

Blue Haired Gals with Pronouns

:: Jasmine Yuen (PM), Ziyang Jian, Talia Hsia

SoftDev

P00: Collaborative Storytelling Website Design Document

Target ship date: {2022-11-27}

PROGRAM COMPONENTS (WITH SPECIFICATIONS):

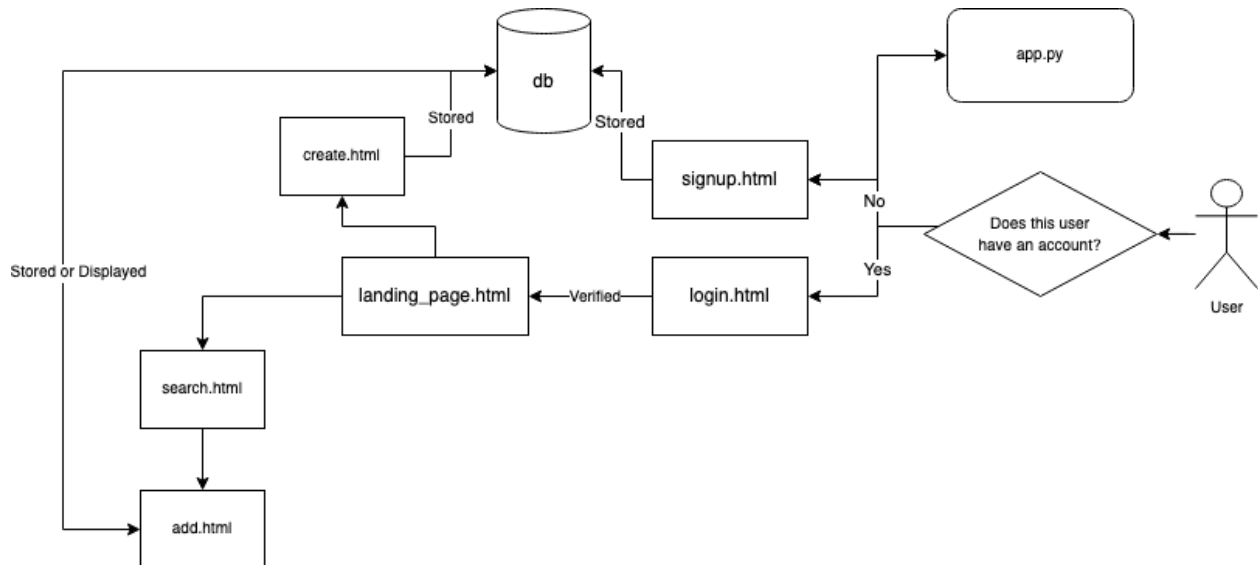
- flask app: `app.py`
 - allows for user input to be served onto the website and incorporates functionality onto HTML pages
- html templates: templates
 - login/signup:
 - `signup.html` - a form for the user to create an account providing the username and password, which is stored to SQLite database (DB)
 - `login.html` - a form for the user to input a username and password combination. through DB and flask, the login information will or will not be verified.
 - if it doesn't, the page does not proceed the user will be prompted to check their account info via a automated message
 - if it does, continue to landing page for that account
 - landing page
 - `landing_page.html` - display the ID, thumbnails, titles, genres, and synopses of each story the user has already added to
 - the ID will constitute 5 randomized numbers
 - these stories will exist within the DB, and they will be listed by ID under that person's account
 - navbar that has hyperlinked options:
 - search - `search.html`
 - requests query submission: you can search by genre, title, or ID
 - will display a list of related stories with image, synopsis, and genre tags
 - add to existing story - `add.html`
 - user can access a story that is stored in the DB

- users will be able to glean the last submission of the story
- includes a form where user will be able to add a 200 characters story

◦ create new story - *new.html*

- opens up query submission with empty input forms for title, genre(s), and synopsis
- required .jpg, .jpeg, or .png image upload for thumbnail
- randomly generates a five digit ID for that story (if it doesn't already exist, then proceed)

COMPONENT MAP (VISUAL) :

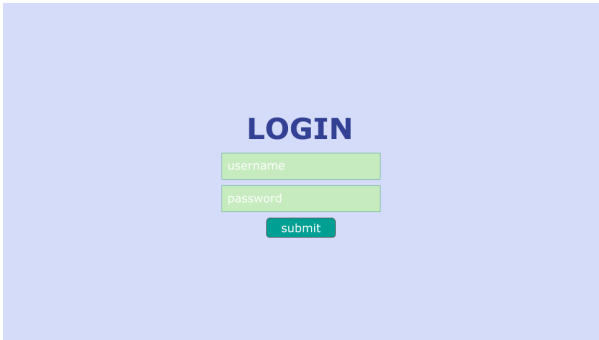
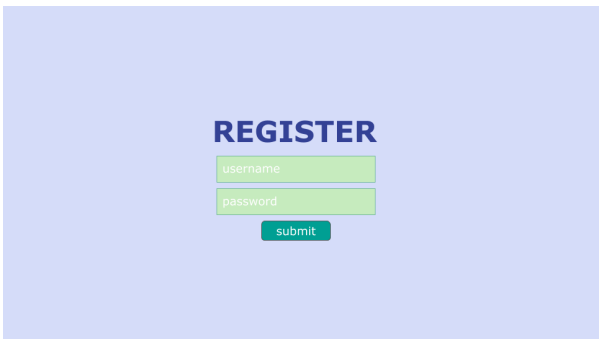


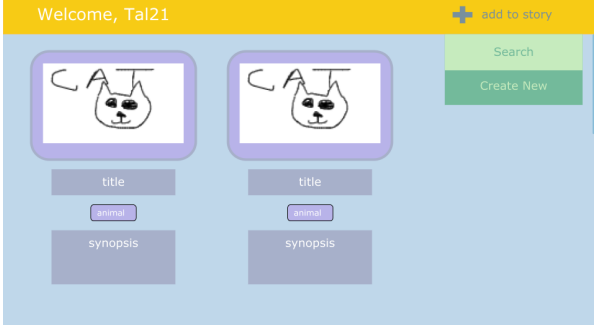
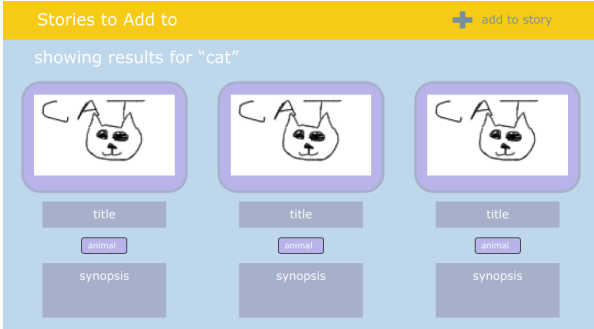
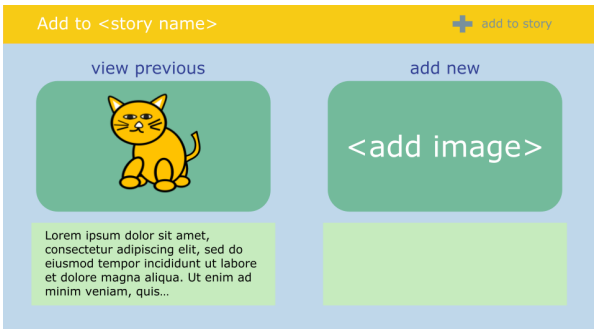
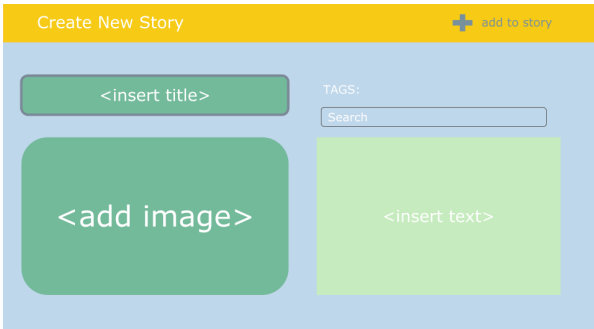
DATABASE ORGANIZATION:

- SQLite3 Database Tables
 - accounts (username STRING, password STRING,)
 - Table with all accounts and corresponding username and password
 - Username and password asked upon login
 - Usernames must be unique
 - Creation of a new account checks that the username does not already exist in DB
 - 1 table created under each account with the stories they've added to on creation of a new account.

- [username] (addedTo INTEGER)
 - Ie. urmomStories (addedTo INTEGER)
 - addedTo contains the id of the story
-
- stories (id INTEGER, title STRING, genre STRING, synop STRING, image STRING)
 - [storyID] (image STRING, text STRING)
 - everytime a story is started/created by a user, the story gets its own table
 - each row of the table will be an update to the story
 - id is a randomly generated integer (5-6 digits)
 - Creation of a new story will generate and id and add it as a new row with form input
 - Checks that id doesn't already exist in DB before adding

SITE MAP (VISUAL) :

	http://127.0.0.1:5000/login
	http://127.0.0.1:5000/signup

	<p>http://127.0.0.1:5000/landing_page</p>
	<p>http://127.0.0.1:5000/search?query=<search></p>
	<p>http://127.0.0.1:5000/add</p>
	<p>http://127.0.0.1:5000/new</p>

TASK BREAKDOWN:

Talia:

- login/signup page

Jasmine:

- populate DB with fake data and queries

Ziying:

- create create.html

ALL:

- create flask app in app.py, define routes, and connect routes to templates