Technical Features of “Drink Counter”

# Product Documentation:

## System Documentation:

## Compatibility:

Devices: PC (IOS)

Browsers: Chrome

Additional Hardware: Trackpad, Mouse, Touchscreen, Keyboard (Login/Sign up only)

### Currently NOT Compatible/Untested On:

Devices: iPhone, Android, other handheld devices, Linux, PC (Windows), tablets (Ipad)

Browsers: Firefox, Opera, Safari, Internet Explorer/Microsoft Edge

Additional Hardware: Stylus + Drawing Pad

# Files

Languages used to code the app:

* HTML
* CSS
* JavaScript

The central file (the webpage) is called “index.html”.

The linked styling is called “homeScreen.css”, the project also uses JQuery with JavaScript, it is in a separate folder called “css” inside the “homeScreen” folder.

The JavaScript file, called “homeScreen.js” is for all the global functions that also interact with the database. It includes all the variables, functions and data that is to be used in the app. These files are in the “js” folder inside the “homeScreen” folder.

There is also an images folder which contains all the images used in the app.

# Function:

There are many “global” functions that is used throughout the app, they are generic functions and that can pass multiple parameters. These include:

All Screens:

* “loading” is a function that shows a loading screen for a certain amount of time and used almost every time when data is being process to and from the external database (rest.db). This is due to the lag between the fetching the database and pushing information into it, so a pause is needed for the function to complete so that the app is useable.

“Fhide” is a function that hides the loading screen, it is used so that a “setTimeout” function could be used to delay the hiding of the screen.

A parameter called “trans” is passed to the “setTimeout” function to indicate how late the delay will be.

Login Page:

* “getSignUp” is a function that gets the sign up from an external database (rest.db) using the url and api key. It stores the entire database into an array of objects “arrLogin” for access.
* “addSignUp” is a function that uploads information entered in the sign up screen to an external database (rest.db). It calls the “getSignUp” function to summon the database using url and api key and uploads the information that was put in it.
* “validateSignUp” is a function that runs when the button “signUpSubmit” is pressed. This function checks for whether the entered sign up information is valid and then submits it to the database and logs the user in. “arrLogin” is the array ran through the function, as it is the array that the function gets its information from.

Various local variables are created within the function to serve as temporary variables to hold the login information for validating and before submitting it to the database, such as:

* + “rePassword”
  + “username”
  + “password”
  + “fname”
  + “lname”.

A boolean “found” is used to check if errors are found within the submitted information. These include:

* + Empty/unfilled fields
  + Password and confirm passwords not matching
  + Username already existing within database

If an error is found, an error message will be shown. If everything is validated, “use” will be set as “username” to log in, then all the temporary variables will be uploaded into an external database (rest.db) for future use by using the function “addSignUp”. It then hides the Login screen and shows the Home screen.

* “login” is a function that runs when the button “loginSubmit” is pressed. This function checks for whether the entered login information is valid and logs the user in. “arrLogin” is the array ran through the function, as it is the array that the function gets its information from.

Various local variables are created within the function to serve as temporary variables to hold the login information for validating, such as:

* + “username”
  + “password”

A boolean “found” is used to check if the correct login information is submittededThese include:

* + Whether the username matches an existing username
  + Whether the corresponding password and confirm passwords matches the entered password.

If an error is found, an error message will be shown. If everything is validated, “use” will be set as “username” to log in, then hides the Login screen and shows the Home screen.

Home Screen:

* “datePicker” is a function that finds the current date today as well as allow users to choose their own date (Variable setting). It runs immediately when the program starts.

It uses the function “getMonth”, “getDate”, and “getFullYear” to find today’s date upon access to the app and displays it in the format “YYYY-MM-DD” in the div “day” and keeping it in the temporary variable “output. It also sets the variable “date” to the current date. Users can choose their own date by clicking on the div “day”, which hides the element “day” and shows the elements “todayDate” and “submitDate”. Then, the user is able to change the date in the input “todayDate”, confirming and submitting it by clicking the button “submitDate”. It then replaces the current “day” display to the chosen date as well as update the “date” variable to the chosen date (“output”), it also hides elements “todayDate” and “submitDate” and shows the element “day” again.

* “checkDrink” is a function that checks whether the specific drink choice and drink goal has been chosen before allowing “addDrink” to upload into database. It runs when either “HundredMl”, “twoFiftyMl”, “sixHundredMl”, “eightFiftyMl”, or “litre”is clicked

Using the flags “checkBev” and “checkGoal” to confirm, if both of the flags aren’t true, an error message displays. Otherwise, it allows “drinkMeterSet”, “drinkData”, “drinkMeterMoveAmount” and “getDrink” to run.

* “getDrink” is a function that gets the drinks data from an external database (rest.db) using the url and api key. It stores the entire database into an array of objects “arrDrinkData” for access.
* “addDrink” is a function that uploads information entered in the sign up screen to an external database (rest.db). It calls the “getDrink” function to summon the database using url and api key and uploads the information that was put in it.
* “drinkData” is a function that is run within “checkDrink” that uses case to switch the specific beverage type in correspondence to the image shown on screen and the div that was clicked. It will then store the data in a “tempDrink” temporary variable and uploaded onto the Drinks database using “addDrink.
* “drinkMeterSet” is a function that is run within “checkDrink” that uses case to switch the goal setting, so that different goals can be chosen when the user wants to. It changes the “Goal” variable.
* “myFunction” is a function (not jquery) to toggle “differentDrinks” to visible.
* “drinkMeterMoveAmount” is a function that is run within “checkDrink” that uses case to switch the amount of width added to “drinkMeter” in correspondence to the drink amount input button clicked. It does the calculation for the drink bar movement amount (340/(Goal/’set integer’)) and sets the drink bar width to the correct amount. If the calculation is larger that 340 and will move off the bar, it will automatically be set to 340. Also, it changes the drink goal display to the percentage of completion as well as show the total amount of beverage intake (ML).
* “slider” is a function that moves the navigation bar when “navibtn” is clicked. It changes the width of “navigation” as well as hide and show the divs inside of it.

Insights page:

* “sdatePicker” is the same function as “datePicker” but for the statistics page.
  + is a function that finds the current date today as well as allow users to choose their own date (Variable setting). It runs immediately when the program starts.

It uses the function “sgetMonth”, “sgetDate”, and “sgetFullYear” to find today’s date upon access to the app and displays it in the format “YYYY-MM-DD” in the div “sday” and keeping it in the temporary variable “soutput”. It also sets the variable “sdate” to the current date. Users can choose their own date by clicking on the div “sday”, which hides the element “sday” and shows the elements “stodayDate” and “ssubmitDate”. Then, the user is able to change the date in the input “stodayDate”, confirming and submitting it by clicking the button “ssubmitDate”. It then replaces the current “sday” display to the chosen date as well as update the “sdate” variable to the chosen date (“soutput”), it also hides elements “stodayDate” and “ssubmitDate” and shows the element “sday” again.

* “plusUser” Is a function that runs when “btnDay” aka. Submit is pressed. This function passes through and filters the entire “arrDrinkData” array, pushing all objects that have a matching username within the object with the variable “use” from the login into an new array called “arrUser”.
* “plusDate” Is a function within the function “addAmount”. This function runs after the “plusUser” function and passes through and filters the entire “arrUser” array, pushing all objects that have a matching date within the object with the variable “sdate” from the login into an new array called “arrresult”.
* “selectionSortAmount” is a function within the function “addAmount”. This function runs after the “plusDate” function and has a parameter called “inputArr” that passes through the entire “arrresult” array. It sorts the array by ascending order of the “Amount” integer inside of the objects within the array. (This is an selection sort)
* “addAmount” Is a function that runs when “btnDay” aka. Submit is pressed. It first runs the functions “plusDate” and “selectionSortAmount”, It then adds all the “arrresult.Amount” together, as well as set “sgoal” to “arrresult.Goal”. It also sets “sdrinkMeter”, “stotalBevStat”, and “sdrinkGoal” back to zero for a reset. It then does the calculation for the drink bar movement amount (340/(goal/total)) and sets the drink bar width to the corresponding amount. If the calculation is larger that 340 and will move off the bar, it will automatically be set to 340. Also, it changes the drink goal display to the percentage of completion as well as show the total amount of beverage intake (ML).
* “addName” Is a function that runs when “btnDay” aka. Submit is pressed. It is the same function as “addAmount” expect it extracts all the specific types of drinks and does their calculations individually. It first gets and adds all the specific types of drinks together (amount) and stores it in a temporary variable. It also sets all the meters and counters to a blank state. It does. It then does the calculation for the drink bar movement amount (340/(goal/total)) and sets the drink bar width to the corresponding amount. Also, it changes all the counters display total amount of the specific beverage intake (ML).

# Variables

Most variables and elements use camel case. All global variables are declared at the top of the file ( “homeScreen.js”)

## Arrays

Arrays all start with the prefix “arr”.

There are multiple relevant arrays in the program. Ones that contain the gotten databases and others that hold important information.

* “arrDrinkData” is an array that stores the entire Drinks collection in the database when the “getDrink” function runs, to hold the information for further calculation.
* “arrLogin” is an array that stores the entire Login collection in the database when the “getSignUp” function runs, to hold the information for further calculation.
* “arrresults” is an array that stores the entire calculated database after all the calculations within the code.
* “arrdrinkGoal” is an array that contains objects holding the drink goal informations.
* “arrUser” is a transitional array that just acts as an object holder.

## Boolean

There are various flags used throughout the JavaScript code.

“dateVisible” – used to determine if “todayDate” is visible or not, so that the “datePicker” function is able to show and hide the div.

“sdateVisible” – used to determine if “stodayDate” is visible or not, so that the “sdatePicker” function is able to show and hide the div.

“found” – check if user exists or not in the Login collection in the database.

“checkBev” – used to determine if a specific beverage was selected or not.

“checkGoal” – used to determine if a specific goal was selected or not.

## Strings

Strings are used in the code to both display on screen and/or upload to the database in an object.

Example:

“fname” is a string containing a person’s first name to be uploaded into the Login collection in the database.

“output” is a string used to display the date in the div “day”

## Integers:

Integers are used both as calculations or displayed on screen.

Example:

“Goal” contains an integer of the goal (amount of ML for intake), it is used for calculations.

“total” contains the total amount of beverage intake for the day for calculations, but also displayed in the statistic screen for users to see.

## Objects/Classes

Objects are used in arrays, which are mostly to upload corresponding information into a database.

“tempDrink” stores the Date(str), Name of beverage(str), Cups drank(int), User(string), Goal(int), together, making a temporary object to upload onto the drinks collection in the database using “addDrink” function.

“tempSignUp” stores the Fname(str), Lname(str), Username(str), Password(str), Cpassword(str), together, making a temporary object to upload onto the Login collection in the database using “addSignUp” function.

# Databases:

The app uses 1 database with 2 collections. Both uploads and returns data as an array of objects. The database is an external database connected with to rest.db with an API key.

The collections in the main databases are:

1. Login – This collection collects and stores all the Login information that was entered while signing up. This collection includes:
   * “Fname” – First name (string)
   * “Lname” – Last name (string)
   * “Username” – Chosen Username (string)
   * “Password” – Chosen Password (string)
   * “CPassword” – Confirming chosen Password (string)
2. Dailydrinkcount - This collection collects and stores all the Drink information that was entered by the user from the home page. This collection includes:
   * “Date” – Date (string)
   * “Name” – Name of drink (string)
   * “Cups” – Cups drank (integer)
   * “Amount” – Amount of liquid intake per cup (int)
   * “User” – Username (string)
   * “Goal” – Goal (integer)

Link:

<https://github.com/pymblelc/mw2022-ayang0326>

<https://pymblelc.github.io/mw2022-ayang0326/>