Special Server Setup Procedure

Individual Contributions

- Ye Sung (Rebecca) Kim:
    - POST /feedback: migrate this route to use MongoDB, this route is coupled with the function postFeedbackData. It is authorized for the author of that feedback.

- Anh Bui:
    - Get /user/:userid/interviews: use MongoDB to get all the interviews for a user, authorized for that userid
    - Set up MongoDB, and reset database functionality
    - Fix significant bugs

- Tri Nguyen:
    - POST/endinterview: migrate this route to MongoDB by implementing postAfterInterviewData helper function to use MongoDB.

- Tien Dao:
    - POST /searchpeople route: migrate this route to MongoDB
    - GET/user/:userid route: migrate this route to MongoDB by implementing getUserData helper function. Fix bugs from previous submission
    - GET/user route: migrate this route to MongoDB implementing getAllUserData helper function

- Ngan Hoang:
    - POST /interview route: Implemented the helper function for this route, postInterviewSession, with MongoDB queries. Fixed little things in the app.post function for the route.

- Thanh Pham:
    - GET/interview/:interviewid: migrate this route to MongoDB by implementing getInterviewSession helper function to use MongoDB.

Lingering Bugs / Issues / Dropped Features

- We are still looking into WebRTC, and we have a rough idea of what to do. We think we will be able to find a suitable way to demo this functionality next week.
- When the server matches users for interviews, it doesn’t care whether the user is currently in the matching pool or not. This feature will be implemented when we have a proper user authentication.

- Since there is no user authentication on the roadmap, we won’t be able to implement location tracking for users.

Honors features

DELETE /chatsession/:chatid/memberlists/:userid : deleteChatMember(), remove chat member from the chat session (the button is disable when you visit another user’s session).
PUT /chatsession/:chatid/memberlists/:userid : addChatMember(), add chat member to the chat session (the button is disable when you visit another user’s session)
DELETE /notification/:notificationid : deleteNotification(), delete a notification in the database using the ID, authorized only for the current login user
PUT /notification/:notificationid/status/:status : updateNotificationStatus(), update a notification’s status in the database using the ID, authorized only for the current login user
POST /chatmessage : postChatMessage(), post a chat message, authorized only for the current login user checked with the owner of the chat message
GET /chatsession/:chatid : getChatSessions(), get the chat session using the ID, authorized for the current login user
POST /notification : postNotification(), post a new notification, authorized for the requester of the notification
GET /onlineusers : getOnlineUsers(), get a list of online users authorized for the current login user
GET /user/:userid/nearby : getNearbyUsers(), get a list of user nearby (within a default radius 10000), authorized for the current login user
GET /user/:userid/notification : getNotifications(), get notifications that the user needs to see (using the requestee field in the notification), authorized for the user with that userid.

- Change GET /user/:userid/online => GET /onlineusers
- Now adding and deleting chat member is only authorized for the user with the right token that matches chat session initiator.
- Now finding nearby users only search upon onlineUsers group.
- The problem with authorization for adding chat members and deleting chat members is complicated because if I match the initiator to the auth token. Then, when the other side accepts a notifications, it doesn’t work.

BUGS

- When we accept a notifications, it doesn’t automatically update the page, so that the check mark turn into a button to go to the chat session.