be redirected to storypage.html with added set to false and created set to false. This will lead to the generation of the standard adding form, but with a title menu as well. They will fill out an HTML form with the title and the first entry to the story. The flask component will then add a table to the database and populate the first row with the entry information. Then, also through the flask component, the user will be redirected to the newly created story template page.

Adding to Stories

Adding to stories is similar to creating a new one, and directs the user to storypage.html with added set to false and created set to true. The user gets a preview of the story (the most recent addition) retrieved from a table in a database using python and then displayed on the web page, and then submits a form to add their part of the story to the table.

Viewing Your Stories

Once the user contributes to a story, they are allowed to see all contributions that have been/get made to it, giving them an understanding of how great (or not great) their writing was! They are sent to storypage.html with created = true and added = true, which means that all story entries are put on the screen for them to peruse.

Authentication

Authentication has four main parts: create an account, log in, maintaining a session, and log out. There are separate templates for logging in and creating an account, in which the user

submits a form that the framework uses to either create a new row in the user table and start a session, or just start a session for an existing user. The session is maintained by flask and is used to load the page with stories the user has contributed to, as well as when the user creates/ adds to stories. Logging out is a button that ends the session.

storypage.html

	Added = true	Added = false
Created = true	DO NOT generate submission form DO NOT generate title form DO generate all previous entries	DO generate submission form DO NOT generate title form DO NOT generate all previous entries (except last entry)
Created = false	This state should never be reached by the user.	DO generate submission form DO generate title form DO NOT generate all previous entries (they don't exist)

UPDATE: We have replaced the storypage system with a much simpler system of 3 pages: storyCreate, storyAdd, and storyView. This is really nice because it's more foolproof...and because I'm too lazy to code it the other way.

Database Structure

Story Tables (NAME: [story_title])

Each story will have its own table within the database, with its title as the name of the table. The id can be used to track the latest entry as well as the order of entries. The schema of the table for stories will be:

id	user_id	story_contrib
INTEGER	INTEGER	TEXT (200 max)

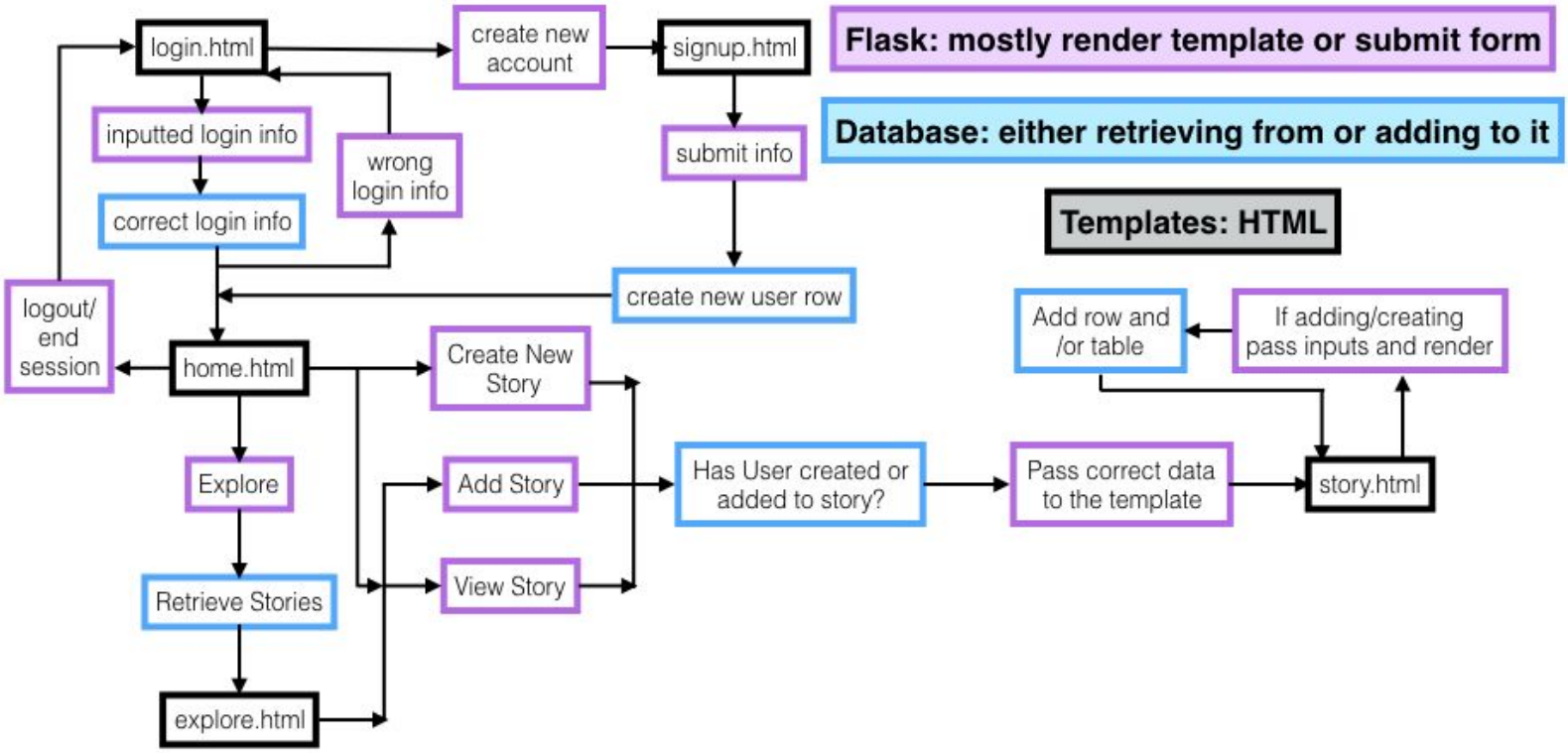
User Table (NAME: users)

The User table stores the user information of users, like username, password and the list of story_ids they have contributed to.

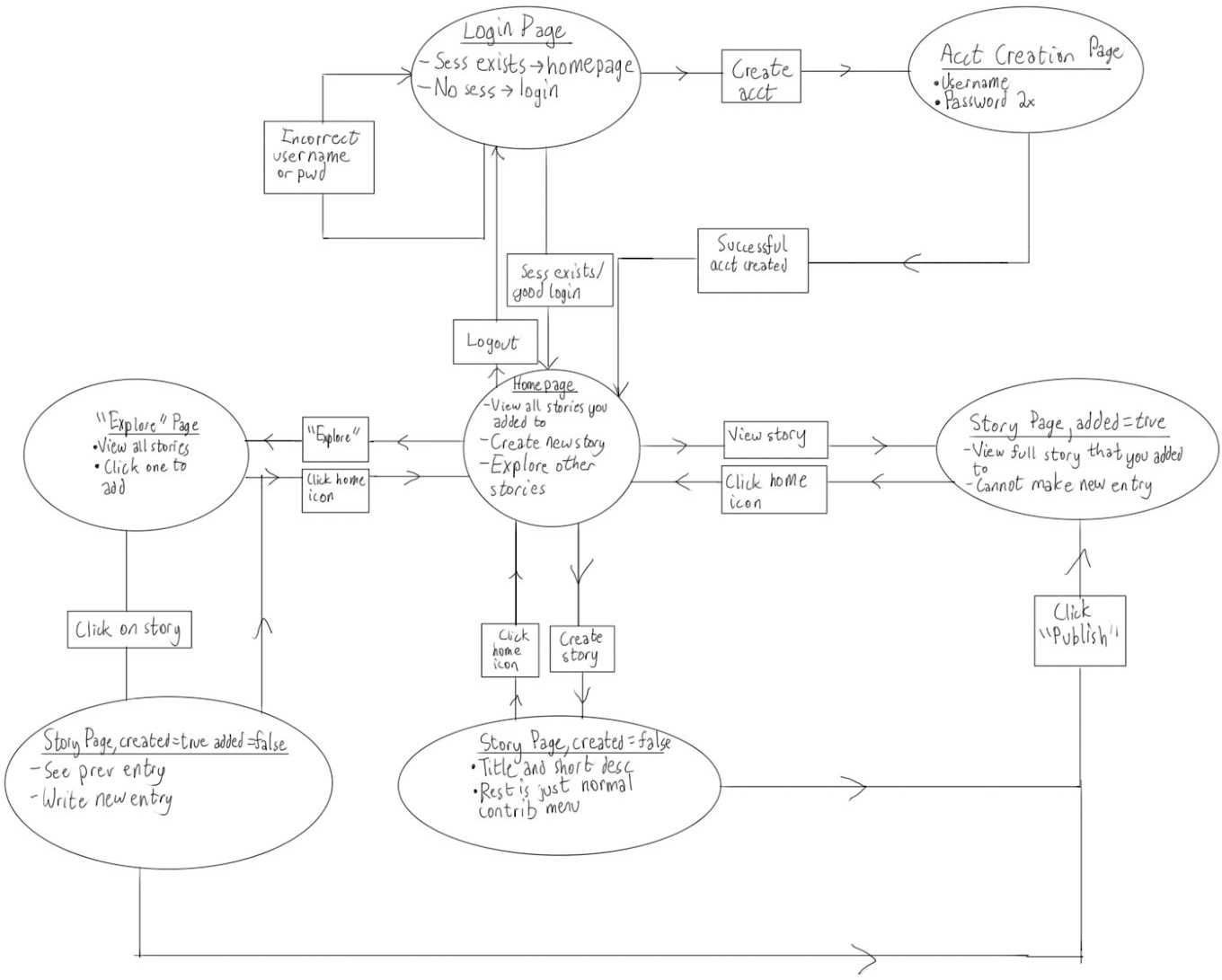
username	password	story_ids

TEXT	TEXT	TEXT (delimited by ,)

Component Map



Site Map


Login Page
- Sess exists → homepage
- No sess → login
Create acct
Acct Creation Page
• Username
• Password 2x
Incorrect username or pwd
Successful acct created
Sess exists/ good login
Logout
Home page
- View all stories you added to
- Create new story
- Explore other stories
"Explore" Page
• View all stories
• Click one to add
"Explore"
Click home icon
View story
Click home icon
Story Page, added = true
- View full story that you added to
- Cannot make new entry
Click on story
Click home icon
Create story
Click "Publish"
Story Page, created = true added = false
- See prev entry
- Write new entry
Story Page, created = false
• Title and short desc
• Rest is just normal contrib menu

Tasks

Timeline ish

0. 01.06.21: Successfully host a blank template HTML page
1. Finished: Steal login code from previous HW, give it a new paint job, successfully integrate it with blank template homepage
2. Finished: Add create account section, integrate it with login code, confirm that the full loop works
3. Finished: Make the homepage nice and spicy, create explore and create buttons
4. Finished: Create button should take user to storypage.html with the created = false query string, allow them to write up story component and add it
5. Finished: Any story that the user creates with the create button will now be visible on their homepage, they can click on it and will be taken to storypage.html with created=true
6. Finished: Add explore button, takes them to explore.html page where all stories written by all users are displayed but cannot be clicked on
7. Finished: Add “added” query string to storypage.html, if user added to the story they can view it when they click on it in Explore, but if they haven’t added to it they will be taken to an add entry feature
8. Finished: Ensure that any story the user adds to will be visible from their homepage by adding in database features
9. 01.06.21: put it all together
10. 01.07.21: Add monstrous CSS

MINIMUM VIABLE PRODUCT BY 1/6/20
We’re adding this cuz we forgot to originally, but we expect our project to be up and running by the 6th (was previously 5th, but PM got sick :(so it was pushed back). All members have pushed their code, and we just have to connect the wires to make it all function!

Frontend- Helena

- login.html
- signup.html
- home.html
- explore.html
- story.html

Create/Add Story Function- Arib

- Create a create_story function that accepts input from an html form to @app.route(“/create”) consisting of a string (story_id) and another string (story_contrib) and will create a table with the name story_id in the database, passing the story_id and story_contrib to story_add_helper afterwards.
- Create a story_add function that accepts input from an html form to @app.route(“/add”) in the form of story_contrib, and story_id and passes it to story_add_helper

- Create a story_add_helper function that will take the input story_id and story_contrib, grab user_id from a different function. and insert story_contrib and user_id into a table titled story_id in the database. Will also update the corresponding row in the user table to include the story_id for the corresponding user_id. Finally, will call a render_story function with the input added=True

Rendering story.html- Arib

- Create a render_story function that will accept boolean inputs of create and add to @app.route("/render").
    - If create=False, just render_template create.html that'll output to the create_story function.
    - If create=True and add=False, also search for a string input of story_title from form, grab story_contrib from the last row of the corresponding table and send it as input to render_template add.html which will output to the story_add function.
    - If create=True and add=True, also search for a string input of story_title from form, grab story_contrib from all rows of the corresponding table and send it as a list as input to render_template story.html

Rendering home.html - Benjamin

- Simply retrieve the stories that the user has contributed to and display them
- Each user has a list of story ids that they have contributed, and each contribution has a user id assigned to it

Explore page rendering - Benjamin

- View titles of all stories

Authentication (login/create user/logout functions) - Anya

- Re-use code to create a login system, as well as a create user which adds a row to the db
- Start/end sessions
- Also, create functions that others can use to retrieve the active user id or username
- Create functions that append the story id/ name to "story_contrib"