Testing Document for Rate My Course

Team 102: Bowen Cheng, Phaneendra Amruthur Ravi, Venkata Sai Srikar Jilla, Yifan Feng, Yifan Wang, Wenjing Xu

1. Test Objectives

The purpose of the Testing Phase is to ensure that we deliver a quality product. We will be validating if we are building the right product and verify if we are building the product right against the requirements of the product. Rate My Course is built on modern web application frameworks using ExpressJS backend and RectJS for frontend. The application requires a mix of both Manual and Automation testing, Due to time constraints and considering the scope of the project we will be relying more on Manual testing compared to Automation testing.

2. Testing Approach

We will be conducting both Manual and Automated testing for the project. The tests included are:

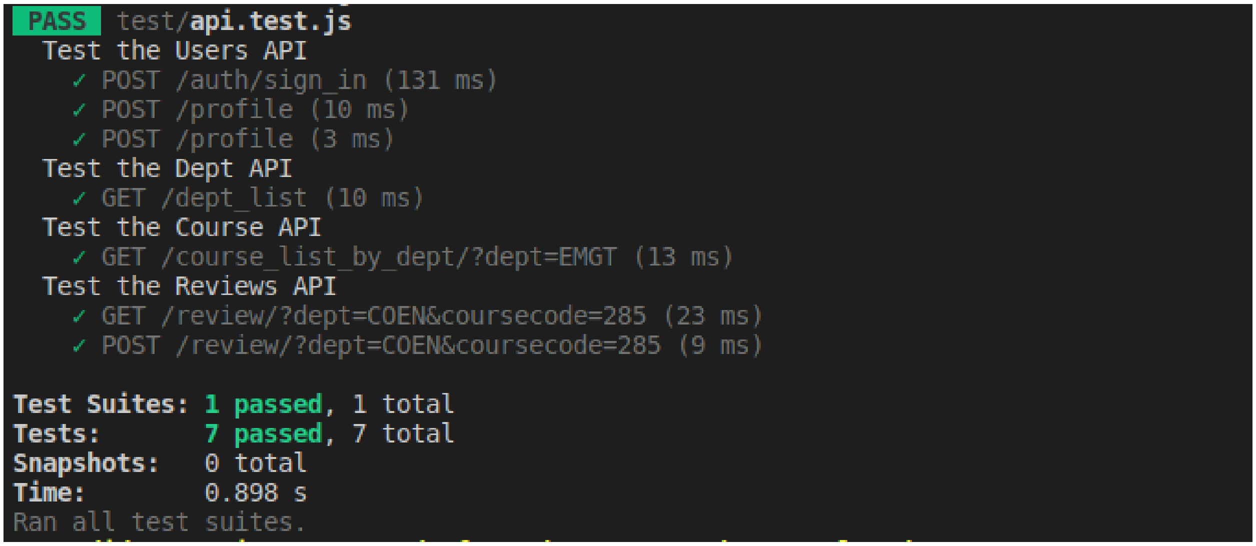
- Unit Testing of APIs
- End-to-end testing
- Cross-browser testing

- Unit Testing of APIs - The Unit test of the APIs have been automated with 85% code coverage.
- End-to-end testing - Manual testing after integration of both frontend and backend.
- Cross-browser testing - Manual Testing on different browsers to check compatibility

3. Unit Testing of APIs

The ExpressJS APIs are tested using the JEST test framework with moderate test coverage. The framework sends mock data for POST calls and expects results, and the GET calls send a predefined request and expects the result to validate the test cases.

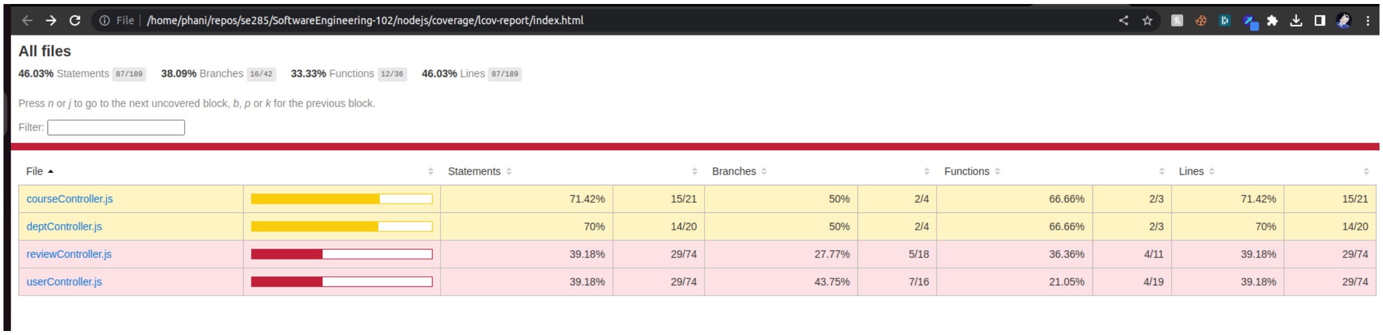
The APIs also have logging for more observability and monitoring.
The screenshot below shows the status of the test suit run and the results of the tests.



```
 PASS  test/api.test.js
  Test the Users API
    ✓ POST /auth/sign_in (131 ms)
    ✓ POST /profile (10 ms)
    ✓ POST /profile (3 ms)
  Test the Dept API
    ✓ GET /dept_list (10 ms)
  Test the Course API
    ✓ GET /course_list_by_dept/?dept=EMGT (13 ms)
  Test the Reviews API
    ✓ GET /review/?dept=COEN&coursecode=285 (23 ms)
    ✓ POST /review/?dept=COEN&coursecode=285 (9 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        0.898 s
Ran all test suites.
```

Below is the code coverage report generated by the JEST framework, Due to the limited time we couldn't cover most of the code, which we are planning to cover as a future enhancement.



File
/home/phani/repos/se285/SoftwareEngineering-102/nodejs/coverage/lcov-report/index.html

All files

46.03% Statements 87/189 **38.09%** Branches 16/42 **33.33%** Functions 12/36 **46.03%** Lines 87/189

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲		Statements		Branches		Functions		Lines	
courseController.js		71.42%	15/21	50%	2/4	66.66%	2/3	71.42%	15/21
deptController.js		70%	14/20	50%	2/4	66.66%	2/3	70%	14/20
reviewController.js		39.18%	29/74	27.77%	5/18	36.36%	4/11	39.18%	29/74
userController.js		39.18%	29/74	43.75%	7/16	21.05%	4/19	39.18%	29/74

4. End-to-End Testing

After the integration of the ExpressJS backend with frontend ReactJS, we were able to write a few test cases to verify and validate our software end to end from the client side, that is from the browser.

4.1. Testing user interaction with Sign In/Up

Test case number	Test case	Expected Behavior	Status
1	Page load on first load	Loads Sign In / Register page	Passed
2	Sign in with incorrect	Loads message that	Passed

	credentials	credentials failed	
3	Sign In with correct credentials	Loads filter page of the app	Passed
4	Sign Up with a different email other than SCU	Loads message that user should try with SCU email only	Passed
5	Sign up with password whose length is lesser than 8 characters	Loads message that user should try a stronger password which is at least 8 characters long	Passed
6	Sign up with credentials which meets criteria of test case 3 and 4	Loads message that account is created and user needs to check email to validate email	Passed
7	Sign up with credentials which meets criteria of test case 3 and 4 but account already exists	Loads message to try to sign in instead of sign up	Passed
8	User checks email and clicks link to verify email	Loads message that account is verified and user can now try signing in with credentials	Passed
9	Load page on the browser after closing it once after sign in	Session is maintained after login and loads filter page	Passed

4.2. Testing user interaction with Filter Page

Test case number	Test Case	Expected Behavior	Status
1	Click on logout button	Logs user out and redirects to sign in/up page	Passed
2	Select a department	Loads the course list	Passed

	from the list	based on the selected department	
3	Select a course from the list	Selects the course and search button appears	Passed
4	Click the search button	Loads the reviews page based on department and course selected	Passed

4.3. Testing user interaction with Reviews Page

Test case number	Test Case	Expected Behavior	Status
1	On Page load of Reviews page	List of all reviews of all instructors	Passed
2	Click Add Review	Loads add review page	Passed
3	Click Download Syllabus	Downloads the syllabus copy	Passed
4	Load of Add Review page	Loads page form and populates form if a review by that user already exists	Passed
5	Click submit review button	Saves the user’s review and loads review list page	Passed

5. Cross-browser Compatibility testing

To ensure that functionality of the UI is compatible for all clients we tested our application on all major modern browsers and ensured all functions work with no issues. Below were the browsers that were tested on:

- Chrome - Safari - Firefox
- Edge - Opera