## Progress Report: React - Pair'd

Guan (Raymond) Chen, Brent Condon, Katie Ho, Ella Holmes, Helena Kleinschmidt, Wesley Liao

Include a report in reports/submission 06.pdf in your client repository with the following information.

## **Special Server Setup Procedure**

As in the previous submission, if your server has any advanced features that require additional setup besides npm install and node src/server.js, include a section in the report that describes how to set up these features. Include details on how to tell if the feature is working properly.

N/A

## **Individual Contributions**

Include a section that describes each startup founder's contribution. Each startup founder should be responsible for at least one product feature and its HTTP route(s). Name the feature and the HTTP routes that the founder updated to use the database. Each founder *must* be responsible for converting at least one HTTP route.

Katie: I worked on the HTTP route app.post('/search', function(req, res) {. This route allows users to search open deals from the navbar. Currently, the query text is only matched with deal titles and deal links. When users type a query into the search bar, they are taken to a new page with their particular search results. The query text appears in the url, like it did in Facebook. To do this, we first got all the users and then went through the users to get their individual open deals. Then, we search within all the open deals and match query text with deal titles or deal links. I also pair programmed with Ella and Helena to migrate the search and postDeal methods over from client to server files. Earlier on in this submission, I also implemented some of the features from our class's Kanban style Notes app to allow fields on our deal profile and user profile pages to be editable, recognizable as editable when hovered over.

Helena: I worked on the HTTP route app.put("deal/:id", function(req, res) to go along with the Deal Profile component I am responsible for. This route handles pairing a user with a specific deal posted by another user (the owner of the deal). It updates the deal with the id of the pairee in the database. It also updates the pairee's closed deals list with the given deal id, and moves the deal id from the owner's open deals list to its closed deals list. There is one hard-coded part of the HTTP route I wrote, where paireeld is set to "000000000000000000000001"; this is because I had to use Postman to test my route, as the way to navigate to the button that activates my route in the UI is not operant (because the route the gets deals is broken).

Ella: I worked on the HTTP routes app.get("deal/:id", , function(req, res)), app.get("deals/:ids", , function(req, res)), app.get("user/:id/opendeals", function(req, res)), app.get("user/:id/closeddeals", function(req, res)), and took on the challenge of creating the helper method getDealData. These route happen in home, user profile, deal preview, and get a specific or a series of deals data. getDealData does not run correctly and as a result all four of these route do not function properly because of the dependency. Additionally to these features, I did much of the migration from client/server to server/src/server (i.e. most of workshop 6) along with Helena and Katie. After completing the migration we reached out to the other group member and assigned everyone to a route that they were then going to transfer over to read from mongodb.

Brent: I worked on the HTTP routes for POST('/resetdb') and PUT("/user/:userId/deal/:id/:field") which respectively reset the mongo database and look up and change a field of a deal. I also set up the resetdatabase.js by copy and pasting from our database and by using the functions given in the startup submission. I wasn't able to test out the

PUT feature within the website; however, I did test it in postman which returned the number of changes made in the database. I also created a function updateDealData which does the updating for the PUT command mentioned above. The deal profile page will attempt to update the deal after any changes are made in the text boxes.

Wes: I worked on the HTTP route app.post('/deal') which allows for a new deal to be created and posted to the database with all relevant information. I seperated post new deal from deal profile, which required creating a new page. I reimplemented the post deal page with several functions that support inputting a deal image, inputting a date (required conversion with date objects) and posting a new deal. Currently, by navigating to the post new deal page from the navbar, the user is able to fill in the data fields (assuming it is done correctly) and select post deal, which successfully writes the new deal to mongoDB. The route also uses findOne to access the deal owner's information and update his/her open deal list accordingly with the new deal. The route still uses an unencrypted HTTP token to check if the user has access to perform the operation.

Raymond: I worked on the HTTP route app.put('/user/:id/:field'), along with the function updateUserData. This route allows the user to edit a certain field within their profile. I also worked on the HTTP route get('/user/:id') which retrieves user data for a specific userId from the database, along with the helper functions getUser and getUserData. Currently, the route is not functional.

## **Lingering Bugs / Issues / Dropped Features**

If your application has any lingering bugs, issues, or missing/dropped features, include a section listing these. Similarly, if any particular features or components are not using the database, please mention those features here.

- Biggest problem: getDealData helper function in src/server does not return the deal data correctly. This will
  get fixed ASAP and fix many of the errors we come across (also related to search problems)
- Instead of having a universal deal profile page for edit, view and posting a new deal, we now use two separate pages. One page for viewing or editing an existing deal, and another page for posting a new deal.
- The "post a new deal" page assumes that the user inputs the data correctly (Error messages do pop up for incorrect inputs, but it does not catch all of them). Some data inputs (like date or tags) require a certain format in order to properly work.
- The search from the navbar does not yet match query text to deal descriptions or deal tags.
- The following routes do not successfully use the database, get('/user/:id'), get('/deal/:id'), put("/user/:id/:field"), put("/user/:id/closeddeals"), put("/user/:id/closeddeals"), post('/search')
- Note: our intent when you use our application is for you to be logged in as Abu and can look at a guest view of Sleeping Beauty.