Team The Bored: Alvin Wu, William Yin, Jeffrey Huang
SoftDev
P3:
2021-04-23

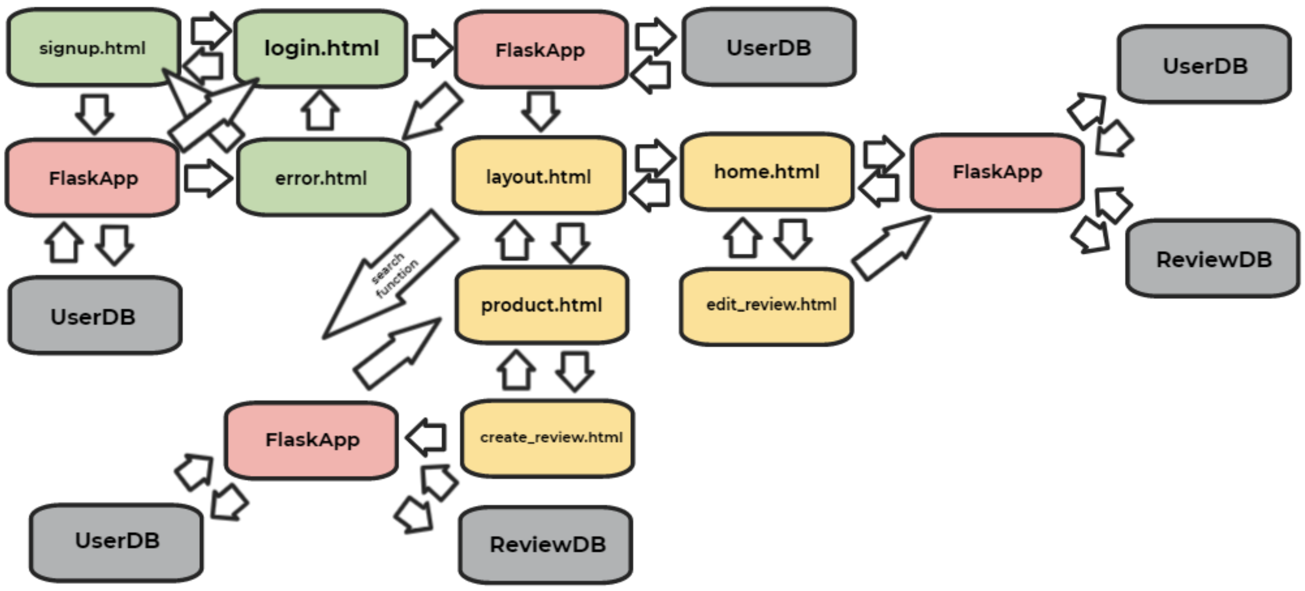
Product Review Site

Components

- README.md:
    - Clearly visible at top: **<Project Name> by <Team Name>**
    - Roster with Roles
    - Description
    - Launch Codes:
        - How to clone/install
        - How to run
- SQLite Database:
    - Table 1: Users
        - ID
        - Username (unique field)
        - Password
        - Biography
    - Table 2: Reviews
        - ID (non-negative auto-incrementing integer)
        - User ID
        - Product (Either book/movie/recipe)
        - Product ID
        - Rating
        - Title
        - Body
- APIs Used:
    - RecipeAPI
    - NYTBooksAPI
        - 4,000 requests per day
        - 10 requests per minute.
    - OMDBAPI
        - Search by IMDB ID
        - No rate limit
- Flask App:
    - connects backend files (i.e. database) to frontend files (i.e. html templates). This Python script will redirect users to another webpage based on the buttons they click and their input in HTML forms.

- Templates:
    - signup.html: contains an HTML form that allows users to create a username and password for their account. Users will create a short biography for themselves. The backend will check if the usernames chosen by new users already exist in the Login table and verify if they fulfill other requirements (i.e. password length). Users will be redirected to login.html or error.html depending on whether their account was created successfully.
    - login.html: contains an HTML form that allows the user to enter username and password. When the user clicks the “Submit” button, app.py will check if the credentials match an entry of the Login table in the SQLite database. Users will be redirected to layout.html or error.html depending on whether they logged in successfully.
    - error.html: If users fail to login or create an account successfully, they will be directed to this page and be told what went wrong. Below the error message, there is a “Sign Up” button that redirects the user to signup.html and a “Login” button that redirects the user to login.html
    - layout.html: contains the main layout of the page and has a logout button. Also contains a search bar to allow users to search for products, entering a category and then a search term to call using APIs
    - home.html: contains the user’s past reviews and most highly rated movies/books/recipes
    - product.html: displays the product being reviewed and the most recent reviews for it
    - create_review.html: form for a user to create a review for a product
    - edit_review.html: form for a user to edit one of their reviews

Component Map



Database Organization

- Users table:
  - There will be a row for each user.
  - There will be 4 columns: ID, Username, Password, Biography

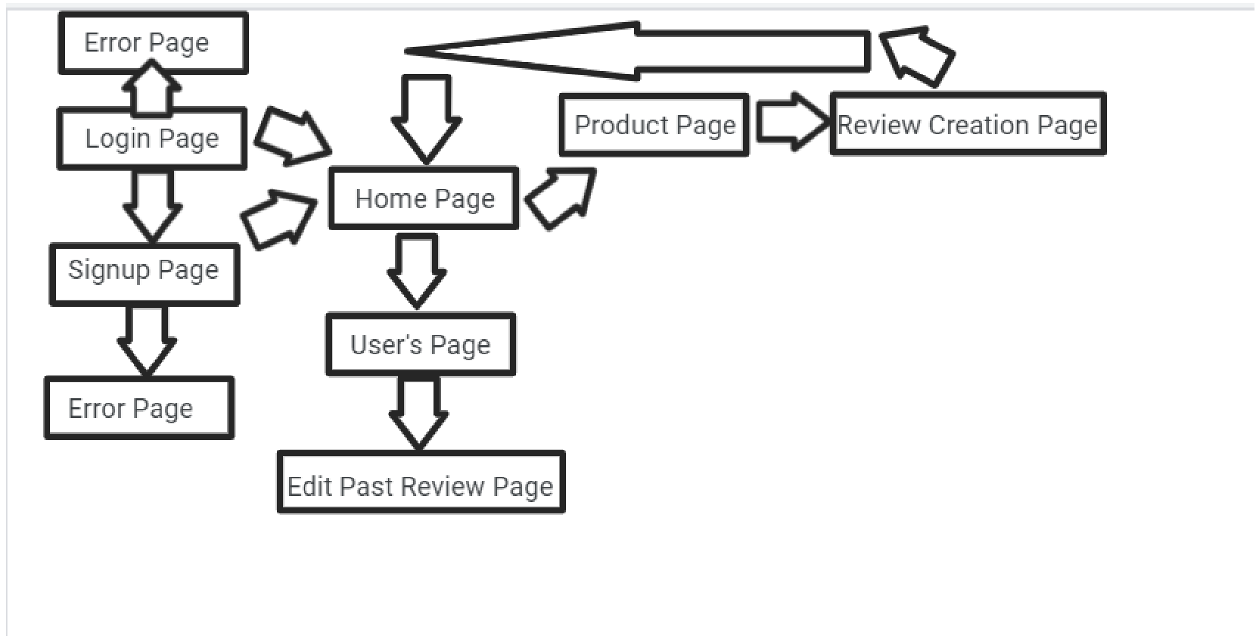
ID	Username	Password	Biography	ID’s of all User reviews
0	blogger1	t0g2ka	blogger1 is currently a journalist working for the NY Times ...	1~3~5~10
<non-negative integer>	<string>	<password>	<biography>	<string of ID’s separated by ~ >

- Review Table
  - There will be a row for each movie, book, recipe (reveiw)
  -

ID	Title/Name (of	Rating (1-10)	Review by User	Product Type

	thing)			
0	Grilled Cheese	7	Great tips and recipe but it doesn't specify the exact time on medium heat ...	Recipe
<non-negative integer>	<string>	<integer>	<string>	<string>

Site Map for Frontend



Tasks

- Create Flask app with different routes and functions:
  - Render Login Page
  - Render Error Page
  - Login Function
  - Render Main Page
  - Logout Function
  - Signup Function
  - Render Signup Page

    - Review Creation Function
    - Render User Profile Page
    - Edit Function
    - Product Search Function (APIs)
    - Database and Table Creation
- Create HTML files/templates
- Generate and store API keys
- CSS and JS (Where necessary)

Timeline

- Basic templates created and finished by 4/26
- Basic Flask app created by 4/26
- Bug fixing and front-end development runs from Flask app creation until 4/29