Project: Milestone 3

Name: Sean Allgaier

UIN: 01138928

1. Overview

Currently, my website is a simple UI that prompts users to register for courses. The information to keep track of the registration information is being stored in a database via the back-end. The project is written with a combination of Node, React, JavaScript, HTML, and CSS. It is using the Express framework. The database is being hosted on Clever-Cloud, the backend is being hosted on Render.com, and the frontend is being hosted on Firebase.

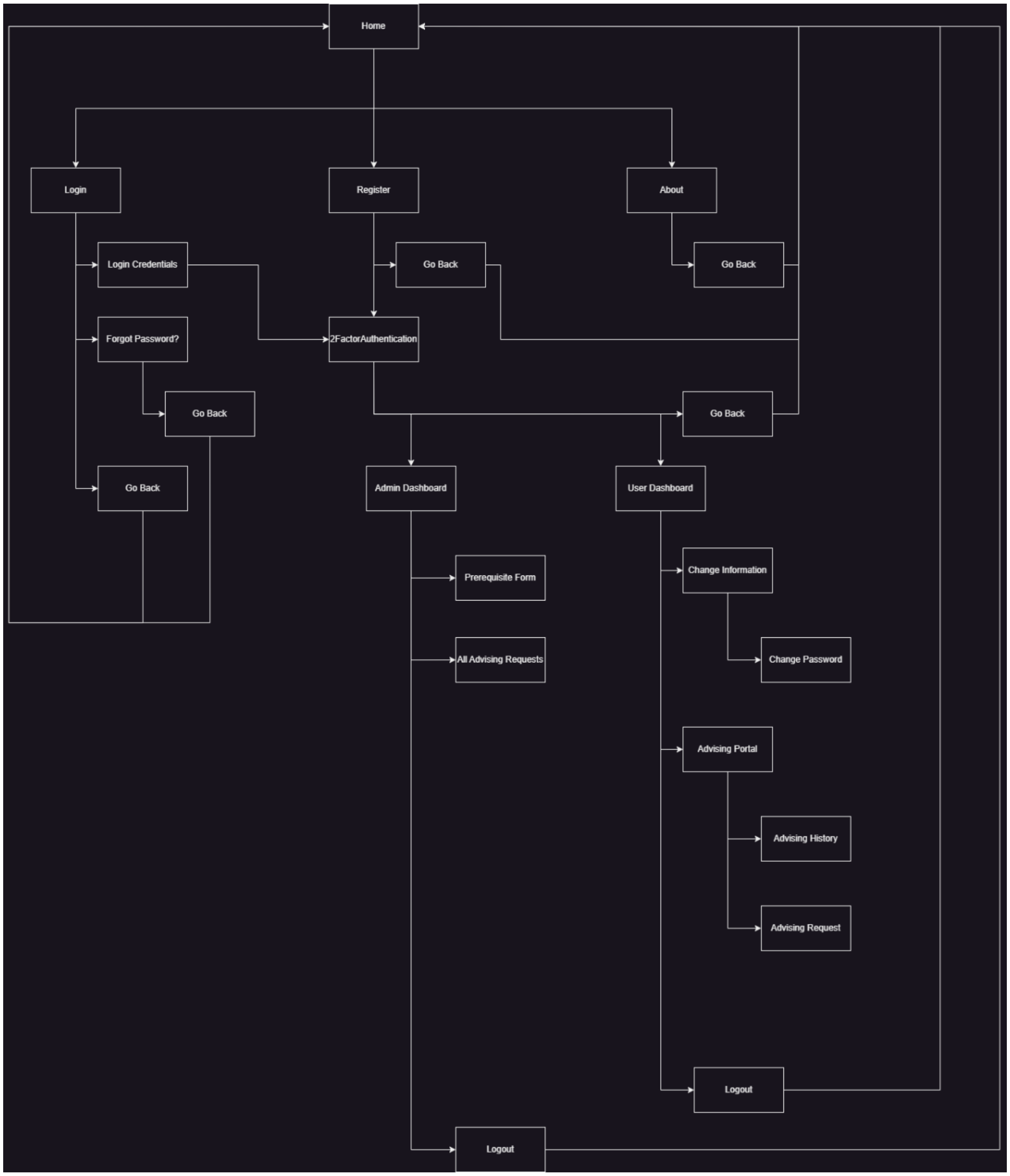
2. Milestone Accomplishments

Fulfilled	Feature #	Specification
Yes	1	Develop a screen to display advising sheets submitted by CS department students.
Yes	2	Clicking on a student's name will redirect to a page displaying the student-submitted record. On this page, there will be options to approve or reject the record. When the admin submits their decision, they must also provide a text message with their feedback on the advising sheet. After clicking the submit button, the system will redirect to the advising sheet form (as described in point 1), and the new status of the student’s record will be updated accordingly.
Yes	3	Implement status update of student records upon submission of approval or rejection.
Yes	4	Upon submission, student will receive an email where they can see the status and message provided by admin.
Yes	5	New student will be able to see the updated status of their advising sheet on Course Advising History form.
Yes	6	Add reCAPTCHA to the login page. Verify the reCAPTCHA before login. Once the reCAPTCHA is verified then user can log in into the system.
Yes	7	Prevent your application from clickjacking attack. Implement the prevention of click jacking. Show the clickjacking prevention by using <iframe> in .html form.
Yes	8	Add a favicon to the website.
Yes	9	Add a password rule requiring a mix of capital letters, lowercase letters, special characters, and numbers (implement regex for all password fields in the application). The password length should be at least 8 characters.
Yes	10	Create test cases and execute in your BE application (Create at least 3 test cases).

3. Architecture

Like I mentioned in the overview, this project uses the Express framework and was written with a combination of Node, React, JavaScript, HTML, and CSS.

Below is a diagram that illustrates the layout of my website.



4. Implementation

1. Develop a screen to display advising sheets submitted by CS department students - (Completed)

- The front-end code is located at:
  - Client->src->components->AdminAdvisingRequests.jsx
- The back-end code is located at:
  - Server->routes->user_registration.js

2. Clicking on a student's name will redirect to a page displaying the student-submitted record. On this page, there will be options to approve or reject the record. When the admin submits their decision, they must also provide a text message with their feedback on the advising sheet. After clicking the submit button, the system will redirect to the advising sheet form (as described in point 1), and the new status of the student's record will be updated accordingly. - (Completed)

- The front-end code is located at:
  - Client->src->components->AdminAdvisingRequests.jsx
- The back-end code is located at:
  - Server->routes->user_registration.js

3. Implement status update of student records upon submission of approval or rejection. - (Completed)

- The front-end code is located at:
  - Client->src->components->AdminAdvisingRequests.jsx
- The back-end code is located at:
  - Server->routes->user_registration.js

4. Upon submission, student will receive an email where they can see the status and message provided by admin. – (Completed)

- The front-end code is located at:
  - Client->src->components->AdminAdvisingRequests.jsx
- The back-end code is located at:
  - Server->routes->user_registration.js

5. New student will be able to see the updated status of their advising sheet on Course Advising History form. - (Completed)

- The front-end code is located at:
  - Client->src->components->AdvisingHistory.jsx
- The back-end code is located at:
  - Server->routes->user_registration.js

6. Add reCAPTCHA to the login page. Verify the reCAPTCHA before login. Once the reCAPTCHA is verified then user can log in into the system. - (Completed)

- The front-end code is located at:
  - Client->src->components->Register.jsx
  - Client->src->components->Login.jsx
  - Client->src->components->ChangeInformation.jsx
  - Client->src->components->ForgotUserPassword.jsx

7. Prevent your application from clickjacking attack. Implement the prevention of click jacking. Show the clickjacking prevention by using <iframe> in .html form. - (Completed)

- When the user is prompted to provide prerequisites, they are given a dynamic amount of dropdown menus.
- The front-end code is located at:
  - Client->src->components->main.jsx

8. Add a favicon to the website. - (Completed)

- Each of the "major pages" of the website will dynamically change the favicon and the title of the tab.
- The front-end code is located at:
  - Client->src->components->About.jsx
  - Client->src->components->AdminDashboard.jsx
  - Client->src->components->Dashboard.jsx
  - Client->src->components->Home.jsx
  - Client->src->components->Login.jsx
  - Client->src->components->mysqlTesting.jsx
  - Client->src->components->Register.jsx
  - Client->src->components->TwoFactorAuthentication.jsx

9. Add a password rule requiring a mix of capital letters, lowercase letters, special characters, and numbers (implement regex for all password fields in the application). The password length should be at least 8 characters. - (Completed)

- The front-end code is located at:
  - Client->src->components->ChangeInformation.jsx
  - Client->src->components->Register.jsx

10. Create test cases and execute in your BE application (Create at least 3 test cases) - (Completed)

- The back-end code is located at:
  - Server->routes->test.js