

Gastric Bypass Train - Sam Cowan, Anna Fang, Sadi Nirloy

Poo: Move Slowly and Fix Things

SoftDev

11-2-2022

Time Spent: 2 hrs

Target ship date: 2022-11-8

Design Plan

TABLEHANDLER.PY

Import sqlite
Fon for handling registration: Add the username & password to the 1
Fon for handling login: Check the login table for the password of the given username
Fon for reading the entirety of the 2 as a list of list of string (split) for function
Function for taking an ID Num and returning the last only of a story just title
CnjectFunction for taking an ID Num and returning an entire story; uses title
Over Function for checking if a user has written in a story get (checking need in StoryJoin)
Story Start Foo: to create a new table for a story
Add2Story Foo: to add an entry to a story table using writepage form

DATABASE

TABLE 1: LoginInfo	
USERNAME	PSSWD

TABLE 2: HMPGE	
Title	ID Num

TABLE 3: ID Num: STORY		
<ID>: <title>		
Entry Num	Entry Text	Submitter

1/statik

1/css

1/template

SERVER.PY

Import Flask, Session
app = Flask(__name__)
app.config['SECRET_KEY'] = 'secretkey'
Fon for form of login & registration
• receive a post or info
• receives information call to Foo or Fon
to decide to move pages.
app.route ("/home")
Function to handle the creation of the home
• Creation of login button
• Creation of central home table via
• Table has buttons via URLS
These id nums lead to writing
or reading page based on the
Over function
• Form to write a new story gives
Start story
app.route ("/write")
writepage Foo & Fon • applies HTML
to start from Function
• Form for next entry
app.route ("/read")
readpage Foo & Fon • applies HTML
to CnjectFunction

Login page:

- HTML form(s)
 - The register form stores and adds new user login information (username and password as text) into a table
 - The login form references existing entries in the user-login table and allows the user to continue their ***session***

Homepage:

- HTML Table with all of the different pages
 - Each entry in the table consists of a row ID and story title
- Story creation form
 - Each entry consists of a pair of textboxes for the title and inputted text
 - Adds the story to the main homepage table (storing a unique row ID and the inputted title)

Loaded Pages:

- A way to return to the home page (button)
- Will either be a editing page or a viewing page based on the existence (or lack of existence) of user edits (reference Database Organization and Site map for more details)
 - The editing page will contain the story title, the most recent entry, and a textbox for the user to enter new text
 - The viewing page will display the full story (text)
 - Pulls from individual story tables

[Files]

Database:

- Table 1: stores user login information
 - Username (text) and password (text)
- Individual tables for each story
 - Stores all information
 - Reads username from Table1 to check if the user has already added to ***this*** specific table
 - Text (sentence) ID
 - Story text
 - Username

Python:

- Flask app with four routes:
 - / (login)
 - /home (homepage)
 - /write (additions to story)
 - /read (displays full story)

- Adds to database based on corresponding post requests from login, write
- Stores cookie to check whether one should be redirected to /read or /write when they navigate to a specific story page on homepage
- Uses render_templates to display pages properly

HTML:

- Templates for the display pages
 - Like a form

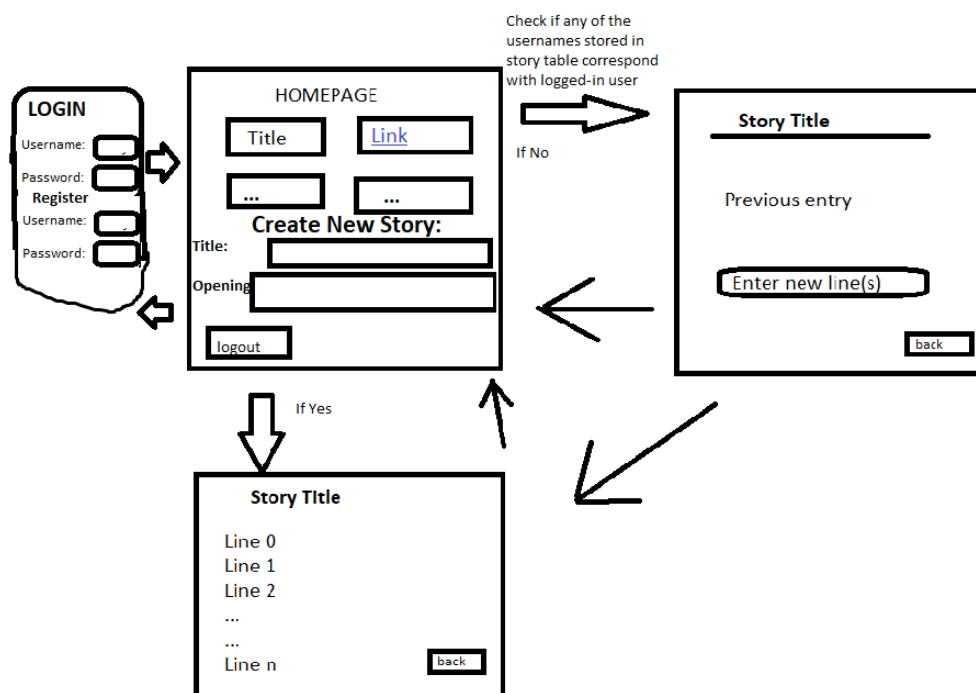
Sqlite database (which is dynamically added to by flask app):

- Contains all tables with user inputted information

Database Organization

One table for homepage which stores a row ID (integer), story title (text)

- A table to store user's login information: username (text) and password (text)
- Each story is housed in a table (of its own) which has three columns - row ID (integer), story text (text), and the user who wrote the story (text). The story can be seen by looking at the text row one by one



Site map

MVP:

- Login/register capability using sessions (with no practical use)
- Homepage which links to a single story that can be added to by user without limit

Expectation:

- Login/register capability using sessions
- Homepage with a table linking to every story
- Boxes on homepage to seed a new story
- When the user goes to a story page, their username is compared with the username column of that story's table. If there is a match, the user has already edited the story and is taken to a page showing the full table. If not, the user has not edited and is taken to an edit page where they can add a line to the story

Stretch Goal:

- N/A

Task Assignments:

MVP/Expectation:

- Flask app - **Sadi**
- HTML templates and CSS - **Anna**
- Sqlite-managing Python files - **Sam**

Stretch:

- [Functional] Logins - **Sadi**