

Contributions:

Alex: I helped Eli with adjusting the given code to form a baseline for our admin platform. I wrote HTML files to display the content and create forms. I also used CSS files to improve the user interface design of the data entry form.

Anjali: I worked on deleting a user. In order to delete a user, I created a link on the "Show all" page that takes you to a new page - delete.html. Then, I created a delete.html page that contains a textbox and a submit button. The admin must input the username of the User that needs to be deleted and then click a submit button. Jack and I also worked on connecting Express to MongoDB.

Carly: I edited Person.js to conform to our description of a user, adding in fields such as username, password, accountNum, and the various scores generated through our wellness assessment. I also helped link our nearly complete node.js project to mongoDB so our data could be accessed by everyone instead of just stored locally.

Eli: I worked on adjusting the given code to form a baseline for our admin platform. I adjusted person.js to instead work for our desired User class by adding in a number of fields (username, password, mental health attributes). I also adjusted a few routes in index.js to work with the new updated User class.

Jack: I wrote and extended on a handful of functions that allow the admin to create, edit, and view all of the users in our database. I used personform.html as a baseline for our HTML page, added in all of the correct fields, and wrote created.ejs as a land page for when the user has been created. This page will congratulate the admin for creating a user and list all of his/her data. In the all.ejs page, which the user can access by pressing "Show All" button, I made each users name a hyperlink that when pressed will take the admin to a user specific page that will allow the admin to edit the users data. This will properly save the changes to the database when the admin finishes editing a user, and after completion, the program will take the admin to updated.ejs which will describe the new fields of the user that was just updated. The admin can

now go back and create another user, or view all of the users which have properly been saved to mongoDB.

User Stories

1. As an administrator, I want to be able to see all users in a database. (1)
2. As an administrator, I want to be able to add users to the database. (1)
3. As an administrator, I want to be able to remove users from the database. (1)
4. As an administrator, I want to be able to click on a user to see their full data. (1)
5. As an administrator, I want to be able to edit a user's profile data. (1)
6. As an administrator, I do not want to allow the creation of users with the same username. (1)
7. As an administrator, I do not want to allow the creation of users with a blank username, password, or name. (1)
8. As an administrator, I want to allow the creation of users with blank data for sleep, physical fitness, stress, and community. (1)
9. As an administrator, I want to interact with a simple and aesthetically pleasing interface when entering data. (1)
10. As an administrator, I want to be able to see the reflection of changes made on other administrator devices in real time. (1)

Commit Log

Commits on Apr 8, 2019			
added deleted functionality anjalimaheshwari committed 3 minutes ago	Verified	85e3f9a	<>
updated html files anjalimaheshwari committed 4 minutes ago	Verified	9755f0f	<>
added files including EJS files jackg235 committed 9 minutes ago	Verified	97e1ee0	<>

We have a short commit log because we met up multiple times throughout the week and primarily worked on a single laptop for simplicity.