

START UP ASSIGNMENT 07

Special Setup Procedure

- We do not have any special server setup procedures.
- Developer can change user id from (1->6) by clicking on the Console icon in the navigation bar.
- Reset Database button is put inside the dropdown menu that can be accessed by clicking on the Console icon in the navigation bar.

Individual Contribution

Thien

GET /messagebox/:box_msg_id/participantlist

This route handles accesses that require connection to the Mongo database to retrieve a list of participants chatting in a message box using its id, and once the participant list is retrieved from the matched document in the collection 'messagebox', the server returns that list to the client.

GET /messagebox/:box_msg_id

This route receives a message box id from the client. After that, it retrieves data of the requested message box from the matched document in the collection 'messageboxes' in Mongo database and return the data of the message box to the client.

POST /messagebox/:box_msg_id/send/:user_id

This route handles messages that are posted by clients. Once the message is received at the server, it is saved into Mongo database; more precisely, it will update an array containing messages in the message box document from the collection 'messageboxes'.

GET /users/:userid/recentmsgboxes/:numberofboxes

This route handles requests that retrieve a number of recent message boxes that the user with the given id has participated. The server performs find query to the collection 'messageboxes' in the Mongo database to get data and return it to the client.

PUT /messagebox/create/:user_id

This route handles requests by the client to open a new message box, which will create a new document in the collection 'messageboxes' in Mongo database, to chat. After the message box is created, it is returned to the client.

PUT /messagebox/:box_msg_id/add/:user_id

This route handles requests by the client to join a new participant into the current message box, which will update the existing document in the collection 'messageboxes' in Mongo database, so that the new participant can chat in the same message box.

Timothy:

Feature: Implemented search over feedItems in the MongoDB, elaborated below. Also improved the styling of the welcome pages i.e. the academic and service category pages respectively. The categories now zoom in on hover to create a simulate an animated effect.

POST /search /queryText

This route handles searching for feedItems. Before implementing MongoDB it was quite challenging to query text from both academic and service feed. However, after implementing MongoDB, we can simply query text from both types of feeds by using the \$or operator specifying the fields of the database to look in.

JianYi:

Feature: Setting Page of the website, it will initially display the username, password and email of the user. Then the user can change these attributes by editing on the corresponding box.

HTTP Routes: Get / config / :userid -> view the current user's username, password and email.

Put / config / :userid -> change the username/ password/ email.

Karen:

Feature: Profile page of the site displays the first and last name of user, profile picture, education level, academic institution, area of interest, and classes taken. When the user makes changes in the textbox, the save and cancel button will appear on the bottom of the page. When save is clicked the new info will be saved to the database and refreshing the page will reflect that. If cancel is clicked then any changes will disappear. Add classes and remove classes functionality also add. When user clicks add classes and modal box will appear and user can check off the classes they want to add. If a class is already in the list of classes taken by the user then adding that class again will not change anything. When user clicks remove class a modal box will appear and user can check off classes to remove. If the class that the user selects is not in the list of classes taken then no changes will happen. The user must click the save button to keep changes

or cancel to discard changes. Also added the edit profile picture functionality. When user hovers over profile pic, it blurs and if the click on the image or on the upload button they can change their photo. User must click save so changes go to database.

HTTP Routes: Get /user/:userid/profile -> handles connection to mongoDB to retrieve user information (name, education level, school, fav quote, classes, and interest) and return it to the client

PUT /user/:userid/profile -> handles connection to mongoDB to save user information

Changed the getUserData function in server side to interact with the mongoDB

Jucong:

Feature: Whenever a user post a comment, new requests, it will automatically write into the database. Retrieve data is bit different from before, since there is more powerful mongo function that we can use to filter data we need. Like addset and \$or. I add new page name My request, this help the user to organize all his feeds, which very useful when user try to delete a request he posted. The other function is stay the same without the database.

HTTP Routes: Only add get/:user/requests page. This will return all the request that user had requested before.

Xin:

Feature: The schedule page is a personal calendar for the user. User can add schedule to that page through making an appointment in the message page as well as adding a schedule in the schedule page itself. User also is able to delete the the schedule Item if they complete the task. When user add the schedule, user has type correct first name (the first letter has to be capital letter which is corresponding to the database). Otherwise, it will get auth error.

HTTP Routes: Get/schedule/:userid -> handle connection to mongoDB to retrieve user's schedule data (username, date, time, service content and partner) and return it to the client.

POST/schedule -> handle connection to mongoDB to adding a new schedule item data including username, date, time, service content and partner and return it to the client.

Delete/schedule/:userid/:scheduleid -> handle connection to mongoDB to delete a schedule Item showing on the user's schedule page and return it to the client which means the page will be refreshed without that schedule item just deleted.

Lingering Bugs / Issues / Dropped Features

Issue in schedule page : The button “Add to schedule” in schedule page occasionally works. It just requires manual refresh sometimes. However, user usually add a new schedule through making an appointment in the message page. This one works perfectly, so this bug doesn't affect a lot.

Issue in search results:

A little bug that persists is that even though the search function is functional, the results make the error banner appear, saying `Could not GET /user/undefined/profile/: Received 401 Unauthorized: .` It seems as though, there's a get request to profile with an undefined user. However, I check if user posting the request is valid, in the search function. So, I'm not entirely sure how this error came about.