

IT314: Software Engineering

Unit Testing
(Group 16 - Common Ground)

Prepared & Done by: Sabva Jay Dilipbhai - 202101224

Course instructor: Prof. Saurabh Tiwari



November 27, 2023

1 Unit Testing - Brief

We used **Jest** Unit testing framework to make unit tests and also added some libraries like **Supertest**, **Mocha**, and **Chai** To Test the NodeJs Server of our Project.

- a) Jest is a powerful and widely-used JavaScript testing framework, as our primary tool for unit testing. Jest provides a comprehensive set of features that make unit testing in our Node.js application efficient and effective.

Jest Features: Zero-configuration setup, fast parallel execution, built-in mocking, code coverage analysis

- b) Supertest Features: Efficient HTTP assertion testing, allowing simulation of HTTP requests, and assertion of expected behaviors, enhancing unit and integration testing for Node.js applications.
- c) Mocha Features: Flexible and feature-rich test framework supporting asynchronous testing, well-defined test lifecycle, and seamless integration with assertion libraries like Chai, providing a robust testing environment.
- d) Chai Features: Expressive assertion library seamlessly integrated with test frameworks like Mocha, enhancing readability and providing multiple assertion styles for clear and concise unit and integration tests.

2 Links

- [Code Coverage Report](#)
- [Source Code Of Unit Test](#)

3 Code Coverage Report

We used nyc instanbul library to generate the code coverage report.

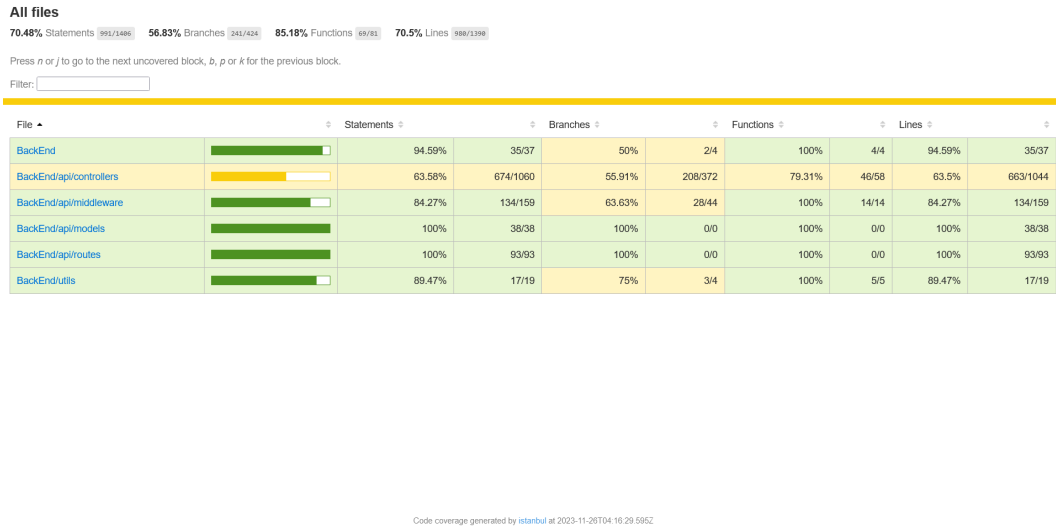


Figure 1: Code Coverage for Backend Folder Structure

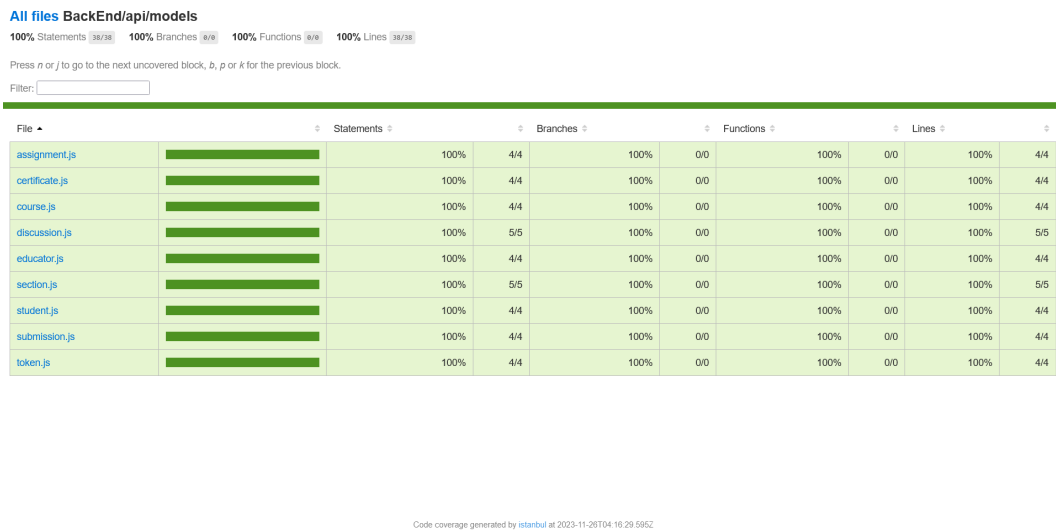


Figure 2: Code Coverage for Models

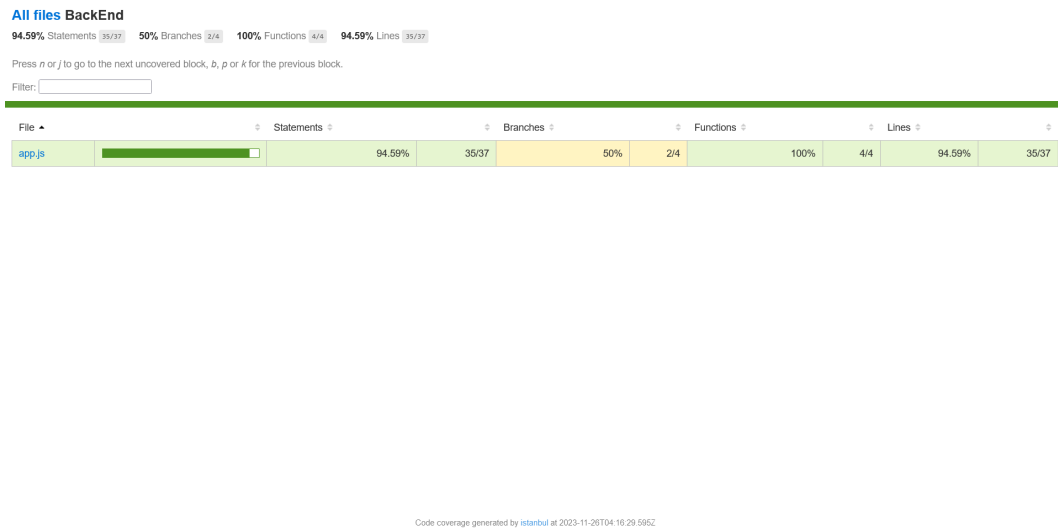


Figure 3: Code Coverage for App.js Configuration

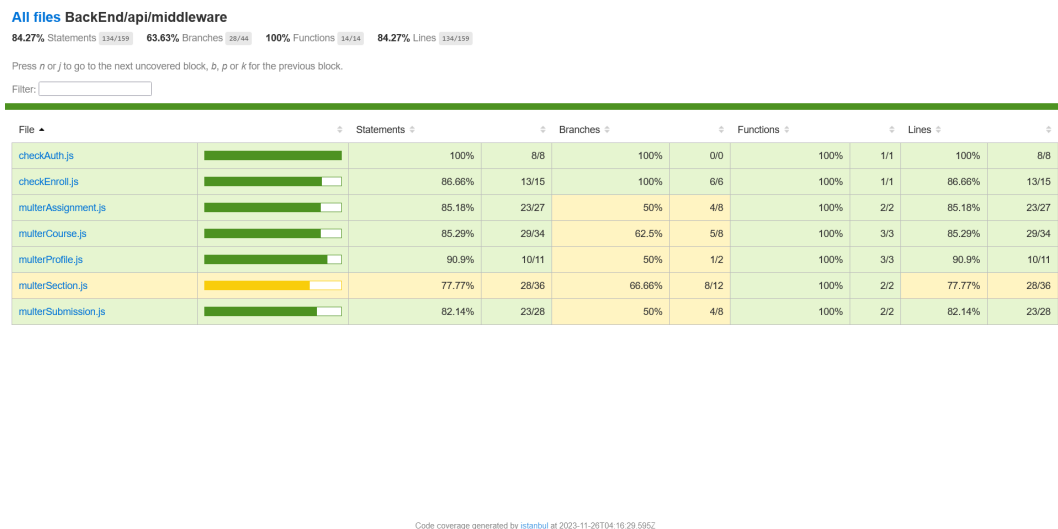


Figure 4: Code Coverage for Middleware Implementation (Multer)

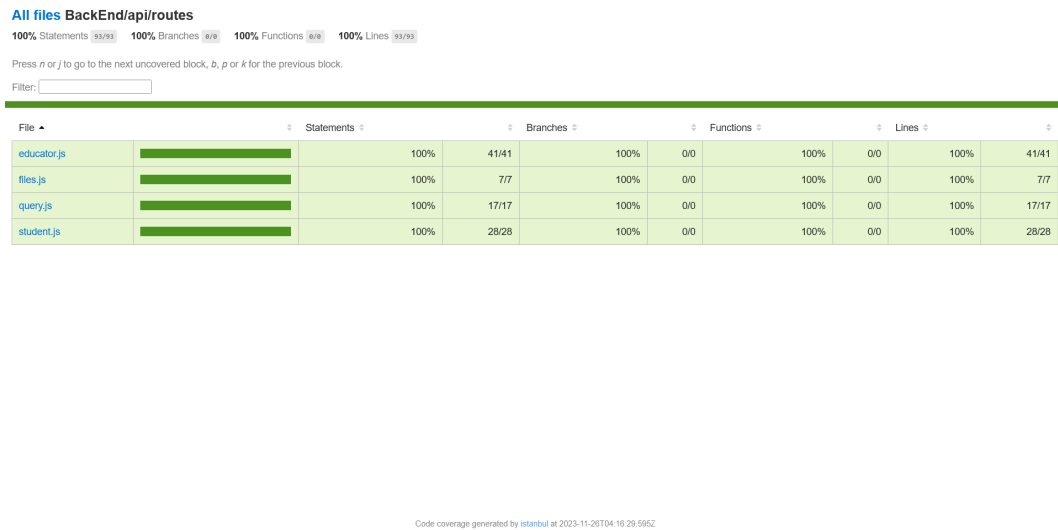


Figure 5: Code Coverage for Route Configuration

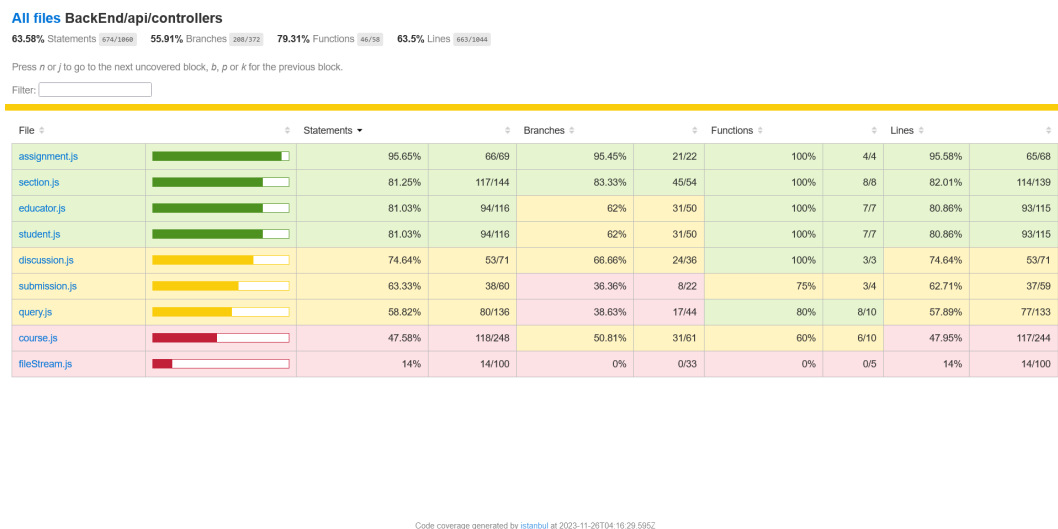


Figure 6: Code Coverage for Controller Logic

All files / BackEnd/api/controllers assignment.js

95.65% Statements 66/69 95.45% Branches 21/22 100% Functions 4/4 95.58% Lines 65/68

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

```

1 |x| const mongoose = require('mongoose');
2 |x| const fs = require('fs');
3 |x| const path = require('path');
4
5 |x| const Course = require('../models/course');
6 |x| const Educator = require('../models/educator');
7 |x| const Student = require('../models/student');
8 |x| const Assignment = require('../models/assignment');
9 |x| const Submission = require('../models/submission');
10
11 |x| const sendEmail = require('../utils/sendEmail');
12 |x| const (deleteFile, deleteFolder) = require('../utils/deleteFile');
13
14
15 |x| exports.createAssignment = async (req, res, next) => {
16 |x|   try {
17 |x|     if (req.userData.userId !== 'educator') {
18 |x|       return res.status(401).json({
19 |x|         message: 'Unauthorized'
20 |x|       });
21 |x|     }
22
23 |x|     const course = await Course.findById(req.params.courseId).exec();
24 |x|     if (!course) {
25 |x|       return res.status(404).json({
26 |x|         message: 'Course not found'
27 |x|       });
28 |x|     }
29 |x|     const educator = await Educator.findById(req.userData.userId).exec();
30
31 |x|     if (educator_id.toString() !== course.createdBy.toString()) {
32 |x|       return res.status(401).json({
33 |x|         message: 'Unauthorized'
34 |x|       });
35 |x|     }
36
37 |x|     const attachment = req.files.map(file => file.path);
38 |x|     const assignment = new Assignment({

```

Figure 7: Code Coverage Line By Line

Windows PowerShell

at Object.equal (test/alltest.test.js:2919:36)

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	70.48	56.83	85.18	70.5	
Backend	94.59	80	100	94.59	
app.js	94.59	80	100	94.59	34-35
Backend/api/controllers	63.58	55.91	79.31	63.5	
assignment.js	95.05	95.45	100	95.05	50,143,188
course.js	47.58	50.81	60	47.95	...-298,313,336-337,344-365,392-393,410,431,438-566
discussion.js	74.64	66.66	100	74.64	...-67,73,91,98,111-112,123,129,140,147,154,167-168
educator.js	81.01	62	100	80.86	...-119-128,130,136-137,168-169,188-189,278,299-310
filestream.js	14	0	0	14	15-81,88-187,192-213
query.js	58.82	38.63	88	57.89	...-235,260-261,272,368-369,434,443-444,456,472-473
section.js	81.25	83.33	100	82.81	...-184,134,145-146,198-191,231-238,262-264,318-315
student.js	81.03	62	100	80.85	...-120-121,131,137-138,162-166,189-190,280,301-312
submission.js	63.33	36.36	75	62.71	35,41,65-66,84-124
Backend/api/middleware	84.27	63.63	100	84.27	
checkAuth.js	100	100	100	100	
checkEnroll.js	86.66	100	100	86.66	25-26
multiAssignment.js	85.18	50	100	85.18	12,17,33-34
multiCourse.js	85.29	62.5	100	85.29	32-33,39,46-47
multiProfile.js	98.9	50	100	98.9	8
multiSection.js	77.77	66.66	100	77.77	14,19,46-55
multiSubmission.js	82.14	50	100	82.14	14,19,24,34-35
Backend/api/models	100	100	100	100	
assignment.js	100	100	100	100	
certificate.js	100	100	100	100	
course.js	100	100	100	100	
discussion.js	100	100	100	100	
educator.js	100	100	100	100	
section.js	100	100	100	100	
student.js	100	100	100	100	
submission.js	100	100	100	100	
Backend/api/routes	100	100	100	100	
educator.js	100	100	100	100	
file.js	100	100	100	100	
query.js	100	100	100	100	
student.js	100	100	100	100	
Backend/utils	89.47	75	100	89.47	
deleteFile.js	98	75	100	98	14
sendEmail.js	88.88	100	100	88.88	26

Test Suites: 1 failed, 1 total
Tests: 5 failed, 122 passed, 127 total
Snapshots: 0 total
Time: 30.891 s
Run all test suites matching /\test\/alltest.test.js/.

Figure 8: Jest Test Coverage Execution in Terminal