

# **React Components**

## **Profile Page**

- Profile
  - Constructs the whole profile page by calling on the subcomponents to render below. Only part of the profile that has to communicate with the server, by calling `getProfileData` and then passing the data down to the parts of the page that need them.
- Navbar
  - Renders the top navbar of the page, currently accepting the single user's name and displaying it on the User's button.
- Sidebar
  - Render's the signed in user's projects and the job board. Dynamically renders the appropriate number of projects depending on the projects they're a part of by using the `ProfilePills` component.
- Profile\_About
  - Simply renders the left hand column of the profile zone containing the user's biography and contact information.
- Profile\_Banner
  - Render's the user's custom banner image and profile picture.
- Profile\_Skills
  - The (observing profile) user's skills and interests are generated inside of a bootstrap button group (respectively, not together) but the button's use a CSS scheme to prevent the ability to click them, because they should not be clicked. The bootstrap buttons have a nice aesthetic and button grouping, so I chose to use them. Also renders the user's name due to the central location of this column.
- Profile\_Teams
  - Render's a pills list of teams the (observed) user is a part of. Should be clickable and react by linking to that project page, but not yet implemented.
- ProfileButton
  - Render's an individual profile button. This is a nonclickable entity, intended to render one for each of the user's skills.
- ProfilePill
  - Render's an individual project pill. This should be a clickable entity, intended to render one for each of the projects the user has been a part of.

## Known Profile Bugs

In rendering the react linking to other pages, the pills on the sidebar work in linking to the project page, but now the pills in the Profile\_Team component do not render. They seem to be appearing as 0x0 pills, and the data is still there. When commenting out the <Link> tag in the ProfilePill component, they do in fact render again. This tag has also lead to a small sliver of a button being clickable above each sidebar pill that will fail to bring the user to the project page, even though when clicking on the projects themselves they do in fact link to the page. (Both working and bug button seem to request the same routing.)

Due to the size of the images attempting to be stored in browser, currently no profile images, banners, or home button in the upper left hand corner of the navbar is rendering. Anticipate this being fixed when moving the images off of the browser and onto a server.

## Main Feed

NOTE: We are in the process of removing followed project updates from the main feed.

- Main-Feed
  - The main feed utilizes the feed and project items from the database.js. The users have associated feeds that have lists of project items and job items. These populate the UI as feed items.
  - **BUGS:** We are working through issues where we cannot get the data to render properly on the page. Hopefully it is fixed by the time you read this.
- Main-Feed-Filter-Bar
  - The filter component has check boxes that filter the different feed items.
  - **BUGS:** This is not functional yet as feed items aren't rendering yet.
- Main-Feed-Job-Item
  - These items utilize the job items in the database. There are associated tags, titles, and descriptions for them.
- Main-Feed-Notification-Item
  - These items utilize the notification items in the database. There are associated titles and descriptions for them.
- Main-Feed-Tag
  - This component creates a tag for an associated job item. It directly utilizes the list of tags from the job item.
- Main-Feed-Tag-List
  - This component creates the list of tags.

## Inbox Page

- Inbox Page

- This component grabs data from database.js for inbox, which is an array of chats.
- This component contains Inbox Items and Message Items components.
- **Inbox Items**
  - Inbox Items grab data from Inbox Page and creates new Inbox for each chat in the array.
  - This is the left side of the page which is the list of different chats the user is involved in.
  - When clicked on, the message items should change to corresponding chat that is clicked on.
  - The “read” color of the inbox panel changes to green from orange when clicked on.
  - **BUGS:** The color of the inbox panel changes to blue when clicked on, but it does not change other panels back to white.
- **Inbox**
  - Inbox grabs the chat data from database.js, which is an array of messages.
- **Message Items**
  - Message Items gets data from database.js and creates each message for each message in the certain chat array.
  - **BUGS:** Message items do not change according to the chat that is clicked on. And I was only able to pull one message from the database.js.
- **Message**
  - This component grabs the message data from database.js which contains the author, the message content, and the side of display.
- **Message Entry**
  - This component is the text box for sending the message.
  - **BUGS:** It does not update the chat with new messages yet.

Bug that was fixed from last workshop is that now the inbox side and the message side of the page extends down all the way to the bottom of the screen no matter what size screen it is. I originally had this for the side navbar too, but while merging all the css files to be on one single css, it went back to leaving space in the bottom. I'll look at this again because it is showing differently for different screens.

We got rid of “sending files” and “receiving files” system. So with that, I deleted the paperclip icon next to the send button. And originally, we had user icons for team chats, but that messed with alignments and we couldn't figure out how to only include icons for inbox(chats) with more than 2 people. So we decided to not include the user icons for anything on the inbox side.

While working on this startup, I realized that making chats an array of messages and inbox an array of chats separately and calling them by id number was a bad idea. It would have been easier to embed it all in database of inbox. There are conflicts in calling the whole array of chat data and setting it in this.state and calling the whole array of messages data and setting it in this.state. Because of this issue, right now it's only rendering one message and one inbox at a time. In future startup, I plan to fix this by changing up the database.js, rewriting the functions for getting those data from database.js, and pulling in the whole array as props to create each messages. Then it will also work better with writing new messages and sending it.

As soon as I have those figured out, I plan to also incorporate time into each messages.

### **Create Project Page**

- Create-project-tag-item  
A single tag that will be used to identify a project when a search is performed
- Create-project-tag-section  
The section where tags may be entered, edited, and saved
- Create-project-single-position  
A single position a job may require, also used to identify the project when a search is performed
- Create-project-position-section  
The section where job position requirements may be entered, edited and saved
- Create-project-checkbox  
A binary operator that a user can select or deselect to determine if the project is in progress.

Dynamically implemented create project page. This page will be used to create test items for all other pages.

Bugs: while the page is interactive and adapts dynamically, it does not yet save to the database. The create project button does not yet create a project because of this. This will be finished by the next startup.

### **Project Page**

- Project Page
  - The Container for all the smaller project components
  - Top part of the page is the Project Banner

- Bottom Left part of the page is the Project Description
- Bottom Middle part of the page is the Project Messages/Updates
- Bottom Right part of the page is the Open Positions for the project
- Grabs project data from the database.js file and passes it onto the other components
- Project-Banner
  - Displays the projects customizable banner image and the project's name
  - BUGS:** The image is not dynamic and can't be changed as we are not currently storing photos in the database. If big enough the name of the project will get cut off the banner
- Project-Description
  - Hold all the Project information including the description, relevant skills for the project, and the current admins of the project
  - Grabs the filled position information for the projects from database.js
- Project-Member-Item
  - Displays the the information of each project admin
- Project-Skill-Item
  - Displays the skill for each skill item
- Project-Messages
  - Displays all of the recent project messages/updates
  - Grabs all of the updates/messages from database.js
- Project-Message-Item
  - Displays each individual message/update for the project
- Project-Positions
  - Displays all of the currently open positions for the project
- Project-Position-Item
  - A single open position being displayed in the position section of the page

## Contributions

Ryan Minichiello: Profile Page and Color Redesign. Dynamish Sidebar and Navbar, react-render implementation

Nick Achin: Main Feed

Caroline Kim: Inbox Page

Ryan LeCours: Create Project Page

Josh Tranfaglia: Job Board Page

# React Routing

The main-feed is the home page rendering at 'http://localhost:8080/'. From here:

