Scam – Sascha Gordon-Zolov, Margie Cao, Sasha (Alex) Murokh, Caden Khuu
Software Development
P00: Move Slowly and Fix Things
Project: Blog Hosting
2024-10-28
Time Spent: 2 hours
Target Ship Date: 2024-11-05

Components:
templates:

- **login.html** (route("/"))
  - Allows users to log in
  - Redirects to creating account page
- **create.html** (route("/create"))
  - Allows users to create a new account, username and password will be stored
  - Redirects back to login page after creating a new account; user will need to actually log in using the new credentials
- **blog.html** (route("/home"))
  - Split into two sections: "My blogs" and "Read blogs"
  - On the "My blogs" half, users can choose to create a new blog or edit an existing one
  - Redirects to (/myblog_) with _ being the index of the blog
  - Even if the blog title is changed the URL remains as the original index
  - On the "Read blogs" half, user can choose to read a blog, which redirects to (/user-myblog_)
- **edit.html** (/myblog_)
  - Users can choose to create entry or edit a pre-existing one, which redirects to (/myblog_/entry_) or (/myblog_entry_) [TBD], with the first _ being the index of the blog and the second _ being the index of the entry
  - Even if the entry title is changed the URL remains the index
  - User can rename their blog via a button that redirects to (/myblog_rename) or (/myblog/rename) [TBD]
  - User can navigate back to (/home) via back button
- **entry.html** (/myblog_entry_) or (/myblog_entry_) [TBD]
  - User can edit the entry title or text and submit
  - User can navigate back to (/myblog_) via back button
- **rename.html** (/myblog_rename) or (/myblog/rename) [TBD]
  - User can choose a new name for their blog and submit
- **reading.html** (/user-myblog_)
  - User can choose an entry to read (/user-myblog_/entry_) or (/user-myblog_entry_) [TBD]
  - User can navigate back to (/home) via back button

app.py

- Flask Server, designating routes and connections between pages + their functionalities

static:

- CSS file for standardized formatting of each page

scam_blog.db

- **users** table
    - Formatted data of usernames and passwords, each corresponding to an index by line
    - New items appended to end
    - Cannot be deleted
- **user_** table
    - Index in table name corresponding to index in users table
    - Formatted data of user's blogs (title), each corresponding to an index by line
    - New blogs appended to end
    - Cannot be deleted
- **user_blog_** table
    - Indexes in table name corresponding to index in users table and the subsequent index of blog in user_ table
    - Formatted data of user's blog's entries (title and text), each corresponding to an index by line
    - New entries appended to end
    - Cannot be deleted

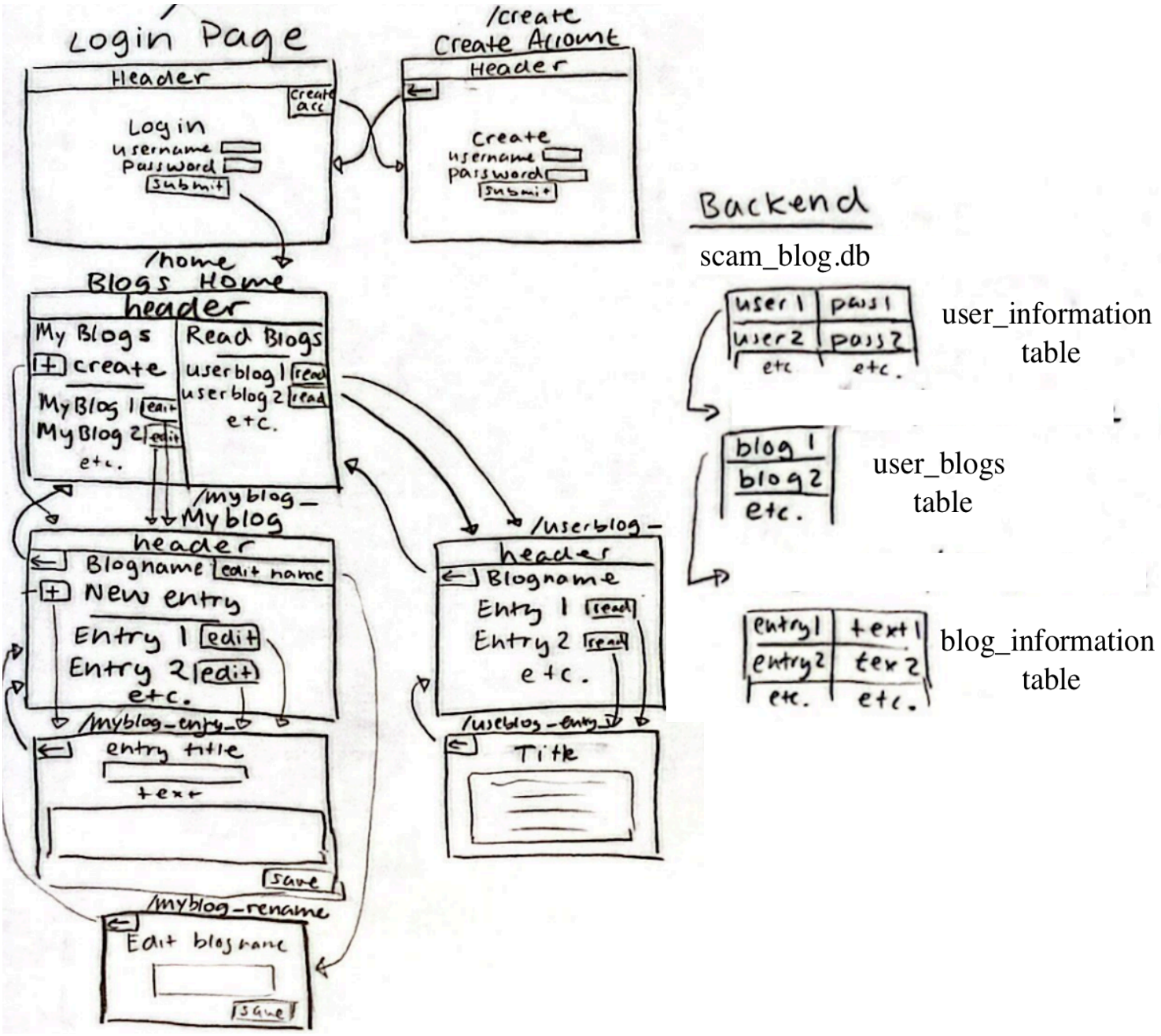
Roles:

Sascha - Login/Logout Functionality + Cookies + Creating new accounts
Margie - Creating new blogs/CSS
Alex - Sqlite, working with the database
Caden - Editing existing Blogs/Entries and CSS

Functionalities:

- Ability for users to log in and log out
- Independent sessions for different users
- Once logged in, users can create new blogs (this will be done through a form in a route)
- Users can edit their own existing entries
- A database with multiple tables:
    - One table to store the user ID and password
    - Each user is given a table containing their blogs; the file begins blank and grows by row whenever a user creates a new blog
    - Each blog is given a table containing a column of entries and column of texts, each entry corresponding to each block of text. This file will also begin blank and will grow as entries are added.

Component Map/Visual layout



Visual layout notes:

- Each page will have a standardized heading, containing our team name and a brief description of our blog hosting site
- Visual design (fonts, colours, etc through one css file) will be standardized across pages
- For navigation, HTML links or buttons will be used to redirect from page to page
- The back button is the built-in browser back button. There will be instructions in the heading for pages that can be navigated to/from the user's view.