Advanced Software Engineering and Industrial Practices

SFSU CSC Help

CSC 890 – Team 02

Prathiba Ramesh - Frontend Lead
Anudeep Katukojwala – Backend Lead

“Milestone 4”

November 21, 2022

Application URL:

http://20.120.20.37/

Connection Details:

Web Server	1. Private key to Download 2. chmod 400 <private key path> 3. ssh -i <private key path> csc890Team@20.120.20.37
SQL Server	1. Private key to Download 2. chmod 400 <private key path> 3. ssh -i <private key path> azureuser@20.228.204.41

Contents

1. Priority Features - Final Commitment

Name of the application: SFSU-CSC-Help

Function	Sub Function - Description	Priority
Account management	Users shall be able to register their accounts with the app.	1
	User shall be able to log into (with registered Cred) and log out of their accounts to gain access and deny access from the app.	1
	Application shall require users to verify their email address upon registration, password should be of 8-13 characters.	1
	User's personal data shall be hashed/encrypted from public access for additional security.	1
	User shall be able to add their role on registering to the page. Roles can be General User or administrator.	1
Deadline (or) Important Date Information	Users shall be able to ask any questions related to academic deadlines and priority dates, and get back appropriate response from AI	1
	User shall be able to ask the same question in any variations repeatedly and they should still get the correct desired answer.	1
	User shall be able to give Feedback on whether they are satisfied or not, based on the answer they received. This is optional.	1
Dashboard (or) History of Data	User should be able to visit the history of questions and answers that they have asked and received so far.	1
	User and Administrator should be able to filter this data based on keyword in the question, semester (Spring/Fall) and date-time.	1
	Administrator shall be able to view the entire list of users and all the questions they have asked so far, along with the answers they have received.	1
	Administrator shall be able to see the feedback provided by users for each question and answer they have received.	1

2. *Non - Functional Specifications:*

Below is the list of non-functional Specs that we mentioned earlier with their current status,

- (High priority) Application UI should be responsive so users can use it on their mobile or their laptop - **DONE**
- (High priority) User-friendly UI - **DONE**
- (High priority) Users' data must be protected - **ON TRACK**
- Database: Store the feedback information dynamically and display based on user role – **DONE**

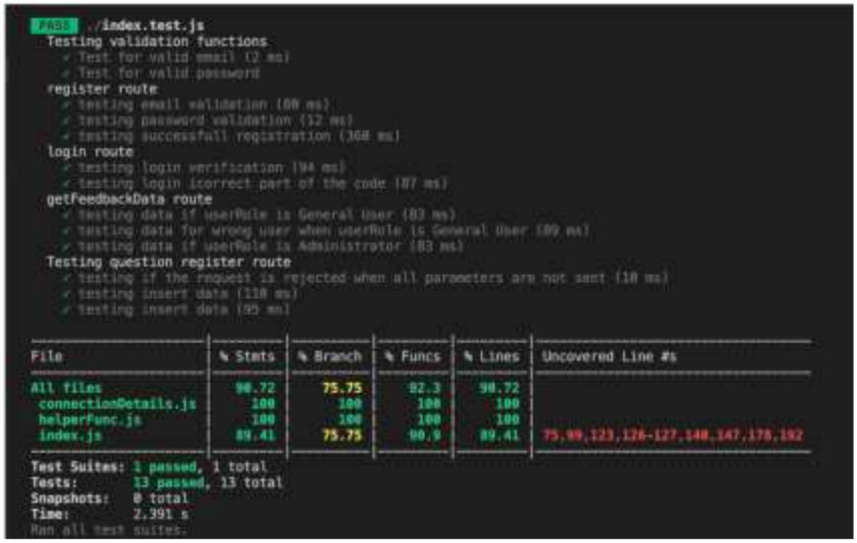
3. *QA Testing:*

a) Unit Test

We have used the Jest to write our unit test cases and for the testing purpose on the whole. In total 13 Unit test cases have been written for all major functionalities so far.

1) Tested the correctness of validation functions such as email validation and password validation
2) Tested post methods in register route including the successful registration function
3) Tested post methods in login route validating both the correct and incorrect login methods
4) Tested post methods in *getFeedbackData* route for both General and Administrator users
5) Tested post methods in *questionRegister* route with both correct and incorrect parameters

Total functional coverage of unit test cases is >90%
Below is the screenshot for test coverage and test result,



```
PASS  ./index.test.js
  Testing validation functions
    ✓ Test for valid email (2 ms)
    ✓ Test for valid password
  register route
    ✓ testing email validation (80 ms)
    ✓ testing password validation (12 ms)
    ✓ testing successfull registration (368 ms)
  login route
    ✓ testing login verification (94 ms)
    ✓ testing login icorrect part of the code (87 ms)
  getFeedbackData route
    ✓ testing data if userRole is General User (83 ms)
    ✓ testing data for wrong user when userRole is General User (89 ms)
    ✓ testing data if userRole is Administrator (83 ms)
  Testing question register route
    ✓ testing if the request is rejected when all parameters are not sent (10 ms)
    ✓ testing insert data (118 ms)
    ✓ testing insert data (95 ms)

------------------------|---------|----------|---------|---------|--------------------------------
File                    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------------------|---------|----------|---------|---------|--------------------------------
All files               |   90.72 |    75.75 |    92.3 |   90.72 |
 connectionDetails.js   |     100 |      100 |     100 |     100 |
 helperFunc.js          |     100 |      100 |     100 |     100 |
 index.js               |   89.41 |    75.75 |    90.9 |   89.41 | 75,99,123,126-127,140,147,178,192
------------------------|---------|----------|---------|---------|--------------------------------
Test Suites: 1 passed, 1 total
Tests:       13 passed, 13 total
Snapshots:   0 total
Time:        2.391 s
Ran all test suites.
```

In progress → We are implementing the CICD part, where the Unit test cases will automatically run on every GitHub Merge.

b) Integration Test

The template for integration testing was fetched from below link,

https://www.guru99.com/download-sample-test-case-template-with-explanation-of-importantfields.html

Please find below the attached excel sheet with Integration test cases and results,

https://docs.google.com/spreadsheets/d/1eTi4TPXyJyI5WHklVuYsDxBJ2WUB9L31aC3RdOAZ0NM

The below functionalities are covered as part of Integration testing.

User Register: Test the Register New User Functionality in SFSU CSC Help App (**PASS**)

- TC_001: Verify on entering valid input entries and user should be able to register
- TC_002: Verify on entering different password on confirm password field
- TC_003: Verify on entering incorrect email format
- TC_004: Verify on entering all mandatory fields

User Login: Test the Login/Logout Functionality in SFSU CSC Help App (**PASS**)

- TC_005: Verify on entering valid userid and password, the customer can login
- TC_006: Verify on entering invalid user id
- TC_007: Verify on entering correct user id, but wrong password.
- TC_008: Verify Logout operation

Ask Questions Function: Test the Help or Ask question to AI functionality in SFSU CSC Help (**PASS**)

- TC_009: Check the Ask help feature by entering any question regarding SFSU and with no feedback
- TC_010: Check the Ask help feature by entering any question regarding SFSU and with feedback

Dashboard Function: Test the Dashboard functionality in SFSU CSC Help (**FAIL**)

- TC_011: Check the dashboard functionality for admin users
- TC_012: Check the dashboard functionality for general users

Bug Fixing: *Assignee* → Prathiba Ramesh *Status* → In Progress *Github Url* → Bug Fixing

4. *Code Review*

a) Coding Style:

- The code is maintained in such a way that it is self-explanatory.
- The Naming convention in the code for functions and variables are well maintained as per the standard and is consistent throughout our project file.
- The code is properly indented for better understanding
- Each of the functions are well documented as to know what exactly it does.
- Comments are provided for each of the critical functions in the code.

b) Code Review:

Once the code has been developed by the frontend team, the backend team was assigned for the code review and vice versa.

Below are the Git Pull request URL's that we fetched from GitHub for each code review requests and approvals,

- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/11
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/12
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/16
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/19
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/6
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/5
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/4
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/14
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/13
- https://github.com/PrathibaRamesh/CSC890-03-Fa22-Team02/pull/18