IT314 – Software Engineering

G27 Flight Booking System

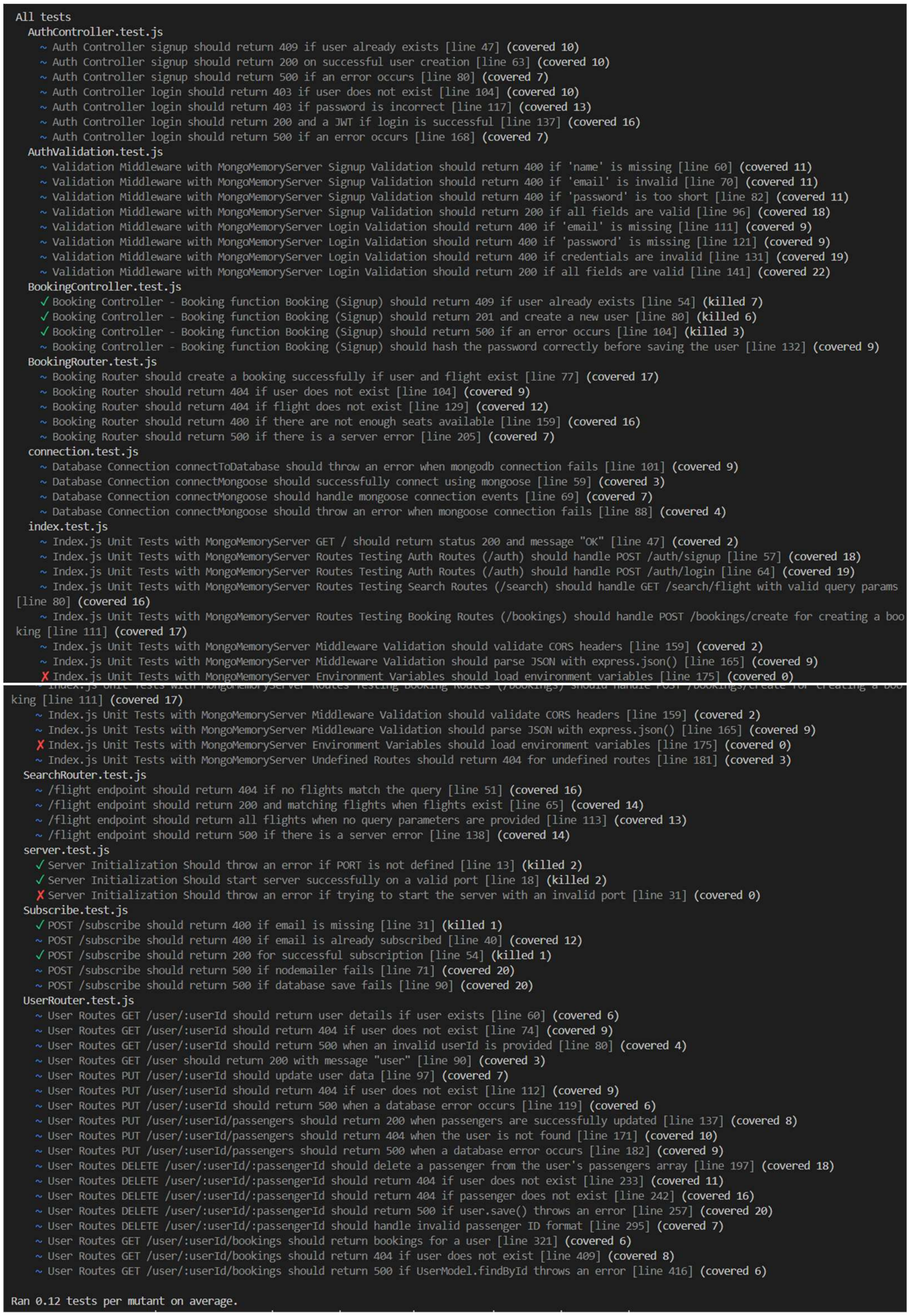
Mutation Testing Report

Document Purpose: This document covers the Mutation testing done on our code using the unit tests. We have achieved a mutation score of 100% in the test covering all our backend files.

Tools used:

Stryker - @stryker-mutator/core: ^8.6.0

The following are the tests ran on the code.



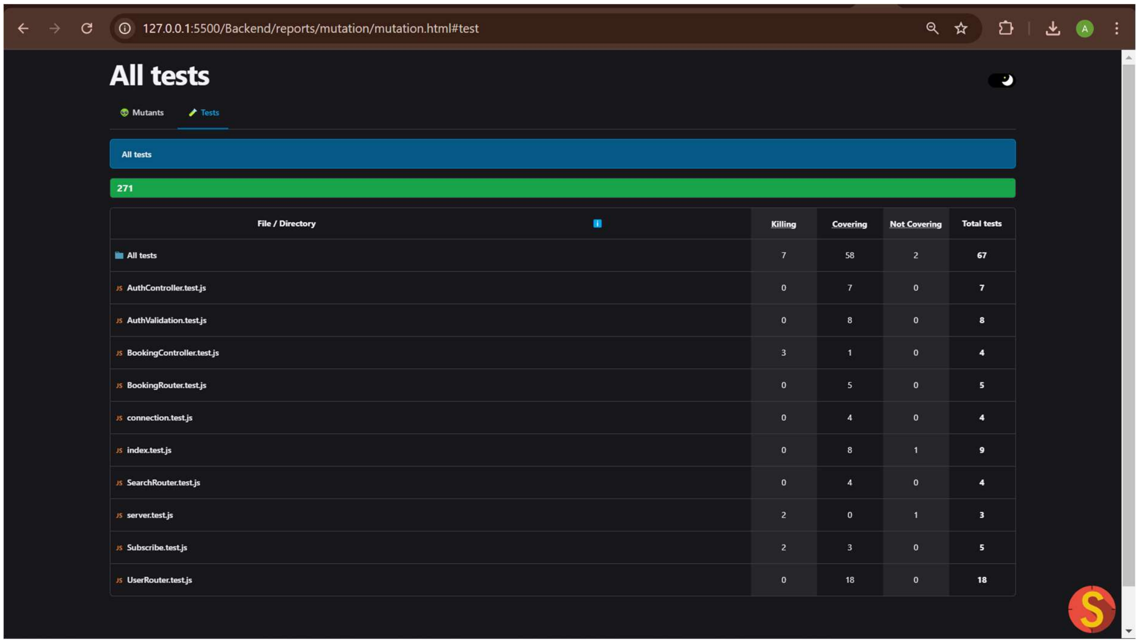
```
All tests
  AuthController.test.js
    ~ Auth Controller signup should return 409 if user already exists [line 47] (covered 10)
    ~ Auth Controller signup should return 200 on successful user creation [line 63] (covered 10)
    ~ Auth Controller signup should return 500 if an error occurs [line 80] (covered 7)
    ~ Auth Controller login should return 403 if user does not exist [line 104] (covered 10)
    ~ Auth Controller login should return 403 if password is incorrect [line 117] (covered 13)
    ~ Auth Controller login should return 200 and a JWT if login is successful [line 137] (covered 16)
    ~ Auth Controller login should return 500 if an error occurs [line 168] (covered 7)
  AuthValidation.test.js
    ~ Validation Middleware with MongoMemoryServer Signup Validation should return 400 if 'name' is missing [line 60] (covered 11)
    ~ Validation Middleware with MongoMemoryServer Signup Validation should return 400 if 'email' is invalid [line 70] (covered 11)
    ~ Validation Middleware with MongoMemoryServer Signup Validation should return 400 if 'password' is too short [line 82] (covered 11)
    ~ Validation Middleware with MongoMemoryServer Signup Validation should return 200 if all fields are valid [line 96] (covered 18)
    ~ Validation Middleware with MongoMemoryServer Login Validation should return 400 if 'email' is missing [line 111] (covered 9)
    ~ Validation Middleware with MongoMemoryServer Login Validation should return 400 if 'password' is missing [line 121] (covered 9)
    ~ Validation Middleware with MongoMemoryServer Login Validation should return 400 if credentials are invalid [line 131] (covered 19)
    ~ Validation Middleware with MongoMemoryServer Login Validation should return 200 if all fields are valid [line 141] (covered 22)
  BookingController.test.js
    ✓ Booking Controller - Booking function Booking (Signup) should return 409 if user already exists [line 54] (killed 7)
    ✓ Booking Controller - Booking function Booking (Signup) should return 201 and create a new user [line 80] (killed 6)
    ✓ Booking Controller - Booking function Booking (Signup) should return 500 if an error occurs [line 104] (killed 3)
    ~ Booking Controller - Booking function Booking (Signup) should hash the password correctly before saving the user [line 132] (covered 9)
  BookingRouter.test.js
    ~ Booking Router should create a booking successfully if user and flight exist [line 77] (covered 17)
    ~ Booking Router should return 404 if user does not exist [line 104] (covered 9)
    ~ Booking Router should return 404 if flight does not exist [line 129] (covered 12)
    ~ Booking Router should return 400 if there are not enough seats available [line 159] (covered 16)
    ~ Booking Router should return 500 if there is a server error [line 205] (covered 7)
  connection.test.js
    ~ Database Connection connectToDatabase should throw an error when mongodb connection fails [line 101] (covered 9)
    ~ Database Connection connectMongoose should successfully connect using mongoose [line 59] (covered 3)
    ~ Database Connection connectMongoose should handle mongoose connection events [line 69] (covered 7)
    ~ Database Connection connectMongoose should throw an error when mongoose connection fails [line 88] (covered 4)
  index.test.js
    ~ Index.js Unit Tests with MongoMemoryServer GET / should return status 200 and message "OK" [line 47] (covered 2)
    ~ Index.js Unit Tests with MongoMemoryServer Routes Testing Auth Routes (/auth) should handle POST /auth/signup [line 57] (covered 18)
    ~ Index.js Unit Tests with MongoMemoryServer Routes Testing Auth Routes (/auth) should handle POST /auth/login [line 64] (covered 19)
    ~ Index.js Unit Tests with MongoMemoryServer Routes Testing Search Routes (/search) should handle GET /search/flight with valid query params
[line 80] (covered 16)
    ~ Index.js Unit Tests with MongoMemoryServer Routes Testing Booking Routes (/bookings) should handle POST /bookings/create for creating a boo
king [line 111] (covered 17)
    ~ Index.js Unit Tests with MongoMemoryServer Middleware Validation should validate CORS headers [line 159] (covered 2)
    ~ Index.js Unit Tests with MongoMemoryServer Middleware Validation should parse JSON with express.json() [line 165] (covered 9)
    ✗ Index.js Unit Tests with MongoMemoryServer Environment Variables should load environment variables [line 175] (covered 0)
```

```
king [line 111] (covered 17)
    ~ Index.js Unit Tests with MongoMemoryServer Middleware Validation should validate CORS headers [line 159] (covered 2)
    ~ Index.js Unit Tests with MongoMemoryServer Middleware Validation should parse JSON with express.json() [line 165] (covered 9)
    ✗ Index.js Unit Tests with MongoMemoryServer Environment Variables should load environment variables [line 175] (covered 0)
```

```
    ~ Index.js Unit Tests with MongoMemoryServer Undefined Routes should return 404 for undefined routes [line 181] (covered 3)
  SearchRouter.test.js
    ~ /flight endpoint should return 404 if no flights match the query [line 51] (covered 16)
    ~ /flight endpoint should return 200 and matching flights when flights exist [line 65] (covered 14)
    ~ /flight endpoint should return all flights when no query parameters are provided [line 113] (covered 13)
    ~ /flight endpoint should return 500 if there is a server error [line 138] (covered 14)
  server.test.js
    ✓ Server Initialization Should throw an error if PORT is not defined [line 13] (killed 2)
    ✓ Server Initialization Should start server successfully on a valid port [line 18] (killed 2)
    ✗ Server Initialization Should throw an error if trying to start the server with an invalid port [line 31] (covered 0)
  Subscribe.test.js
    ✓ POST /subscribe should return 400 if email is missing [line 31] (killed 1)
    ~ POST /subscribe should return 400 if email is already subscribed [line 40] (covered 12)
    ✓ POST /subscribe should return 200 for successful subscription [line 54] (killed 1)
    ~ POST /subscribe should return 500 if nodemailer fails [line 71] (covered 20)
    ~ POST /subscribe should return 500 if database save fails [line 90] (covered 20)
  UserRouter.test.js
    ~ User Routes GET /user/:userId should return user details if user exists [line 60] (covered 6)
    ~ User Routes GET /user/:userId should return 404 if user does not exist [line 74] (covered 9)
    ~ User Routes GET /user/:userId should return 500 when an invalid userId is provided [line 80] (covered 4)
    ~ User Routes GET /user should return 200 with message "user" [line 90] (covered 3)
    ~ User Routes PUT /user/:userId should update user data [line 97] (covered 7)
    ~ User Routes PUT /user/:userId should return 404 if user does not exist [line 112] (covered 9)
    ~ User Routes PUT /user/:userId should return 500 when a database error occurs [line 119] (covered 6)
    ~ User Routes PUT /user/:userId/passengers should return 200 when passengers are successfully updated [line 137] (covered 8)
    ~ User Routes PUT /user/:userId/passengers should return 404 when the user is not found [line 171] (covered 10)
    ~ User Routes PUT /user/:userId/passengers should return 500 when a database error occurs [line 182] (covered 9)
    ~ User Routes DELETE /user/:userId/:passengerId should delete a passenger from the user's passengers array [line 197] (covered 18)
    ~ User Routes DELETE /user/:userId/:passengerId should return 404 if user does not exist [line 233] (covered 11)
    ~ User Routes DELETE /user/:userId/:passengerId should return 404 if passenger does not exist [line 242] (covered 16)
    ~ User Routes DELETE /user/:userId/:passengerId should return 500 if user.save() throws an error [line 257] (covered 20)
    ~ User Routes DELETE /user/:userId/:passengerId should handle invalid passenger ID format [line 295] (covered 7)
    ~ User Routes GET /user/:userId/bookings should return bookings for a user [line 321] (covered 6)
    ~ User Routes GET /user/:userId/bookings should return 404 if user does not exist [line 409] (covered 8)
    ~ User Routes GET /user/:userId/bookings should return 500 if UserModel.findById throws an error [line 416] (covered 6)

Ran 0.12 tests per mutant on average.
```

Terminal



All tests

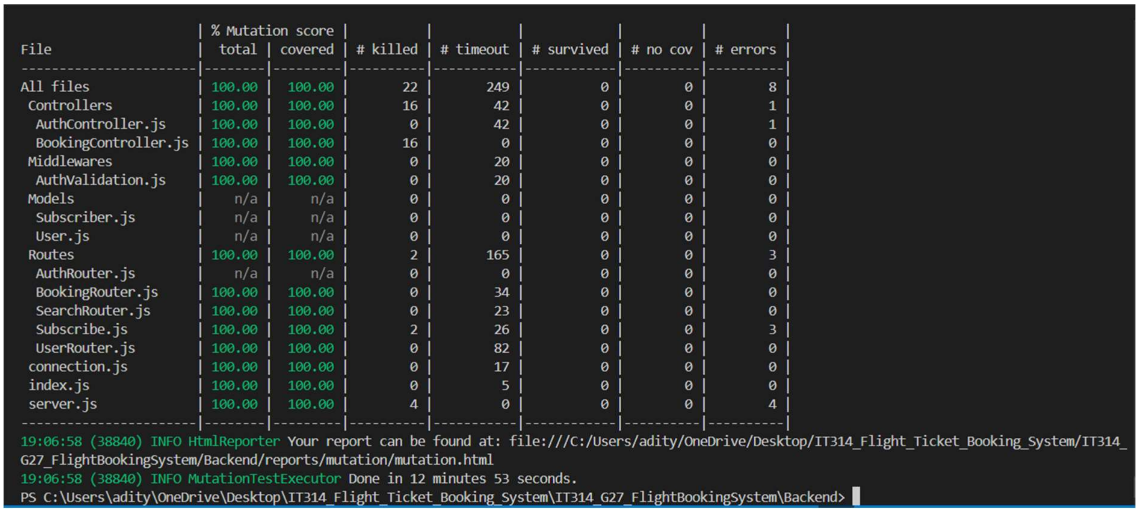
All tests

271

File / Directory	Killing	Covering	Not Covering	Total tests
All tests	7	58	2	67
AuthController.test.js	0	7	0	7
AuthValidation.test.js	0	8	0	8
BookingController.test.js	3	1	0	4
BookingRouter.test.js	0	5	0	5
connection.test.js	0	4	0	4
index.test.js	0	8	1	9
SearchRouter.test.js	0	4	0	4
server.test.js	2	0	1	3
Subscribe.test.js	2	3	0	5
UserRouter.test.js	0	18	0	18

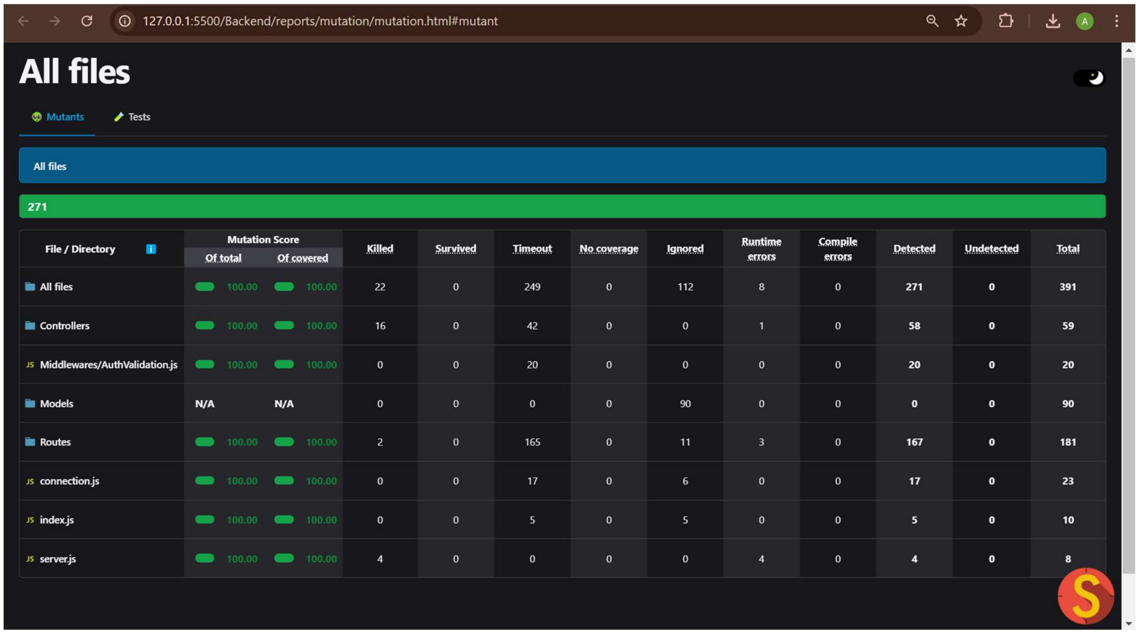
HTML File

The following are the Screenshots of the Mutation Scores for all the Backend files.



```
                       | % Mutation score |          |           |            |          |          |
File                   |  total | covered | # killed | # timeout | # survived | # no cov | # errors |
-----------------------|--------|---------|----------|-----------|------------|----------|----------|
All files              | 100.00 |  100.00 |       22 |       249 |          0 |        0 |        8 |
 Controllers           | 100.00 |  100.00 |       16 |        42 |          0 |        0 |        1 |
  AuthController.js    | 100.00 |  100.00 |        0 |        42 |          0 |        0 |        1 |
  BookingController.js | 100.00 |  100.00 |       16 |         0 |          0 |        0 |        0 |
 Middlewares           | 100.00 |  100.00 |        0 |        20 |          0 |        0 |        0 |
  AuthValidation.js    | 100.00 |  100.00 |        0 |        20 |          0 |        0 |        0 |
 Models                |    n/a |     n/a |        0 |         0 |          0 |        0 |        0 |
  Subscriber.js        |    n/a |     n/a |        0 |         0 |          0 |        0 |        0 |
  User.js              |    n/a |     n/a |        0 |         0 |          0 |        0 |        0 |
 Routes                | 100.00 |  100.00 |        2 |       165 |          0 |        0 |        3 |
  AuthRouter.js        |    n/a |     n/a |        0 |         0 |          0 |        0 |        0 |
  BookingRouter.js     | 100.00 |  100.00 |        0 |        34 |          0 |        0 |        0 |
  SearchRouter.js      | 100.00 |  100.00 |        0 |        23 |          0 |        0 |        0 |
  Subscribe.js         | 100.00 |  100.00 |        2 |        26 |          0 |        0 |        3 |
  UserRouter.js        | 100.00 |  100.00 |        0 |        82 |          0 |        0 |        0 |
 connection.js         | 100.00 |  100.00 |        0 |        17 |          0 |        0 |        0 |
 index.js              | 100.00 |  100.00 |        0 |         5 |          0 |        0 |        0 |
 server.js             | 100.00 |  100.00 |        4 |         0 |          0 |        0 |        4 |
-----------------------|--------|---------|----------|-----------|------------|----------|----------|
19:06:58 (38840) INFO HtmlReporter Your report can be found at: file:///C:/Users/adity/OneDrive/Desktop/IT314_Flight_Ticket_Booking_System/IT314_
G27_FlightBookingSystem/Backend/reports/mutation/mutation.html
19:06:58 (38840) INFO MutationTestExecutor Done in 12 minutes 53 seconds.
PS C:\Users\adity\OneDrive\Desktop\IT314_Flight_Ticket_Booking_System\IT314_G27_FlightBookingSystem\Backend>
```

Terminal



127.0.0.1:5500/Backend/reports/mutation/mutation.html#mutant

All files

All files

271

File / Directory	Mutation Score		Killed	Survived	Timeout	No coverage	Ignored	Runtime errors	Compile errors	Detected	Undetected	Total
	Of total	Of covered										
All files	100.00	100.00	22	0	249	0	112	8	0	**271**	**0**	**391**
Controllers	100.00	100.00	16	0	42	0	0	1	0	**58**	**0**	**59**
JS Middlewares/AuthValidation.js	100.00	100.00	0	0	20	0	0	0	0	**20**	**0**	**20**
Models	N/A	N/A	0	0	0	0	90	0	0	**0**	**0**	**90**
Routes	100.00	100.00	2	0	165	0	11	3	0	**167**	**0**	**181**
JS connection.js	100.00	100.00	0	0	17	0	6	0	0	**17**	**0**	**23**
JS index.js	100.00	100.00	0	0	5	0	5	0	0	**5**	**0**	**10**
JS server.js	100.00	100.00	4	0	0	0	0	4	0	**4**	**0**	**8**

HTML File