PROJECT 0: OPTION 0

Pd 9 | Salading In Yo DMs: James Smith, Stanley Lin, Mansour Elsharawy, Marcus Ng

PROGRAM COMPONENTS

Root

Redirects to Home page

Home page

If logged in: Stories you have contributed to

If not logged in: Explanation of how it works

Login page

Allows you to enter username/password

Login Button

Register Button(creates a new account using text in boxes)

Create account page

Allows you to create a new account to participate in the story

Create page

Option to enter a title

Option to enter story body

Submit creation button

Available Stories

Displays all of the stories that the user has not contributed to.

Display Story page

(if the user has not contributed to the story) Allow users to only see the last entry and add content to the story

(If the user has contributed) Show the whole story, but disable adding content

TEMPLATES

HTML Template

Link to homepage

Log in/out link(depending on login/logout status)

Create new story link

Available stories link

DATABASE TABLES
Account Table

The account table contains 3 fields: username, password, and stories. Username and password are used to log the user in. Username is a primary key so that you can't have a duplicate account. The last field, stories, is a string representation of a list of story IDs that the user has contributed to. To edit the stories list, eval(idList) can be used to get a working list, then the new ID can be appended, then repr(idList) can be used to turn it back into a list to update that field.

Stories Table

The stories table contains 4 fields: ID, title, recent, and body. The first field, ID, is the unique(primary key) integer id assigned to the story to easily identify it. The second field is the title of the story which should also be unique(primary key). The third field, recent, is the most recent addition to the chain story so that we know what to display to the user who wants to add to it. The fourth field, body, is the full story for those who have already contributed.

DATABASE SCHEMA

The two tables are related to one another through the story ID. Account tables stores a list of story ID's that a user has contributed to. This ID can be used to find the story in the Stories Table.

Examples:
Account Table:

username (Primary Key)	password*	stories
Mr Salad	caeser	"[1,0,16,189,2009]"
Smitty	spegooter	"[20, 10, 23]"
soupdood237	troll4lyfe	"[16, 19, 21, 25, 260]"

*Passwords will be encrypted but shown here for clarity

Stories Table:

id	title (Primary Key)	recent	body
0	MrSalad	… who loved veggies.	Samuel Salad was an ordinary man…
1	Goldilocks and the Three Salads	… there were three salads.	Once upon a time…
2	The Ballad of Salads	…Scrumptious and delicious!	Praise the salad!…

FILES

Frontend.py -The flask app frontend that uses the functions in dbtool to pull/add information to/from the database at request from the user.
- **add_session(user)** - creates a session cookie for the current user
- **logout()** - Deletes a session key of a user, if one is logged in
- **home()** - provides functionality for the /home route
- **login()** - provides functionality for the /login route
- **create()** - provides functionality for the /create route
- **available()** - provides functionality for the /available route
- **page()** - provides functionality for the /page route

Authtool.py -The backend tool used to work with the accounts table
- **login(username, password)** - Logs user in if username and password are correct
- **encrypt_password(password)** - Returns SHA-256 version of password
- **create_account(username, password)** - Creates a new entry in .accounts
- **does_username_exist(username)** - Returns true if username exists, false otherwise
- **add_contribution(username, storyid)** - Appends storyid to current list of story IDs using eval(idList) and repr(idList)
- **did_contribute(username, storyid)** - Returns true if user has contributed to the story, false otherwise
- **available_list(username)** - Returns list of available story ids to be edited by a user
- **contributed_list(username)** - Returns list of story ids a user has contributed to

Dbtool.py -
- **create_story(title,body,username)** - Creates a new entry in .stories with the proper page id
- **update_story(storyid, addition, username)** - Updates the story with the specified id to expand its body and change its "recent" field, and calls add_contribution(username, storyid)
- **story_recent(storyid)** - Returns latest contribution to story with that id
- **story_body(storyid)** - Returns body of story with that id
- **story_title(storyid)** - Returns title of story with that id

ROUTES

SITE MAP
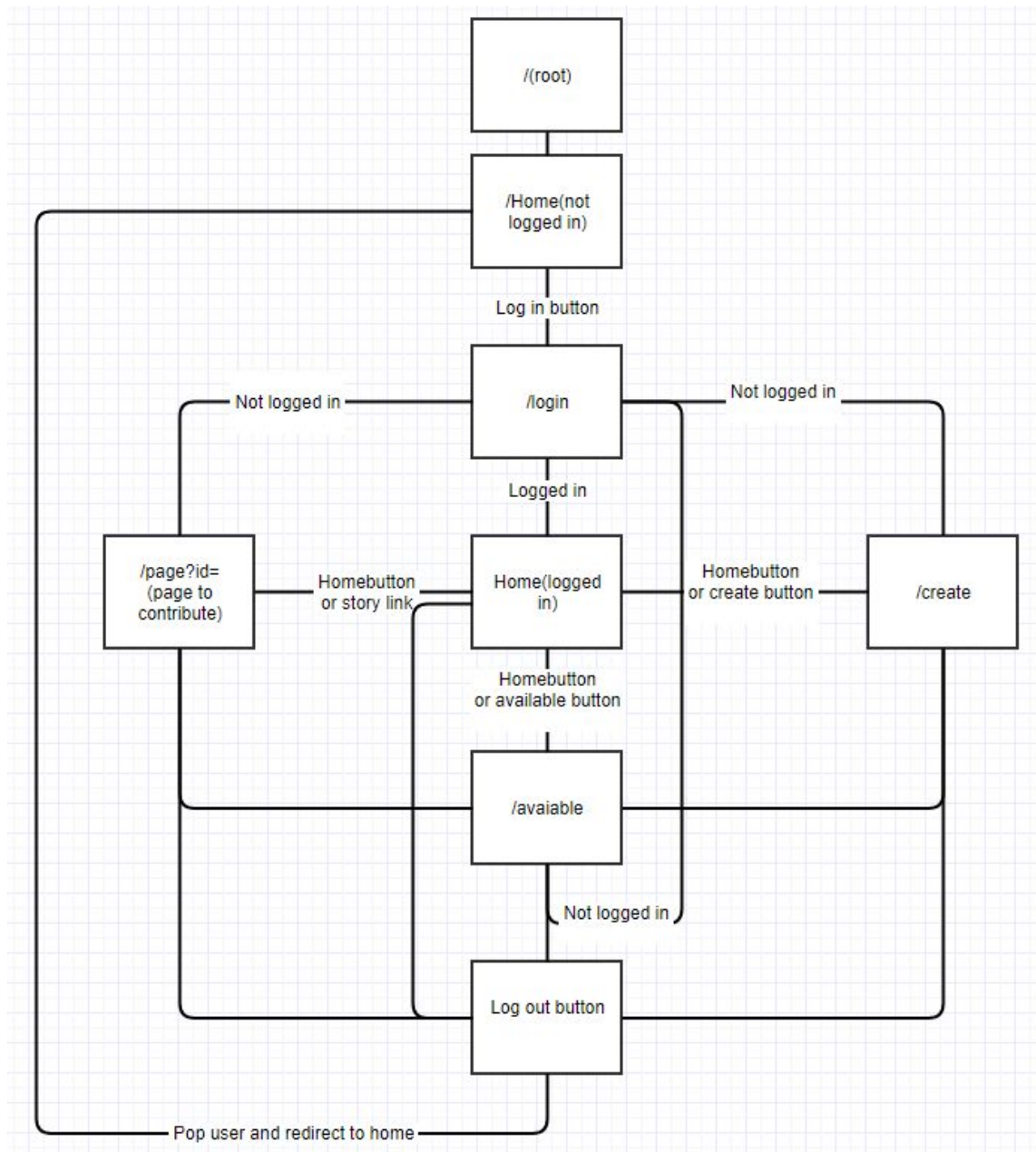
TASK ASSIGNMENT

Project Manager: James Smith

Backend tool(dbtool.py): Mansour Elsharawy(story functions), Marcus Ng(account functions)

Frontend(frontend.py): Stanley Lin

HTML: James Smith

General Debugging: James Smith

Generate Stories: James Smith