Acceptance Test for User story:

cd into the iCare folder, run `npm install` to download all the dependencies you'll need. Then to run the react server you will need to `npm run start` .

Open a new tab in your terminal and cd to iCare/server and run `nodemon app.js` (or node) to run the backend.

1.
open up localhost:3000 to see the login page, to test whether the login page works, you can enter the credentials, 'email@email.com' with the password 'pass' and press the submit button to get redirected to an upload page.

To test the upload file functionality, while on the /upload page, there should be a clear file drop zone outlined with a dashed border. To test the functionality of this file upload, simply upload any file from your computer by dragging & dropping a file, or clicking and selecting a file from the file explorer and check the console if file has been selected and check the server/public to see if it is there.

2. Login as 'otheruser', 'hello', click the query in the nav bar on the bottom left and test the query drop down by selecting the options provided. Make sure the options are there and is clickable

3. Run the steps for userstory 2 test and make sure that there is a download button. Download it and make sure that the file has successfully downloaded

4. Login as 'email@email.com' with password 'pass' and click on the query page on the navbar. A large selection of options will be presented, click as many as you want and make sure that the tables appear when clicking the enter

5. Login as 'email@email.com' with password 'pass' and click on the create new account page on the navbar. Insert the fields and click enter 'create'. Make sure an alert box appears that says it has successfully been entered.

6. Login as 'email@email.com' with password 'pass' and click on the change password on the navbar. Choose an existing email and try changing the password for that user, make sure an alert box appears that says it has successfully changed the password

Code Review Strategy:

Our strategy will focus on the readability, maintainability and reliability. As a team, we will try to identify issues regarding styling, functionality and design.Team members should keep these following questions in mind for the PR:

1. Is there sufficient documentation?
2. Is the code style clean and consistent?
3. Does the code use appropriate naming conventions?
4. Does the code have any magic numbers?
5. Does the code follow SOLID design?
6. Can we implement design patterns?
7. Can one easily understand what the code does?
8. Is there unnecessary debug statements?
9. Is there error handling?

Code Review:

PR:
https://github.com/CSCC01/Team9/pull/4/files

-Code style is well done without weird spacing / indentation issues so it is very easy to real.

-Componentization is well done as shown on Page.js, easy to reuse component and apply css

-There is unnecessary code in loginpage that can be cleaned up. Some functions do not get called as it is in the child component so that needs to go

-Naming convention follows the guidelines and is clean

-Css is easy to use with descriptive names to ensure fast and easy use when applying to pages

Overall, it looks good and the only change requested would be the unnecessary code