

# Tiffany Yang, Claire Song, Jessica Yu, Jackie Zeng  
# Team Anything  
# SoftDev  
# P00  
# 2024-10-23  
# Time spent:  
# Target Ship Date: 2024-11-20

**Scenario Two:** Your team has been contracted to create a web log hosting site, with the following features:

- Users will have to register to use the site.
- A logged-in user will be able to
  - Create a new blog
  - Update their blog by adding a new entry
  - View and edit their own past entries
  - View the blogs of other users

Front end:

- ☐ HTML templates and CSS (if time): create layout of web pages; display tables and information from databases

Back end:

- ☐ Flask app: host web pages
- ☐ Database handling: create and link databases; sort data

### Components:

- User accounts
  - User sessions
  - User login/logout system
- Routes
  - Route to submit new blogs
  - Route to edit past blogs
  - Route to view past blogs
- SQLite db for storing blog info
  - Page\_id
  - Usernames/logins
  - Password hash

Database:

▼	Blog	
	User_ID	INTEGER
	Page_ID	INTEGER
	Page Title	TEXT
	Page Details	TEXT
▼	Edits	
	Page_ID	INTEGER
	Version_ID	INTEGER
	Version_Old	TEXT
	Version_New	TEXT
▼	Logins	
	Username	TEXT
	Password	TEXT
	User_ID	INTEGER

### Routes:

Home (/):

List of blogs

Login (/login):

Login with sessions and keys/cookies

Viewing blog page (/ {username} / {page\_id}):

Displays a single blog page

Create Page (/create):

Goes to a form to create a new blog page

Signing up: (/signup):

Allows you to make an account and saves username & password information

Edit: (/ {page\_id} /edit):

Allows you to edit an already created blog

Viewing past entries: (/viewpast):

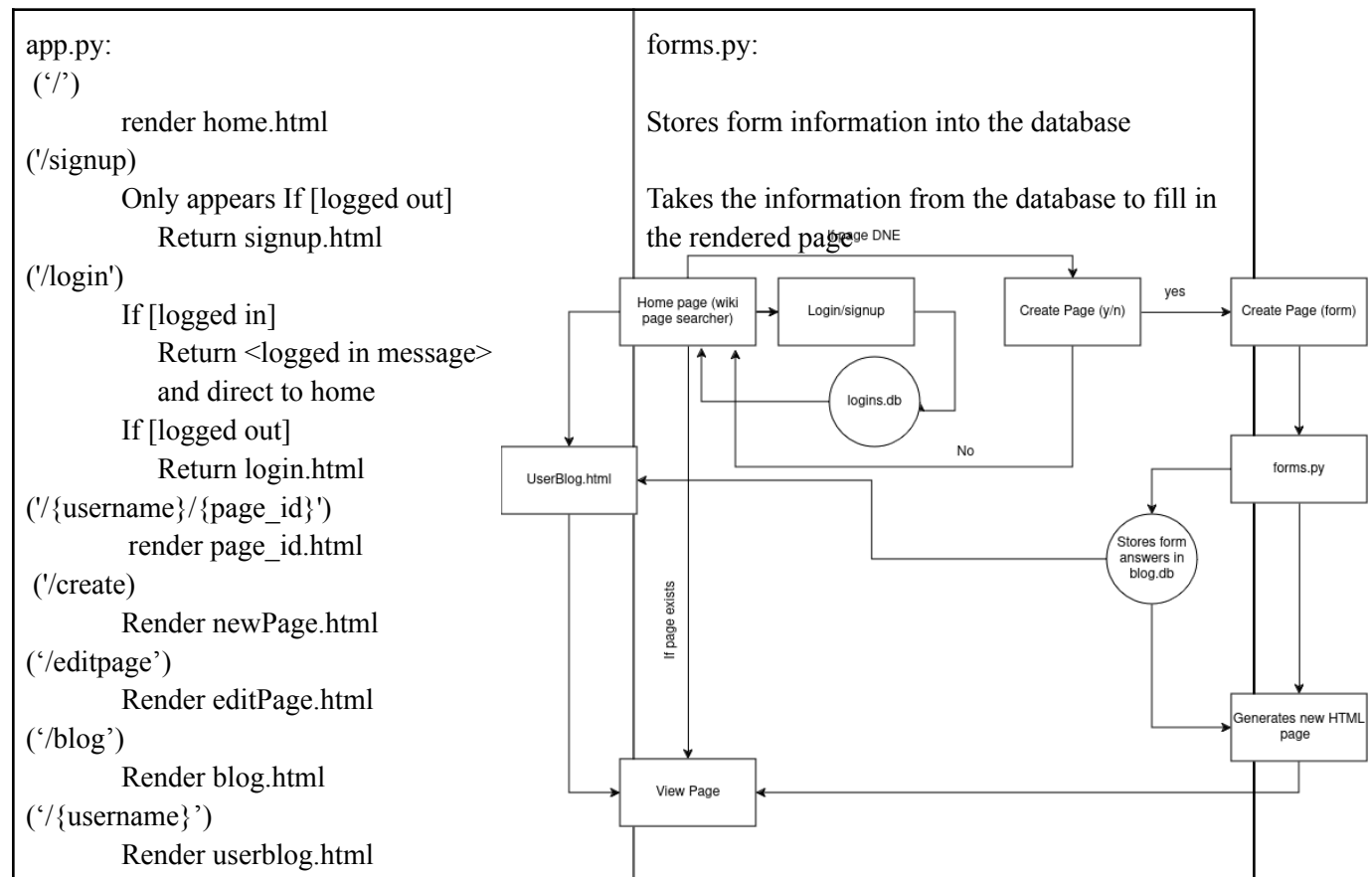
Shows list of past entries stored in database

Blog: (/ {username}):

Shows list of blogs made by {username}

## Task Assignment:

<p>Front end: Claire and Tiffany</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> newPage.html - <i>{page with forms and runs forms.py upon submission; form answers = blog.db record and fields }</i></li> <li><input type="checkbox"/> editPage.html</li> <li><input type="checkbox"/> signUp.html</li> <li><input type="checkbox"/> login.html</li> <li><input type="checkbox"/> home.html</li> <li><input type="checkbox"/> blog.html (list of past entries)</li> <li><input type="checkbox"/> userBlogs.html (view the blogs of other users)</li> </ul>	<p>Backend: Jessica and Jackie</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> blog.db (stores page data)</li> <li><input type="checkbox"/> app.py (main script)</li> <li><input type="checkbox"/> forms.py (generates the new pages)</li> <li><input type="checkbox"/> logins.db (stores login data)</li> <li><input type="checkbox"/> edits.db (stores past edits)</li> </ul>
---	---



## Database Organization:

### Blog.db

	User_ID	Page_ID	Page Title ▲	Page Details
	Filter	Filter	Filter	Filter
1	Stores ID of blog maker	Stores a generated ID of the page	Rendered title	Rendered page details

### Logins.db

Username	Password	User_ID
Filter	Filter	Filter
Stores the username of the user...	Stores the password for the user	Stores the generated ID

### Edits.db

Page_ID ▲	Version_ID	Version_Old	Version_New
Filter	Filter	Filter	Filter
Stores a generated ID of the page	Each number represents a different version of the page	Contents of old version of the page	Contents of the new version of the page

## Page Draft

<div><h2><u>Home (Page name)</u></h2><div><a href="#">Home</a> &gt; <a href="#">Users (dropdown of letter ranges)</a> <a href="#">My Pages</a> <a href="#">Log Out</a></div><div><div><a href="#">Login</a></div><div><a href="#">Sign-up</a></div></div><div><a href="#">Possible search bar?</a></div></div>	<div><h2><u>Users</u></h2><div><a href="#">Home</a> &gt; <a href="#">Users (dropdown of letter ranges)</a> <a href="#">My Pages</a> <a href="#">Log In/Out</a></div><div><a href="#">A - L</a> <a href="#">User1</a> <a href="#">User2</a> <a href="#">User3</a></div></div>
<div><h2><u>My Pages/User Pages</u></h2><div><a href="#">Home</a> &gt; <a href="#">Users (dropdown of letter ranges)</a> <a href="#">My Pages</a> <a href="#">Log In/Out</a></div><div><a href="#">Page1</a> <a href="#">Page2</a> <a href="#">Page3</a></div></div>	<div><h2><u>Page1 [Title]</u></h2><div><a href="#">Home</a> &gt; <a href="#">Users (dropdown of letter ranges)</a> <a href="#">My Pages</a> <a href="#">Log In/Out</a></div><div><a href="#">Page Details here [if own page, edit button here]</a> <a href="#">[if editing, pre-filled textbox with previously saved details here]</a></div></div>

## Site Map:

Home (/)

- Outgoing routes:
  - Login (/login)
  - Sign-Up (/signup)
  - View Blog (/username/{page\_id}) for viewing a specific blog

Login (/login)

- Outgoing routes:
  - Home (/) if successful login
  - Sign-Up (/signup) if the user doesn't have an account

Sign-Up (/signup)

- Outgoing routes:
  - Login (/login)

View Blog (/username/{page\_id})

- Outgoing routes:
  - Home (/) to return to the main blog list
  - Edit Page (/page\_id/edit) if the user is logged in and wants to edit their own post

Create Page (/create)

- Outgoing routes:
  - Home (/)
  - User Blogs (/username) to view all their created blogs

Edit Page (/page\_id/edit)

- Outgoing routes:
  - View Blog (/username/{page\_id}) after editing is done
  - Home (/) after editing is done

View Past Entries (/viewpast)

- Outgoing routes:
  - View Blog (/username/{page\_id}) to view specific entries
  - Edit Page (/page\_id/edit) if editing past entries

User Blogs (/username)

- Outgoing routes:
  - Create Page (/create)
  - View Blog (/username/{page\_id})

