

API DOCUMENTATION

File location: `src/components/navbar.component.js`

Function Name: `toggleFontSize`

Input: event

Output: nothing

Functionality: Adds or removes classes that increase an element's size with css.

Function Name: `signOut`

Input: event

Output: nothing

Functionality: Calls server function `logOut()`;

File location: `src/components/journal.component.js`

Function Name: `countChars`

Input: event

Output: nothing

Functionality: Upon entry into the journal textarea, the number of characters left is updated underneath it. When the number of characters left reaches 0, the font turns red.

Function Name: `addHabit`

Input: event

Output: nothing

Functionality: Gives the habit table its functionality. Upon pressing the + button in the journal page, a textbox is opened allowing input. Should that button be pressed again while the textbox is opened, whatever was inside it will be added as a habit. Creates a new row and checkbox for the habit and it is formatted into the proper database object and added to an array of objects to be sent to the database.

Function Name: `toggleDeleteButton`

Input: event

Output: nothing

Functionality: Showing or hiding delete buttons conditionally

Function Name: `checkedHabit`

Input: event

Output: nothing

Functionality: Sets values to a habit in the array if it becomes checked or unchecked accordingly. Also handles habit deletion if the delete buttons are clicked on.

Function Name: updateEmail

Input: event

Output: nothing

Functionality: Creates the url for creating the email. Grabs the value of the email textbox and concatenates "mailto:" with it.

Function Name: onChangeMood

Input: event

Output: nothing

Functionality: Changes the page's shell color based on which mood you pick upon clicking a mood. Also adds the mood to the object that will be sent to the database.

Function Name: onChangeHabit

Input: event

Output: nothing

Functionality: Also adds the array of habits to the object that will be sent to the database.

Function Name: onChangeDailyText

Input: event

Output: nothing

Functionality: Removes non-ascii values from being added upon entry. Also adds the textarea response to the object that will be sent to the database.

Function Name: onSubmit

Input: event

Output: nothing

Functionality: Asks the user if they would like their input sent to them via email. Does so if "Ok" is pressed and does not if "Cancel" is pressed. Regardless, it sends the information gathered to the database.

File location: src/components/weeklyView.component.js

Function Name: onEditPage1

Input: event

Output: nothing

Functionality: Uses the calculated date for today to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage2

Input: event

Output: nothing

Functionality: Uses the calculated date for yesterday to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage3

Input: event

Output: nothing

Functionality: Uses the calculated date for two days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage4

Input: event

Output: nothing

Functionality: Uses the calculated date for three days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage5

Input: event

Output: nothing

Functionality: Uses the calculated date for four days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage6

Input: event

Output: nothing

Functionality: Uses the calculated date for five days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage7

Input: event

Output: nothing

Functionality: Uses the calculated date for six days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: onEditPage8

Input: event

Output: nothing

Functionality: Uses the calculated date for seven days ago to get database information for that date. Then opens the weekly journal view to view the collected information and edit it.

Function Name: loadJournalEditPage

Input: nothing

Output: nothing

Functionality: Displays information in a stylized way as the page is shown.

Function Name: onBackButton

Input: event

Output: nothing

Functionality: Sets gathered information values to default and sends the screen back to the weekly view screen.

Function Name: onSubmitJournalEdit

Input: event

Output: nothing

Functionality: Submits edited data. If there was no previously created journal, it forms an object from the data and sends that. Also redirects back to the weekly view page.

File location: src/components/archivePage.component.js

Function Name: onBackButton

Input: event

Output: nothing

Functionality: Sets gathered information values to default and sends the screen back to the archive page screen.

Function Name: requestPerDate

Input: beginDate, endDate

Output: array of journals received from the database

Functionality: Gets every existing journal for every day between the two dates given and adds them to an array to be returned.

Function Name: onSubmit

Input: beginDate, endDate

Output: array of journals received from the database

Functionality: Calls requestPerDate on the two dates inputted and generates a page that uses the information received.

File location: src/components/signInPage.component.js

Function Name: changeUsername

Input: event

Output: nothing

Functionality: Updates username value in an object for sending to the database to the value of the textbox labeled username.

Function Name: changePassword

Input: event

Output: nothing

Functionality: Updates password value in an object for sending to the database to the value of the textbox labeled password.

Function Name: onSubmit

Input: event

Output: nothing

Functionality: Sends the object with sign in information to the database.

File location: src/components/signUpPage.component.js

Function Name: changeUsername

Input: event

Output: nothing

Functionality: Updates username value in an object for sending to the database to the value of the textbox labeled username.

Function Name: changePassword

Input: event

Output: nothing

Functionality: Updates password value in an object for sending to the database to the value of the textbox labeled password.

Function Name: changeEmail

Input: event

Output: nothing

Functionality: Updates email value in an object for sending to the database to the value of the textbox labeled email.

Function Name: onSubmit

Input: event

Output: nothing

Functionality: Sends the object with sign up information to the database.

File location: src/features/RequestUserData.js

Function name: addJournalPost

Input: Journal data structure

Output: Returns back journal data if the data was successfully added and console.log message to confirm a successful server request.

Functionality: Allows for a user's Journals to be updated and added with a single function. The function is asynchronous and uses axios to make a request to route "journals/add". The function also looks for a JWT token in localStorage named "user."

Function name: getJournalPost

Input: date={day,month,year} (all values must be integers)

Output: Returns a null value if the get request is unsuccessful, else the function returns a journal on the given date.

Functionality: The function is asynchronous and uses axios to make a request to the route "journals/get." It is used for retrieving journal information on a given date. The function also looks for a JWT token in localStorage named "user."

Function name: getHabitPost

Input: date={day,month,year} (all values must be integers)

Output: Returns a habit post given a date.

Functionality: The function is asynchronous and uses axios to make a request to the route "habits/get." It is used for retrieving a list of habits on a given date. The function also looks for a JWT token in localStorage named "user."

Function name: getHabitPost

Input: date={day,month,year} (all values must be integers)

Output: Returns a null value if the get request is unsuccessful, else the function returns a habit list on the given date.

Functionality: The function is asynchronous and uses axios to make a request to the route "habits/get." It is used for retrieving a habit list on a given date. The function also looks for a JWT token in localStorage named "user."

File location: src/features/auth/AuthService.js

Function name: Login

Input: user={username,password}

Output: Saves a JWT token to the localStorage as the item "user"

Functionality: Sends a username and password to the server via axios on route "users/login". If the database has the combination of username and password, then it saves a JWT.

Function name: logOut

Input: none

Output: Returns a message if the localStorage is cleared

Functionality: Deletes the JWT in localStorage so the database can no longer be accessed.

Function name: register

Input: user such that user={username,password}

Output: Saves a JWT token to the localStorage as the item "user"

Functionality: Sends a username and password to the server via axios on route "users/add". If the database has the combination of username and password, then it saves a JWT. It additionally saves the user's profile to the database.

File Location: middleware/authMiddleware.js

Function name: isValidToken

Input: JWT token

Output: A valid username if it exists

Functionality: Decodes a JWT to see if it is valid. If a username and password can be extracted, it determines if the account for the data exists. If none exist it returns null, else, it returns a username.

File Location: routes/users.js

Function name: createToken

Input: user such that user={username,password}

Output: Returns a JWT token

Functionality: Encrypts data into a JWT token.

File Location: src/components/archivePage.component.js

Function name: requestPerDate

Input: startDate, endDate such that each is structured as such {day,month,year}. Each value should be an integer larger than 0.

Output: A list of journals between the start and end dates given to it (inclusive)

Functionality: Requests the server to get every journal between the start and end dates.