1) An Overview of the Function of the Code

Our project focused on making improvements to the LiveDataLab service used in CS410. The description for this project's choice stated that we would be given access to the current website and could progress from there; however, our group was unable to acquire this, leading our group to code from scratch. Instead, we not only made changes/improvements to the site, but also an entire rework of it.

When run using 'npm start,' our code launches a locally hosted webpage of our redesign of LiveDataLab. It has a similar structure to the current live version of the site, but with some changes that we think will improve it. Notably the following:

Home Page:

- Added list of projects the user is working on/worked on, apart from submission history for the user to access these from the home page.

Projects Page:

- Added additional information about leaderboard score/placement that is visible without having to enter the project's specific page.
- Changed layout to be easier to read and able to have more projects listed while taking up less room

Courses Page:

- Changed the UI for a more organized way of accessing a list of courses, keeping the original functionality.
- Links to individual course pages, displaying associated projects if the course was specified on creation.
- UI is more organized and occupies less screen space for a better view of the page.

Manage Page:

- Like the Courses Page, changed the UI for a more organized way of accessing the lists of courses and projects.
- Links to individual course pages, displaying associated projects if the course was specified on creation.

Create Dropdown – the create button is a dropdown toggle where you can select to add a "Course", "Project", "Leaderboard", or "Submission". For each of these, a pop-up form will appear where you enter the relevant information. The info will then be sent to Firebase for storage and can be retrieved by the other pages.

Sign out – the sign out button will sign the user out of the application and route back to the authentication page.

2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement

   Our software is implemented using React JS, an API that will be detailed in the following portion of the documentation. Below is the file structure of our project, separated by folders

   - Node_Modules
     - Contains all the files installed by React
   - Public
     - Contains different images and icons that are used on the page, also the manifest.json file which maintains information about these imported images and icons
   - Src
     - The main portion of the directory, contains most of the files and folders that create the page itself
     - Assets
       - Contains additional logos and images
     - Components
       - Contains the Navbar.css and Navbar.js files, these create and structure the navigation bar at the top of the page consistent across all tabs
     - Config
       - This folder contains the firebase configuration, firebase being the 3rd party database that stores user information
     - Hooks
       - The hooks folder contains calls to the Firebase API to store and retrieve data and handle user authentication.
     - Pages

        - Contains folders for each page of the site, including all the different tabs, the different create pages, the user login page
        - Each of these folders contains a .css and a .jx file that create and structure the different pages of the site
    - App .js and .css
        - Responsible for controlling the navigation between the pages, i.e. links the different pages to their respective buttons on the navbar

3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

Installation and Running

1. To run the software, you will need to first install Node.js at the following site:

2. Then install the following packages using *yarn* or *npm* package managers
    a. react-router-dom
    b. firebase
    c. react-bootstrap
3. In directory, navigate to livedatalab_firebase and run npm start

Using the application

1. Once you start the app, it will ask you to login with Google. The Google login window should pop up and the user will select the appropriate Google account.
2. If the authentication is successful, you will be routed to the home page where you will see the LiveDataLab logo in the upper left and your Google profile picture and username in the upper right.
3. From the home page, you will see the navigation buttons on the top. From the navigation, you can go to the “Projects” page, the “Courses” page, the “Manage” page or the “Create” dropdown.

APIs

The following APIs are used in the application

- **Firebase Firestore** – this is a NoSQL cloud database provided by Google. This is where the data is stored in various "collections" depending on if it is a course, project, leaderboard, or submission.
- **Firebase Authentication –** this API manages the user authentication of the user, using their Google account.

4) Brief description of contribution of each team member in case of a multi-person team

Collin:

- Uploading submissions for each portion of the project

- Creating proposal / part of documentation / presentation video

- Projects page of the site

- Navbar layout and design

- Assisted other members on their portion of the site when needed

Seth:

- Created and implemented the backend architecture using Firebase

-Wrote the user authentication hooks using the Firebase Authentication API

- Created the "Authentication Page" and "Sign Out" button

-Wrote the API "hooks" for storing and retrieving data from Firebase

- Created the "Create" pages using React Modals which included "Add Course", "Add Project", "Add Leaderboard", and "Add Submission"

- Structured and organized the codebase

Jonatan:

- Worked on “courses” page based on the implementation of Seth’s backend architecture to display course name and course description.

- Worked on “manage” page based on the implementation of Seth’s backend architecture to display courses (name and description) as well as projects (name and description).

- Started work on base navbar with links to the different pages, later improved by Collin.

- Worked on “coursesPage” to display each course (name and description) as well as its associated projects if chosen on creation of project.

- Modified “Add Project” to allow for course to be initialized correctly to later access on CoursePage.

Ty:

- Worked on site Homepage to display information related to a user's projects and linked accounts.
- Worked on Homepage to display a list of submission data that can be filtered on each submission field.