OVERVIEW

Our project is to design a weblog hosting site, also known as a “blog”. Our blog, called “TOPZ BLOGS” will be able to store user accounts and allow them to create blog pages. The main homepage will have login and register buttons and will display featured blogs. Each featured blog will have its most recent entry shown, but users can click into the blog to show every entry in that blog. Logged in users will be able to add, remove, or edit entries and blogs. Users not logged in will need to log in first in order to view any page on the site. Below, we describe the components, site, and database structures in detail.

ROLES:

HTML + related python/flask

Manfred: Project Manager, HTML designer

- HTML for web pages

Connor Oh: HTML related methods

- Utilize Jinja/Flask
- Login methods
- Make methods to display/format user page, blogs, entries

SQLITE + related python methods

Winston:

- Adding/Remove/Editing blog entries
- Sends entry to database, request entries

Jackson:

- Formatting Databases
- Organizing data

COMPONENTS

Html/Flask:

Flask will host the site while Html and Jinja will receive all the user inputs and pass them along to be used by the python methods. Some example methods related to the webpages are:

- @app.route("/")
- @app.route("/login")
- @app.route("/register")
- @app.route("/search")
- @app.route("/home")
- @app.route("/user/<user>")
- @app.route("/blog/<blog>")
- @app.route("/entry/<entry>")

Python:

This will be the intermediary between the front end HTML and the back end SQLite databases. There will be methods to handle transferring input from users, as well as returning necessary stored information in the databases. Some of the methods include:

- login(username, password): logs user into their account
- register(username, password): adds user data into database
- checkLogin(username, password): returns boolean
- addBlog(userID, blogTitle): adds new blog
- addEntry(userID, blogID, entryTitle, entryContents): adds new entry
- editEntry(userID, blogID, entryID): edits entry
- displayBlogs(userID): displays all blogs for a user
- displayEntries(userID, blogID): displays all entries within a blog
- displayRecentEntries( ): displays recent entries

SQLite Databases (explained in database layout): One file will include

- Users DB: stores the user ids, usernames, and passwords.
- Blogs DB: stores the user ids, blog ids, and blog names.
- Entries DB: stores the user ids, blog ids, entry ids, and the actual entry in text.

SITE

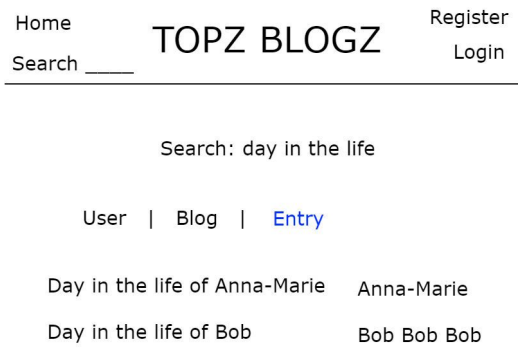
Each page of the site will contain a header with multiple functions. Users will be able to search for other users, blogs, and entries. Users can return to the home page. There will also be a settings and log out button.

On every page of the site, it will check the user's cookies for their account session. If the user is logged in, they are able to continue to the page. If they are not logged in, they will be redirected to the login page.

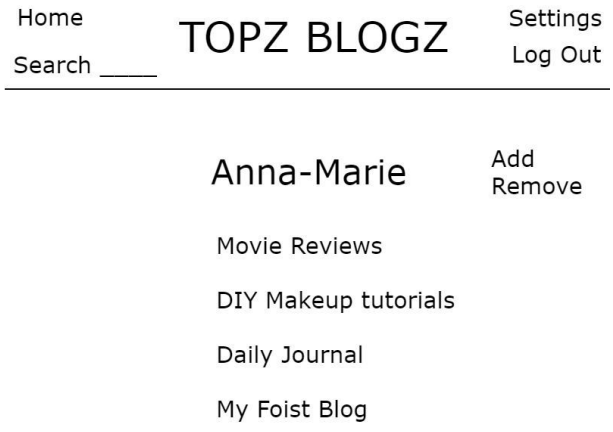
The home page ("website.com/home ") will have recent most entries submitted into the database. The user will be able to click on recent entries and view other users' blogs and entries. They can also click onto their own profile to view their blogs and entries.

The login page ("website.com/login") will have two fields for username and password. After verifying the user's credentials, it will log them into their account and return them to the page they were previously on. There will also be a button to redirect the user to the register page. The register page ("website.com/register") will create a new account for a user and return them to the page they were previously on. An error message will display if the username already exists. The password text will be hidden, and it will have to be entered twice for security.

The search bar will lead to a new page (“website.com/search?keyword”), where it will display items related to the searched keyword. The user can switch between what the keyword searches for. They are able to look at users, blogs, and entries related to that keyword.



Each user will have their own page to display their blogs (“website.com/user/<user>”). The blogs will be listed in chronological order, with the most recently created blog on the top. The creator may add or remove blogs. Users are able to click into each blog to view the related entries. They are also able to change their username or password, provided that they are able to provide their old username and password.

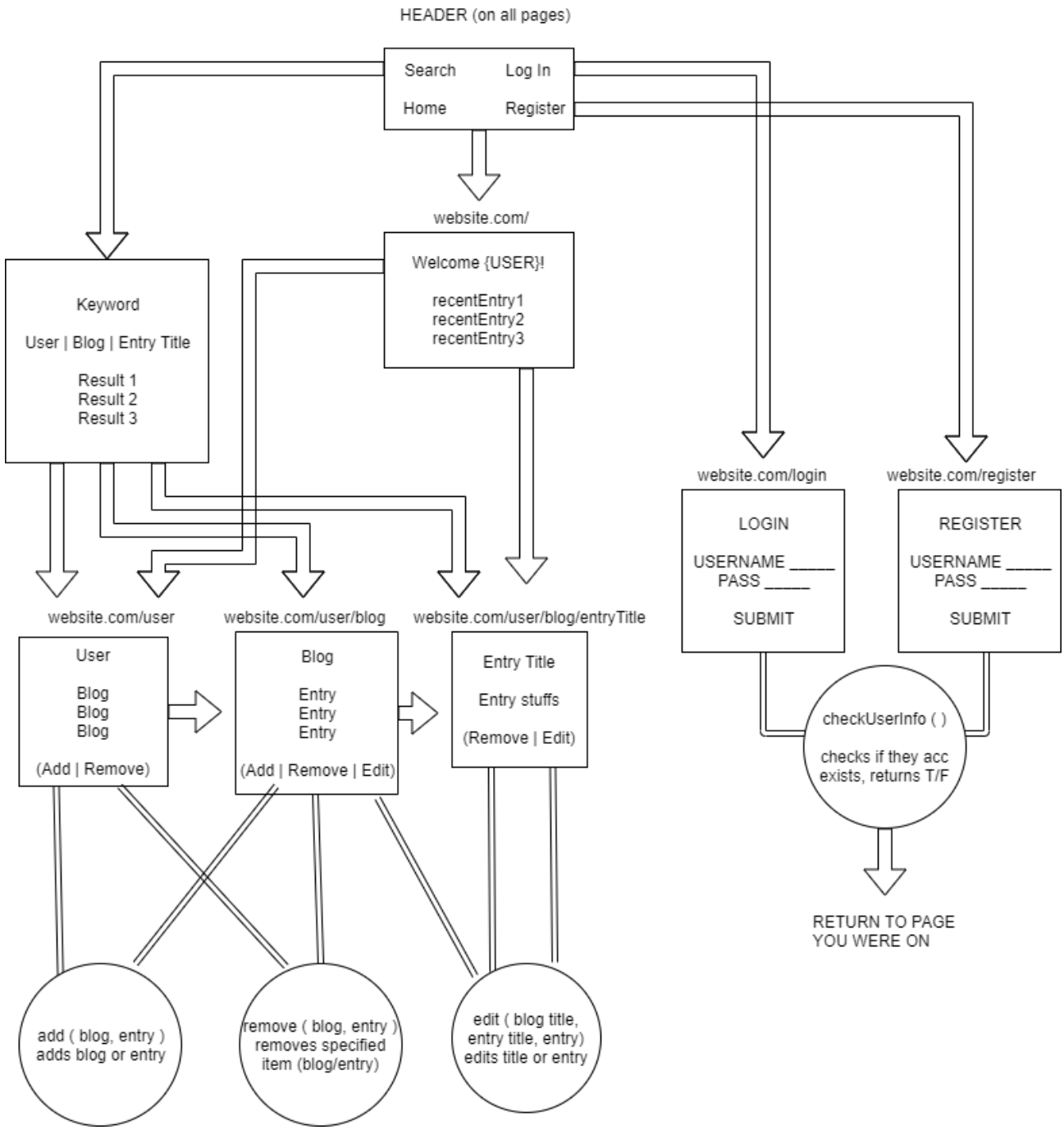


Each blog will also have its own page (“website.com/blog/<blog>”). The title of the blog will be on the top of the page with the authors name. The blog page will display all of the entries in chronological order, with the most recently created entry on the top. The creator may add new entries, remove entries, or edit the name of the blog. Users are able to click into each entry to view the content of the entry.

Each entry will have its own page (“website.com/entry/<entry>”). The title of the entry will be at the top, along with the associated blog and user. The page will display the content of the entry. The creator will be able to delete or edit the entry.

Overall Site map:

The square boxes represent web pages that the user will see, while the circles are some of the basic methods that will run in the background.



DATABASE LAYOUT

DATABASES:

Users:

Each user will be assigned a unique id when they register and will have their username and password stored inside the database.

ID: (PRIMARY KEY)	Username: (TEXT)	Password (TEXT)
1	“jackson”	“blahblah”
2	“winston”	“flqflejkf”
3	“myUser”	“myPass”
4	“hello”	“world”
5	“coolguy99”	“blahblah”
...	...	...

Blogs:

Each blog will also be assigned an ID. The blog ID will keep track of how many blogs a user has created. It will only be related to that specific user. All blog IDs will start at 1 and increase as the user creates more blogs.

USER ID: (INT)	BLOG ID: (INT)	BLOG NAME: (TEXT)
1	1	“Jackson’s life”
1	2	“My name is Jackson”
2	1	“Trending Fashion”
1	3	“My favorite songs”
...	...	...

Entries:

Similar to the blog ID, each new entry will be assigned an entry ID. The entry ID will keep track of how many entries the user has created in that specific blog. It will start at 1 and increase as the user adds more entries to the blog.

USER ID: (INT PRIMARY KEY	BLOG ID: (INT)	ENTRY # (INT)	ENTRY (TEXT)
1	1	1	“Day 1 - I sad”
1	1	2	“Day 2 - happy time!”
2	1	1	“black boots…”
1	1	3	“Day 3 - angry hulk >:(”
1	2	1	“I eat eggs every day”
2	1	2	“Green hats”