Key React components:
navbar.js: (always visible on the top of the page) displays the navigation bar, which includes the following buttons: “Reset Mock DB”, “Interview” to start an interview (which will link to the page created by component Matching), a search bar used to search for other users(currently leads to a page with a list of users), a dropdown menu “USERNAME” with a “Log out” button and a “Profile” button to view the profile info (link to the page created by UserProfile component, containing user’s profile).

homepage.js: (access to the page: clicking on the logo in the far left corner of the navbar) The page displays the home page of the web. Displays some general information about some recent interviews of the user. Clicking on the username at the top left corner will lead to his/her profile page.

userprofile.js: (access to the page: clicking “Profile” button in the “USERNAME” drop down menu or clicking on the username in some other components mentioned later in this report) displays the cover photo and a profile picture of the user, in addition to some general information about the user, such as email address, skills, and location, and a link to their history page, which will show the list of previous interviews.

peopleprofile.js: (access to the page: searching in the search bar in the navbar) displays the list of all the users with their profile picture and information. Clicking on one of the usernames will bring the user to their profile page.

matching.js: (access to the page: clicking on the “Interview” button in the navbar) To be assigned the appropriate interview partner, user will submit some preferences and information about himself/herself, including the language he/she is most comfortable with, what kind of position he/she is looking for, and how many years of programming experience he/she has had. Then the user will click on “Find me an interview!” button to proceed to an interview (the button will lead to the interview page).

interview.js: (access to the page: clicking on the “Find me an interview!” Button in the matcing page) This is the page for two users to conduct the interview practice. When a user finishes the interview, he/she clicks on the “End Interview” button to proceed to the feedback page.

a) interviewquestions.js: a text box that display questions from the interviewer
b) codeeditor.js: a text box that displays the answer code from interviewee
c) voicechat: Tool for communication between the interviewer and the interviewee

feedback.js: (access to the page: clicking on the “End Interview” button in the interview page after the interview ends) User fills out the feedback information, including pros, cons, any additional comments, and rating, for their interview partner and clicks on “Submit Feedback.” The “Submit Feedback” button will store the information filled out by the user in the database and bring the user back to the homepage.

history.js: (access to the page: clicking on the “See History” button in the profile page) The page displays the list of previous interviews of a users in the component PastInterviewsList.

a) pastInterviewLists.js: lists the history of previous interviews by timestamp in the top left corner of the page. When user click on an interview it will display the feedback of that interview on the right side of the page in component InterviewFeedback.
b) interviewfeedback.js: contains the information of the past interview with the profile of the interviewer and the interviewee, the interview questions asked, along with the solution and the answer given by the interviewee, and the feedback and rating given by the interview partner.

Member contributions:
Tri Nguyen: History component & child component, write report.
Anh Bui: Added server, database, NavBar and HomePage component, routing and integrate everything together.
Ye Sung(Rebecca) Kim: Feedback component.
Thanh Pham: Interview component.
Ngan Hoang: Matching component.
Tien Dao: PeopleProfile, UserProfile component.

Current problems:

1. The search bar has not worked yet. When you type in and search for a user, it will display the PeopleProfile page (list all users).
2. Voicechat has not worked yet.
3. UserProfile: CSS bug - profile not in center. When clicking on “See History” button from any user profiles, it will only lead to the same History page.
4. Matching hasn’t successfully updated the database. I also plan to load a user’s existing preferences and data from the database as default option on the dropdown.

Honors Section:
Meetup.js (access : React component that will render a Google Map object with markers of nearby users (the radius is static for now), you can click on the markers and press request to post new notifications to the system, so that the users that you are requesting can receive the notifications.
Meetupchat.js: React component that will render the chatting interface when users want to organize a physical meet up. Currently the component only communicate with the server to get the onlineUsers, there is no database entities for a chatSessions/ messages yet.

- Also in the navbar.js, ask the server for all the notifications for the current user and render notifications cards accordingly.

Honors Current problems:

1. Like mentioned, the radius to search for nearby users is still static.

2. Since we are only logged in as one user, we can't go through an integration test to make sure the notifications will be transmit properly when another user request for us.
3. The buttons on each of the notification cards aren't doing anything since there is no business logic around chatSessions yet.
4. chatSessions and messages haven't been fledged out, I had to help my fellow teammates since they were struggling, and so I haven't been able to figure out this complicated components, it would be nice if I can get some help with the architecture around this feature.