1. Unit Tests

The unit tests are being done in the App.test.js file.

Tests are being done in the JEST framework.

Unit Test # 1:

* I am checking if the main animated gif is there or not

Unit Test # 2:

* I am checking if graphic2.svg exists or not

In unit tests 3,4, and 5, I am actually testing if the redux is working fine or not.

In unit test 5, I am checking if the constants are as per defined or not.

1. Making code functional
   1. Index.js (Line Number 77)

I took a code from the Line Number 77 and added into a function

* 1. Auth.js

I wrote the JSX code of the login and register forms and added into functions, to enhance the readability of the code.

1. Architecture Diagram of School management system

In this diagram, in the left box, there are the actors that will be using the system. They connect to the system via the Browser. They open the website on the browser. The browser requests the UI from the server, the server shows the frontend to the user by sending HTML/CSS/JSS files to the browser.

Now, in the right box, there are multiple web services there. Each service maybe deployed on different servers because it is a micro-services architecture. The functionality is being divided into different services. Each service is independent and does its own distinct functionality. The notification service then notifies the users if there is any kind of notification to send to the users. Example, the students are to be informed about upcoming exams.

Diagram

Description automatically generated

1. Simple CI/CD

Done on the GitHub

1. Design pattern

The design pattern in my code is the “Composite Design” pattern. In the composite design pattern, the programmer partitions any group of objects into sub-objects and treat them equally. Each sub-object then has its own functionality, or in our case, its own respective tasks to display.

We have used this pattern firstly in the Auth.js file, where we have divided the same page into two forms i.e., Login & Register form. We are treating both of the forms equally but their outcome or functionality is distinct.

Secondly, this pattern is being implemented on the main dashboard page, where we have divided our screens into various columns like completed, done, etc. Each column is treated same but their tasks are different.