Comp 266 Unit 5 – Writing JavaScript: Program Design

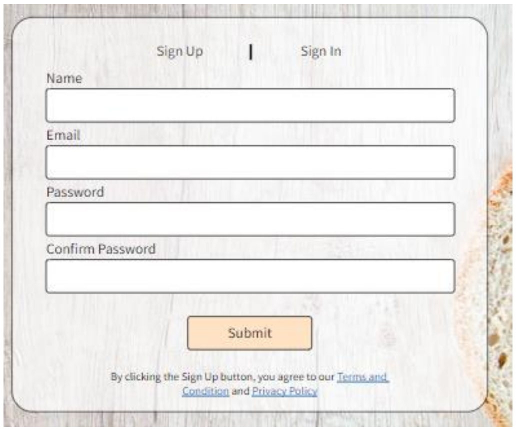
Recipes on Hand

Design Process for each Idea

Design 1: Sign Up/Login Form

The main core of this JavaScript code will encompass the sign up and login classes along with their functions. The logic consists of a set of **conditional statements** that checks the forms if they contain valid text. Several **constant methods/functions** will be created for each error checks, such as checking if a field is blank or empty, or if the field is invalid. I plan on creating an **object** for the user which can be initialized through a **constructor** with parameters for their name, email, and password. These users can be stored as an **object array** to store the information of all registered users. This initialization will hold the user's information and can be called upon when necessary.

Class "SignUp":



By clicking the Sign Up button, you agree to our Terms and Condition and Privacy Policy

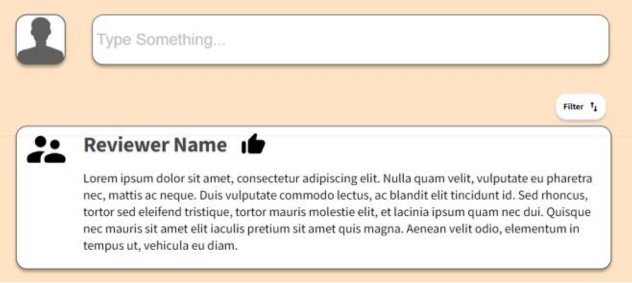
- String variables: "name", "email", "password"
- Functions: "signUpVerify()", "register()"
- Program flow:
  - User selects the Sign Up tab, an event listener for the "SignUp" class will trigger the sign up form to active.
  - The user fills out the fields and presses the Submit button, creating a new instance of the "SignUp" class with the parameters in the fields.
  - The function "signUpVerify()" is called to perform error checks:
    - If successful, call the "register()" function to create a new instance of "User" and add it to the User array.
    - If unsuccessful, display an error on the respective fields.

Class "SignIn"

- String variables: "email", "password"
- Functions: "signInVerify()"
- Program flow:
  - User selects the Sign In tab, an event listener for the "SignIn" class will trigger the sign in form to active.
  - The user fills out the fields and presses the Submit button, creating a new instance of the "SignIn" class with the parameters in the fields.
  - The function "signInVerify()" is called to match with the User array and perform error checks:
    - If successful, direct the user to the recipe list page.
    - If unsuccessful, display an error on the respective fields.

Design 2: Comments under Recipe page and posts under Discussions page

The comments under the recipe page differ from the posts under the discussions page. The main difference being the inclusion of a title in the posts for the discussions page. The class made for the comments or posts can have **two different constructors** with **different parameters** depending on the page that called it. However, designing two different classes for their intended purposes will likely be a cleaner method. In order to filter by ascending or descending time, there needs to be a **timestamp** recorded for each comment made, so this will be one of the parameters passed to these posts. **Object arrays** will be used to store the comments and posts, and **functions** will be called to load this information when the page is active.



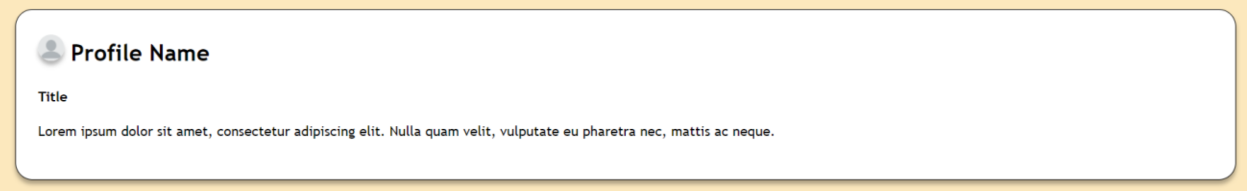
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam.

Class: “Comment”

- String variables: “name”, “content”
- Object variables: “timestamp”
- Functions: “commentVerify()”, “commentSubmit()”, “loadComments()”, “sortByTime()”
- Program flow:
  - User loads the recipe page and the “loadComments()” function is called.
  - When the user enters the comment and submits it, a new instance of the “Comment” class is created with the information within the parameters.
    - If the comment is valid (mainly not empty) through the function “commentVerify()”, the “commentSubmit()” function is called.
    - Otherwise, an error is displayed.
  - The function “loadComments()” is called to render the comments list again from the “Comment” array.
  - The filter button is a toggle to sort by the “timestamp” object which is a parameter for the object “Comment”
    - When pressed, the array is re-ordered and “loadComments()” is called to display the comments in the proper order again.

Class: “Post”

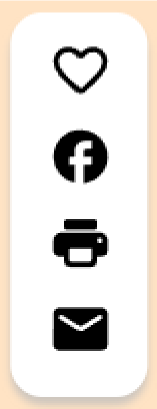
- String variables: “name”, “title”, “content”
- Object variables: “timestamp”
- Functions: “postVerify()”, “postSubmit()”, “loadPosts()”, “sortByTime()”
- Program flow:
  - Similar to comments, but with the inclusion of the “title” parameter.
  - Similar verification, loading process, submission, and variable datatypes will be used.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque.

Design 3: Recipe Page Side Bar Functions

All the side bar functions are based on button interactions, so once an onClick() event is triggered, the respective **function** required for the behaviour is called. Only one of the side bar functions is tied to the user and the recipe page itself, and that is the like button. This is because the page needs to recognize that the user liked the page already and cannot allow another like from the same user. Also, a count must be kept for the recipe page itself. An **object** will be created for the likes function to hold the user list, like count, and like status as parameters. A **function** will be used to handle the status of the like button and update the counter. The other buttons require interactions with external websites, applications, and commands, thus more **functions** will be used to perform these interactions.



Class: “Recipe”

- String variables: “recipeID”
- Object variables: “recipeLikes”
- Functions: “adjustLIkes”, “shareFB”, “printRecipe”, “emailRecipe”
- Program flow:
    - If the user presses the like button, “adjustLikes” is called to increment or decrement the counter based on the status of the button.
    - If the user presses the Facebook share button, “shareFB” will be called to open the Facebook webpage. This function will utilize the Facebook share button API to share content from the recipe page.
    - If the user presses the print recipe button, “printRecipe” is called to print the Recipe portion of the page. This will be achieved by identifying the division of the page that must be shared, and selecting it within the print area.
    - If the user presses the email recipe button, “emailRecipe” is called to open the default email client on Windows. This can be achieved through Mailto links.

References

How to create Mailto Links. in HTML. (n.d.). Retrieved February 10, 2023, from https://www.w3docs.com/snippets/html/how-to-create-mailto-links.html

Inc., K. (2021, December 22). Sharing content on social media by JavaScript. Medium. Retrieved February 10, 2023, from https://medium.com/@knoldus/sharing-content-on-social-media-by-javascript-263db75f2c91

Window print() method. (n.d.). Retrieved February 10, 2023, from https://www.w3schools.com/jsref/met_win_print.asp