INTRODUCTION

Mutation testing checks how strong our test cases are by introducing small changes (“mutants”) into the code. If tests fail correctly, the mutant is killed. If tests still pass, the mutant survives, indicating missing or weak test coverage.

OBJECTIVE

The main goals of the unit testing process were:

To verify that each function behaves correctly and returns expected results.

To validate all logic branches, including success paths, edge cases, and error conditions.

To achieve high code coverage and ensure the controller logic is reliable and maintainable.

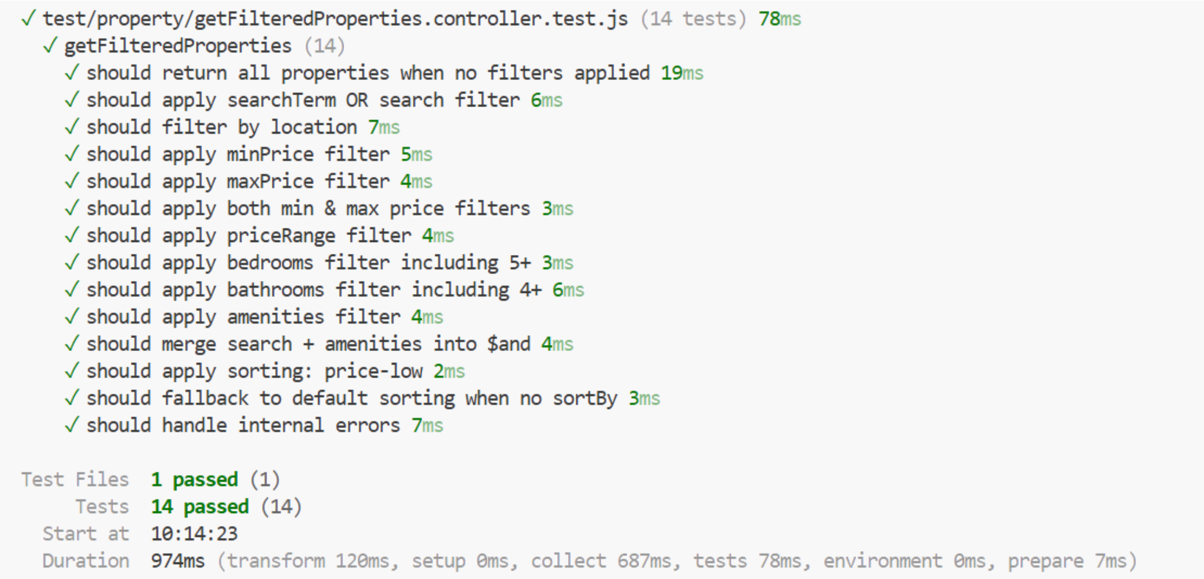
UNIT TESTING FRAMEWORK

Component	Details
Language	JavaScript
Framework	Vitest
Mocking Library	Vitest’s native mocking capabilities
LLM Used	DeepSeek, Chatgpt, Copilot

TEST FILES:

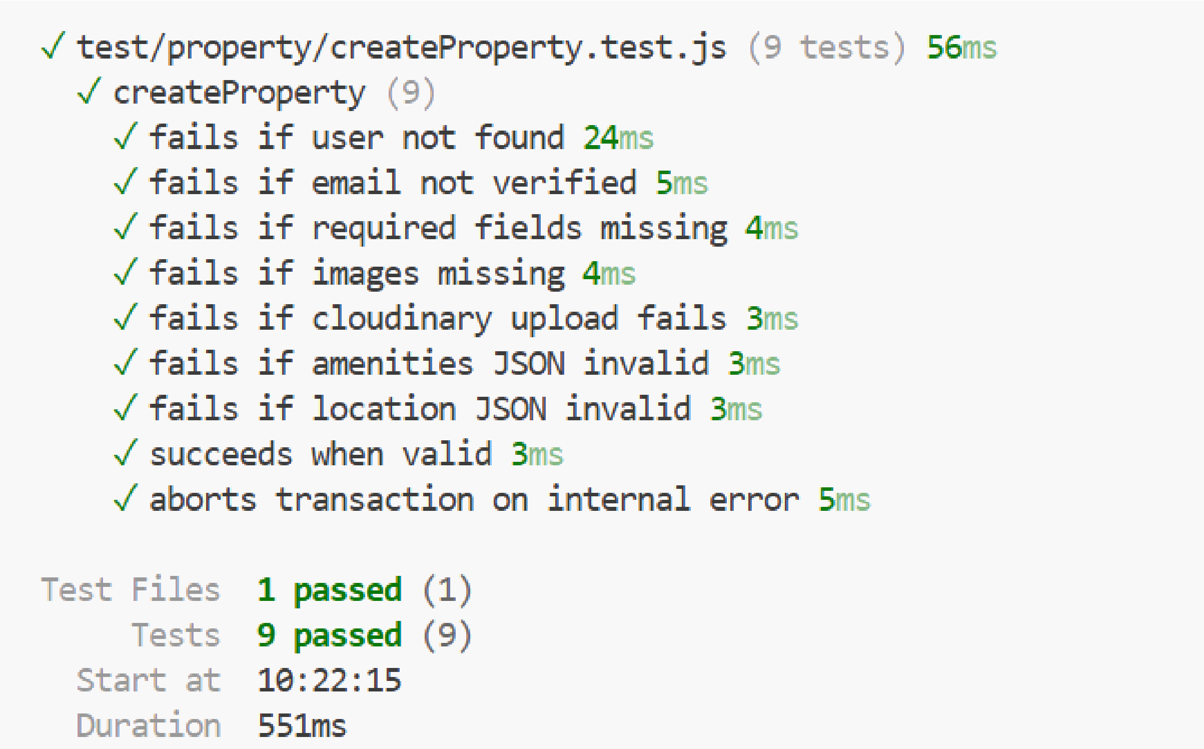
1. property.controller.js

getFilteredProperties



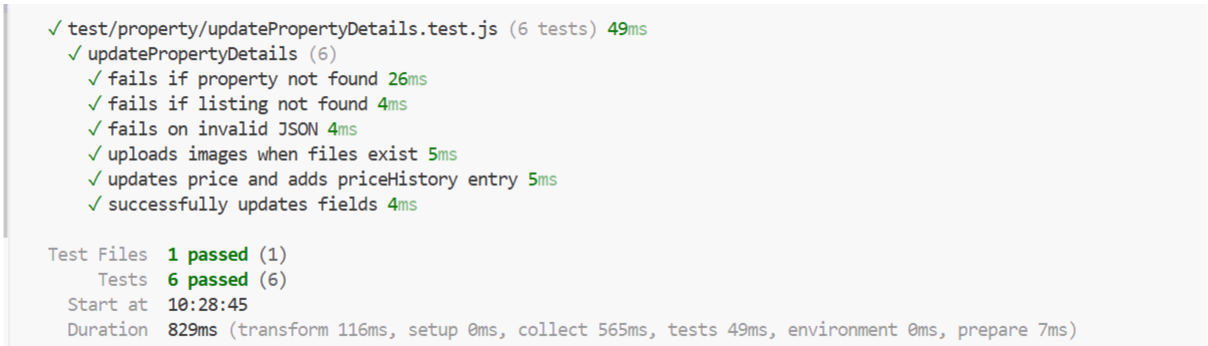
```
✓ test/property/getFilteredProperties.controller.test.js (14 tests) 78ms
  ✓ getFilteredProperties (14)
    ✓ should return all properties when no filters applied 19ms
    ✓ should apply searchTerm OR search filter 6ms
    ✓ should filter by location 7ms
    ✓ should apply minPrice filter 5ms
    ✓ should apply maxPrice filter 4ms
    ✓ should apply both min & max price filters 3ms
    ✓ should apply priceRange filter 4ms
    ✓ should apply bedrooms filter including 5+ 3ms
    ✓ should apply bathrooms filter including 4+ 6ms
    ✓ should apply amenities filter 4ms
    ✓ should merge search + amenities into $and 4ms
    ✓ should apply sorting: price-low 2ms
    ✓ should fallback to default sorting when no sortBy 3ms
    ✓ should handle internal errors 7ms

 Test Files  1 passed (1)
      Tests  14 passed (14)
   Start at  10:14:23
   Duration  974ms (transform 120ms, setup 0ms, collect 687ms, tests 78ms, environment 0ms, prepare 7ms)
```

createProperty



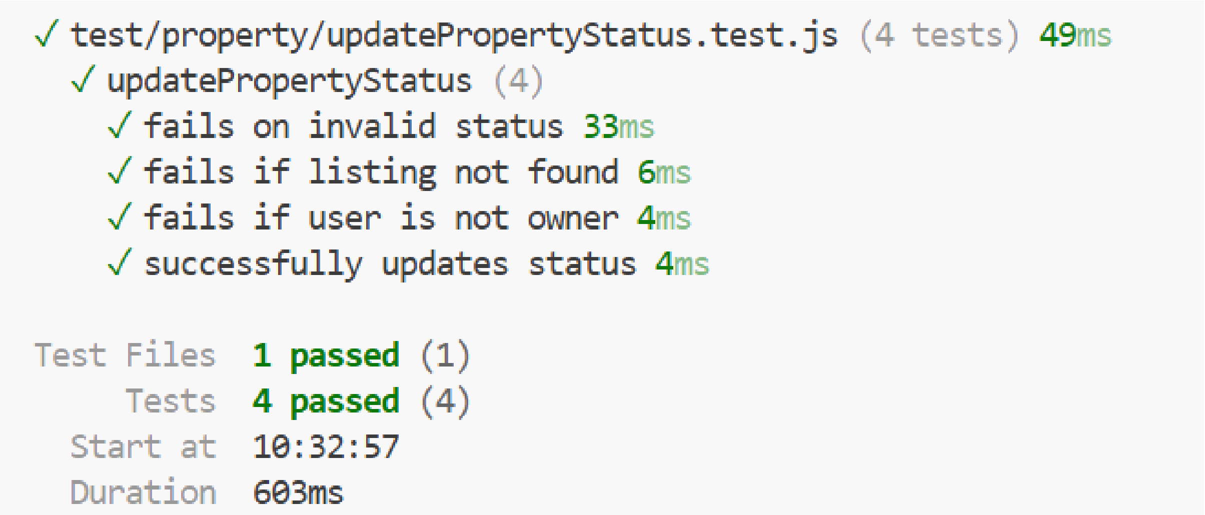
```
✓ test/property/createProperty.test.js (9 tests) 56ms
  ✓ createProperty (9)
    ✓ fails if user not found 24ms
    ✓ fails if email not verified 5ms
    ✓ fails if required fields missing 4ms
    ✓ fails if images missing 4ms
    ✓ fails if cloudinary upload fails 3ms
    ✓ fails if amenities JSON invalid 3ms
    ✓ fails if location JSON invalid 3ms
    ✓ succeeds when valid 3ms
    ✓ aborts transaction on internal error 5ms

 Test Files  1 passed (1)
      Tests  9 passed (9)
   Start at  10:22:15
   Duration  551ms
```

updatePropertyDetails



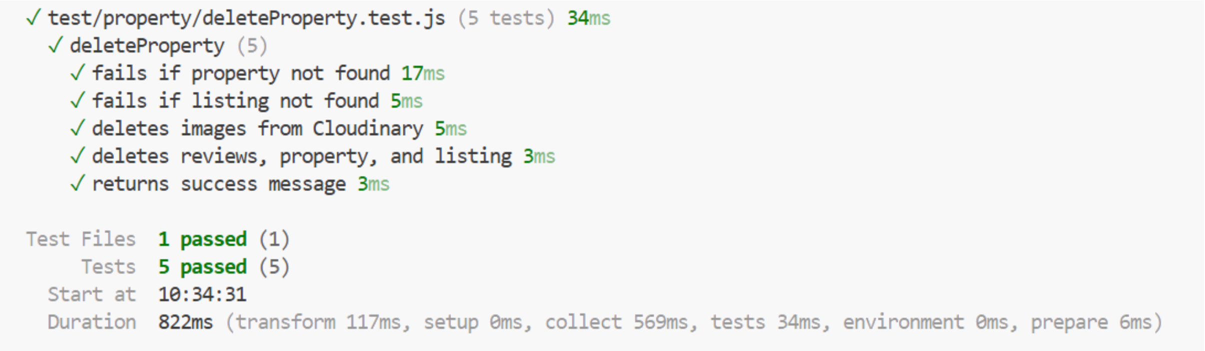
```
✓ test/property/updatePropertyDetails.test.js (6 tests) 49ms
  ✓ updatePropertyDetails (6)
    ✓ fails if property not found 26ms
    ✓ fails if listing not found 4ms
    ✓ fails on invalid JSON 4ms
    ✓ uploads images when files exist 5ms
    ✓ updates price and adds priceHistory entry 5ms
    ✓ successfully updates fields 4ms

Test Files  1 passed (1)
     Tests  6 passed (6)
  Start at  10:28:45
  Duration  829ms (transform 116ms, setup 0ms, collect 565ms, tests 49ms, environment 0ms, prepare 7ms)
```

updatePropertyStatus



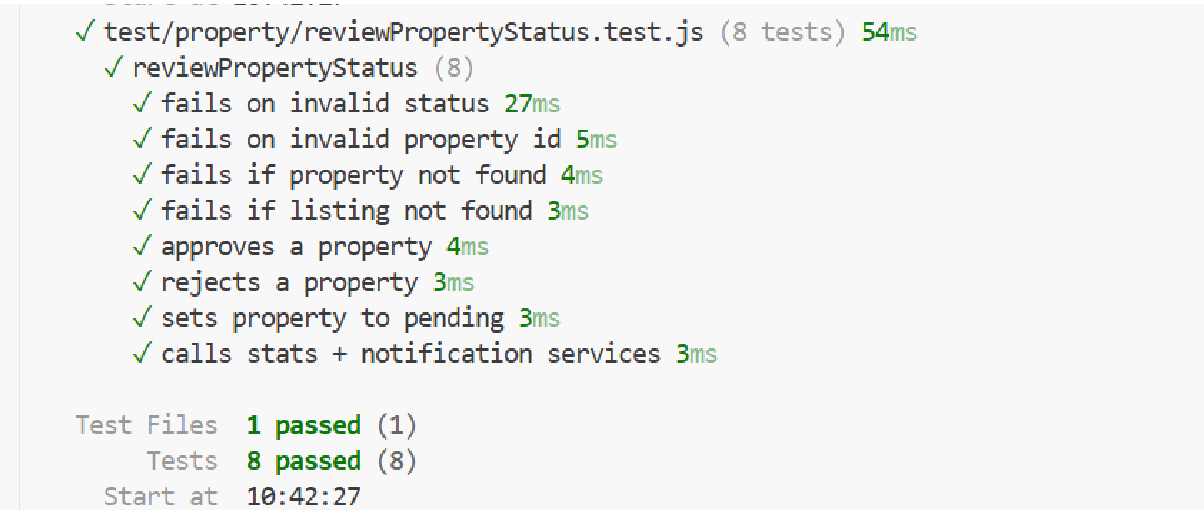
```
✓ test/property/updatePropertyStatus.test.js (4 tests) 49ms
  ✓ updatePropertyStatus (4)
    ✓ fails on invalid status 33ms
    ✓ fails if listing not found 6ms
    ✓ fails if user is not owner 4ms
    ✓ successfully updates status 4ms

Test Files  1 passed (1)
     Tests  4 passed (4)
  Start at  10:32:57
  Duration  603ms
```

deleteProperty



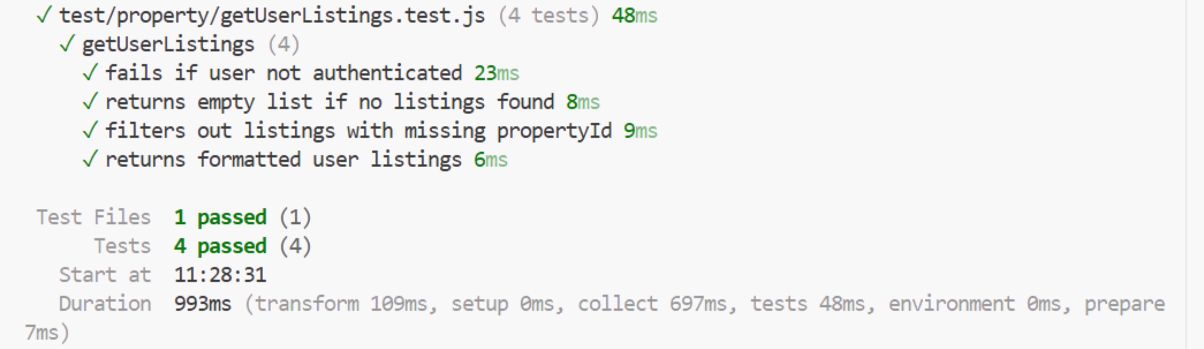
```
✓ test/property/deleteProperty.test.js (5 tests) 34ms
  ✓ deleteProperty (5)
    ✓ fails if property not found 17ms
    ✓ fails if listing not found 5ms
    ✓ deletes images from Cloudinary 5ms
    ✓ deletes reviews, property, and listing 3ms
    ✓ returns success message 3ms

Test Files  1 passed (1)
     Tests  5 passed (5)
  Start at  10:34:31
  Duration  822ms (transform 117ms, setup 0ms, collect 569ms, tests 34ms, environment 0ms, prepare 6ms)
```

reviewPropertyStatus



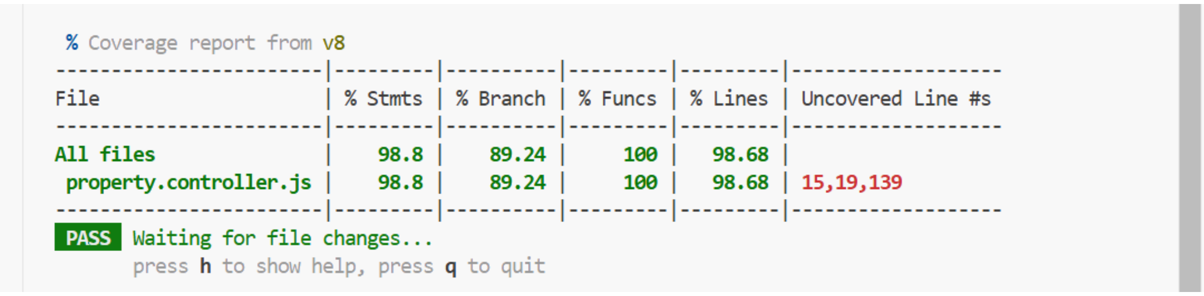
```
 ✓ test/property/reviewPropertyStatus.test.js (8 tests) 54ms
   ✓ reviewPropertyStatus (8)
     ✓ fails on invalid status 27ms
     ✓ fails on invalid property id 5ms
     ✓ fails if property not found 4ms
     ✓ fails if listing not found 3ms
     ✓ approves a property 4ms
     ✓ rejects a property 3ms
     ✓ sets property to pending 3ms
     ✓ calls stats + notification services 3ms

 Test Files  1 passed (1)
      Tests  8 passed (8)
   Start at  10:42:27
```

getUserListings



```
 ✓ test/property/getUserListings.test.js (4 tests) 48ms
   ✓ getUserListings (4)
     ✓ fails if user not authenticated 23ms
     ✓ returns empty list if no listings found 8ms
     ✓ filters out listings with missing propertyId 9ms
     ✓ returns formatted user listings 6ms

 Test Files  1 passed (1)
      Tests  4 passed (4)
   Start at  11:28:31
   Duration  993ms (transform 109ms, setup 0ms, collect 697ms, tests 48ms, environment 0ms, prepare
7ms)
```

coverage report:



```
 % Coverage report from v8
------------------------|---------|----------|---------|---------|-------------------
File                    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------------------|---------|----------|---------|---------|-------------------
All files               |    98.8 |    89.24 |     100 |   98.68 |
 property.controller.js |    98.8 |    89.24 |     100 |   98.68 | 15,19,139
------------------------|---------|----------|---------|---------|-------------------
 PASS  Waiting for file changes...
       press h to show help, press q to quit
```

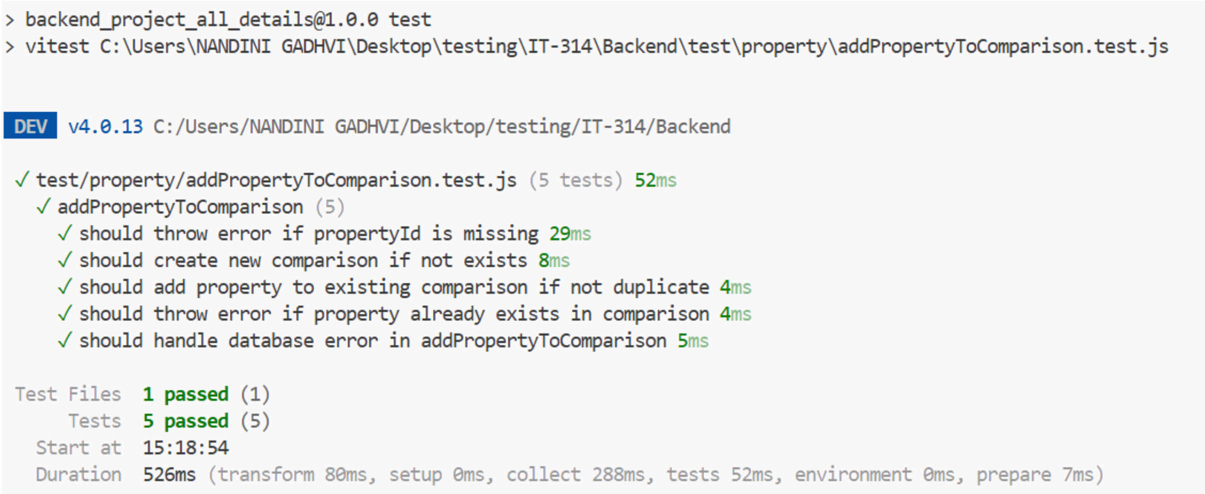
2. propertyComparison.controller.js

addPropertyToComparison

what function do
Adds a property to the user's comparison list and prevents duplicates.

what is tested
should throw error if propertyId is missing

should create new comparison if not exists
should add property to existing comparison if not duplicate
should throw error if property already exists in comparison
should handle database error in addPropertyToComparison
Note:
The UI restricts users to compare a maximum of 4 properties.
The backend does not enforce this limit.

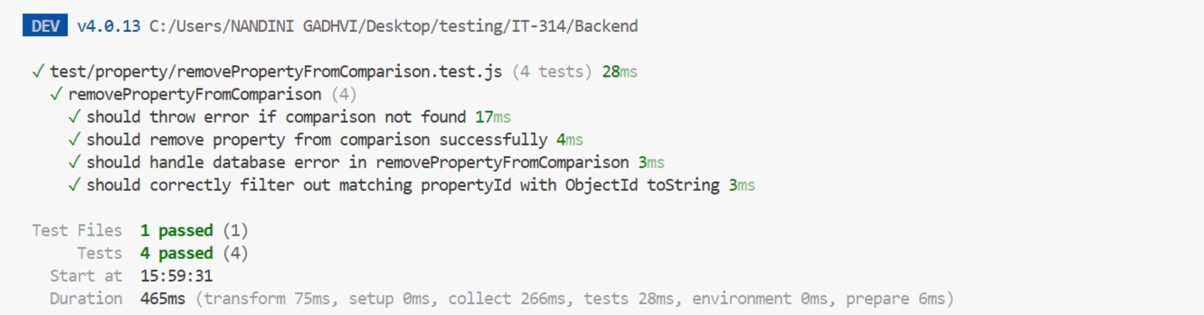


```
> backend_project_all_details@1.0.0 test
> vitest C:\Users\NANDINI GADHVI\Desktop\testing\IT-314\Backend\test\property\addPropertyToComparison.test.js


 DEV  v4.0.13 C:/Users/NANDINI GADHVI/Desktop/testing/IT-314/Backend

 ✓ test/property/addPropertyToComparison.test.js (5 tests) 52ms
   ✓ addPropertyToComparison (5)
     ✓ should throw error if propertyId is missing 29ms
     ✓ should create new comparison if not exists 8ms
     ✓ should add property to existing comparison if not duplicate 4ms
     ✓ should throw error if property already exists in comparison 4ms
     ✓ should handle database error in addPropertyToComparison 5ms

 Test Files  1 passed (1)
      Tests  5 passed (5)
   Start at  15:18:54
   Duration  526ms (transform 80ms, setup 0ms, collect 288ms, tests 52ms, environment 0ms, prepare 7ms)
```

removePropertyFromComparison

what function do
Remove one specific property from a user’s comparison list.
what is tested

should throw error if comparison not found
should remove property from comparison successfully
should handle database error in removePropertyFromComparison
should correctly filter out matching propertyId with ObjectId toString
Note:
If the propertyId doesn’t exist in the user’s comparison list, the backend does not throw an error.
It just keeps the list unchanged and still returns 200 OK.



```
 DEV  v4.0.13 C:/Users/NANDINI GADHVI/Desktop/testing/IT-314/Backend

 ✓ test/property/removePropertyFromComparison.test.js (4 tests) 28ms
   ✓ removePropertyFromComparison (4)
     ✓ should throw error if comparison not found 17ms
     ✓ should remove property from comparison successfully 4ms
     ✓ should handle database error in removePropertyFromComparison 3ms
     ✓ should correctly filter out matching propertyId with ObjectId toString 3ms

 Test Files  1 passed (1)
      Tests  4 passed (4)
   Start at  15:59:31
   Duration  465ms (transform 75ms, setup 0ms, collect 266ms, tests 28ms, environment 0ms, prepare 6ms)
```

getComparedProperties

what function do

It returns the list of properties that the user has selected for comparison.

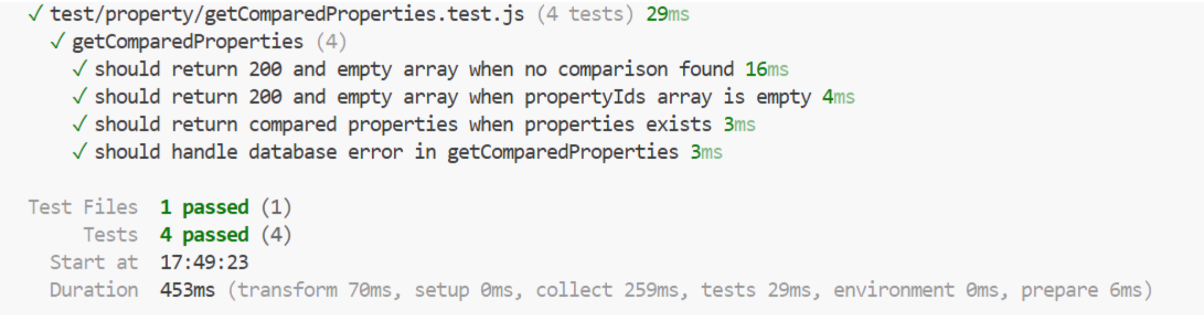
what is tested

should return 200 and empty array when no comparison found

should return 200 and empty array when propertyIds array is empty

should return compared properties when properties exists

should handle database error in getComparedProperties



```
 ✓ test/property/getComparedProperties.test.js (4 tests) 29ms
   ✓ getComparedProperties (4)
     ✓ should return 200 and empty array when no comparison found 16ms
     ✓ should return 200 and empty array when propertyIds array is empty 4ms
     ✓ should return compared properties when properties exists 3ms
     ✓ should handle database error in getComparedProperties 3ms

 Test Files  1 passed (1)
      Tests  4 passed (4)
   Start at  17:49:23
   Duration  453ms (transform 70ms, setup 0ms, collect 259ms, tests 29ms, environment 0ms, prepare 6ms)
```

clearComparison

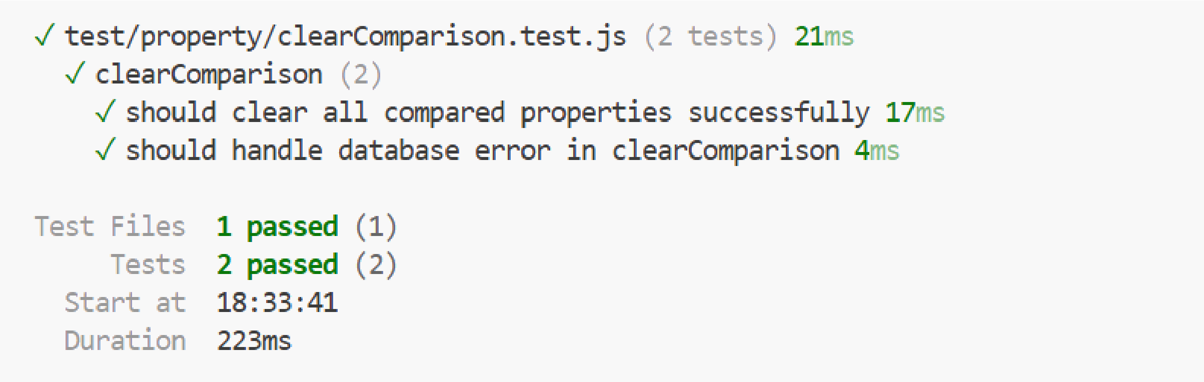
what function do

It deletes the entire comparison list for the logged-in user.

what is tested

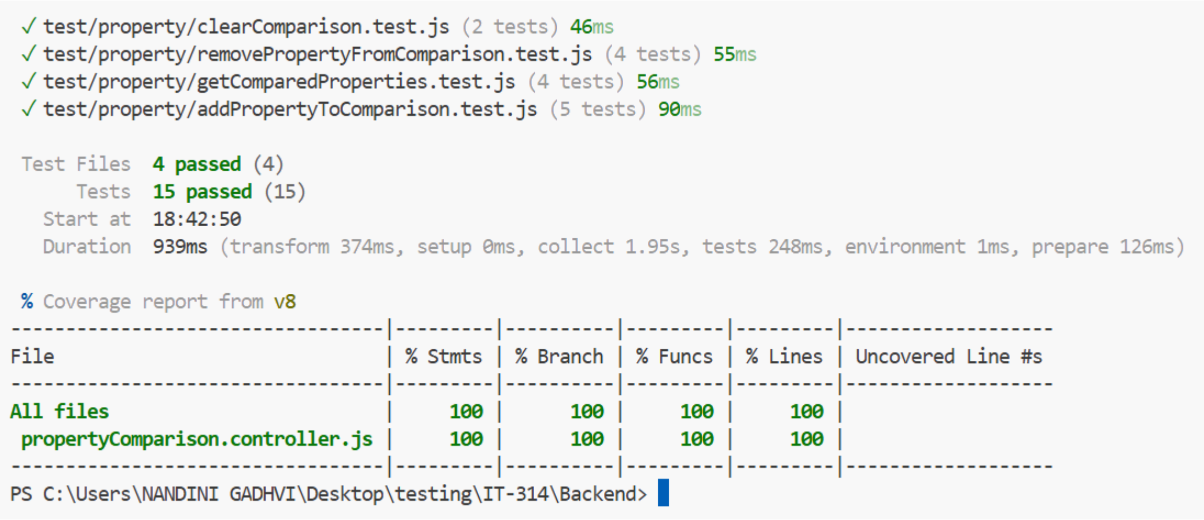
should clear all compared properties successfully

should handle database error in clearComparison



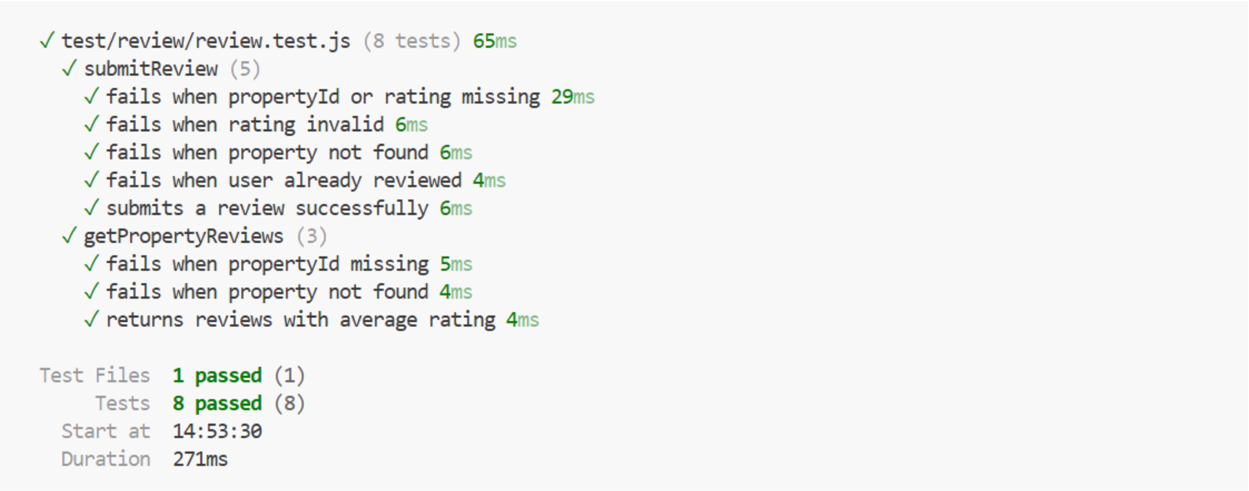
```
✓ test/property/clearComparison.test.js (2 tests) 21ms
  ✓ clearComparison (2)
    ✓ should clear all compared properties successfully 17ms
    ✓ should handle database error in clearComparison 4ms

 Test Files  1 passed (1)
      Tests  2 passed (2)
   Start at  18:33:41
   Duration  223ms
```

Coverage Report:



```
✓ test/property/clearComparison.test.js (2 tests) 46ms
✓ test/property/removePropertyFromComparison.test.js (4 tests) 55ms
✓ test/property/getComparedProperties.test.js (4 tests) 56ms
✓ test/property/addPropertyToComparison.test.js (5 tests) 90ms

 Test Files  4 passed (4)
      Tests  15 passed (15)
   Start at  18:42:50
   Duration  939ms (transform 374ms, setup 0ms, collect 1.95s, tests 248ms, environment 1ms, prepare 126ms)

 % Coverage report from v8
---------------------------------|---------|----------|---------|---------|-------------------
File                             | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------------------|---------|----------|---------|---------|-------------------
All files                        |     100 |      100 |     100 |     100 |
 propertyComparison.controller.js |     100 |      100 |     100 |     100 |
---------------------------------|---------|----------|---------|---------|-------------------
PS C:\Users\NANDINI GADHVI\Desktop\testing\IT-314\Backend>
```

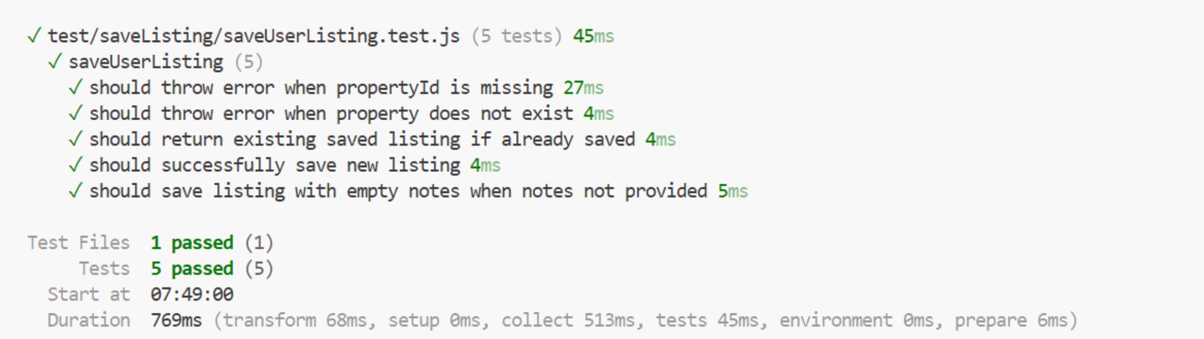
3. Review.controller.js



```
✓ test/review/review.test.js (8 tests) 65ms
  ✓ submitReview (5)
    ✓ fails when propertyId or rating missing 29ms
    ✓ fails when rating invalid 6ms
    ✓ fails when property not found 6ms
    ✓ fails when user already reviewed 4ms
    ✓ submits a review successfully 6ms
  ✓ getPropertyReviews (3)
    ✓ fails when propertyId missing 5ms
    ✓ fails when property not found 4ms
    ✓ returns reviews with average rating 4ms

 Test Files  1 passed (1)
      Tests  8 passed (8)
   Start at  14:53:30
   Duration  271ms
```

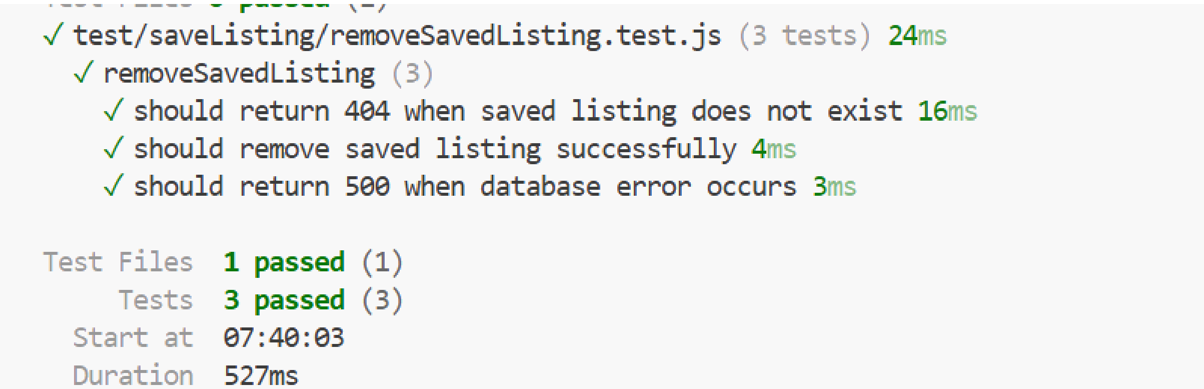
4. savedListing.controller.js

saveUserListing



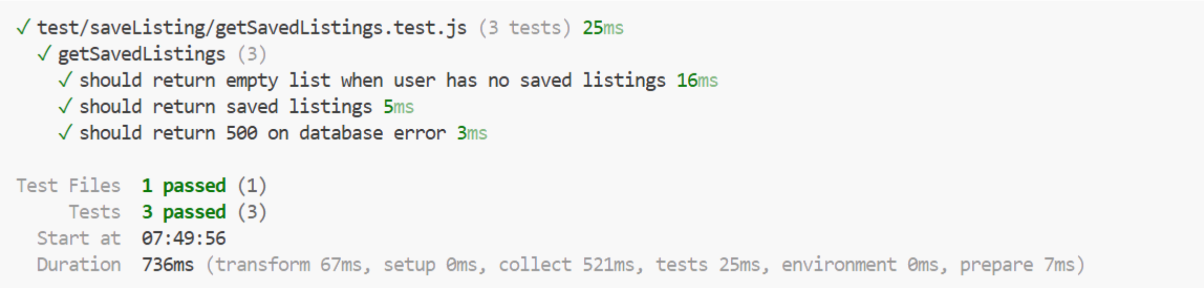
```
✓ test/saveListing/saveUserListing.test.js (5 tests) 45ms
  ✓ saveUserListing (5)
    ✓ should throw error when propertyId is missing 27ms
    ✓ should throw error when property does not exist 4ms
    ✓ should return existing saved listing if already saved 4ms
    ✓ should successfully save new listing 4ms
    ✓ should save listing with empty notes when notes not provided 5ms

Test Files  1 passed (1)
     Tests  5 passed (5)
  Start at  07:49:00
  Duration  769ms (transform 68ms, setup 0ms, collect 513ms, tests 45ms, environment 0ms, prepare 6ms)
```

removeSavedListing



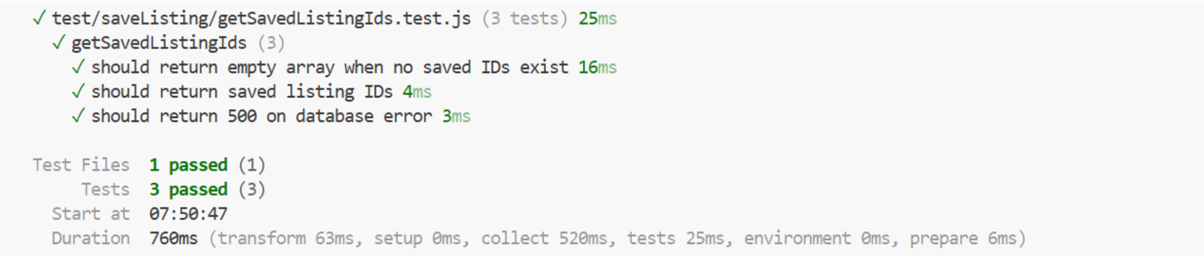
```
✓ test/saveListing/removeSavedListing.test.js (3 tests) 24ms
  ✓ removeSavedListing (3)
    ✓ should return 404 when saved listing does not exist 16ms
    ✓ should remove saved listing successfully 4ms
    ✓ should return 500 when database error occurs 3ms

Test Files  1 passed (1)
     Tests  3 passed (3)
  Start at  07:40:03
  Duration  527ms
```

getSavedListings



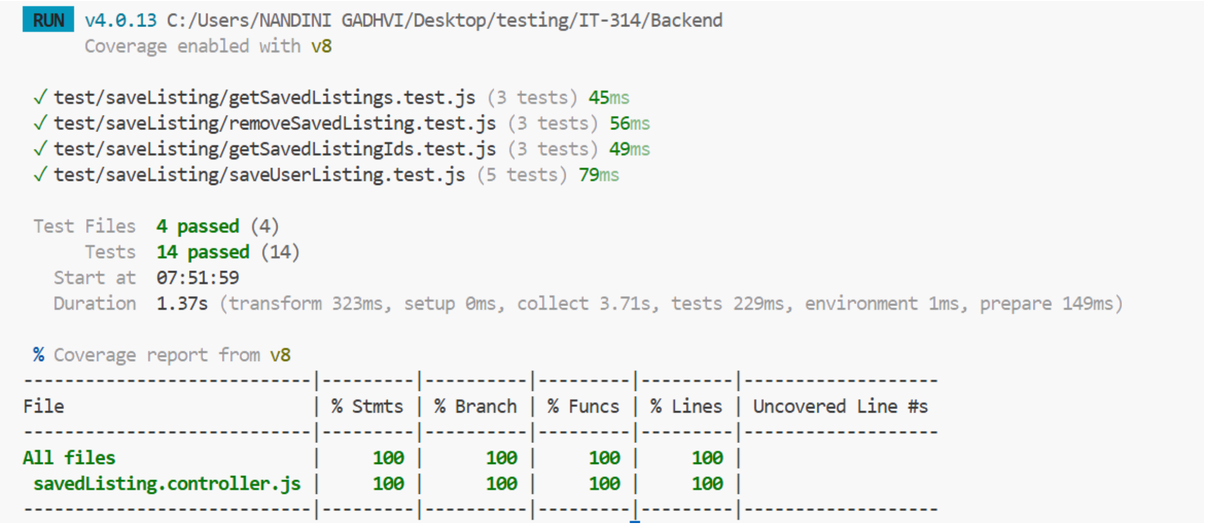
```
✓ test/saveListing/getSavedListings.test.js (3 tests) 25ms
  ✓ getSavedListings (3)
    ✓ should return empty list when user has no saved listings 16ms
    ✓ should return saved listings 5ms
    ✓ should return 500 on database error 3ms

Test Files  1 passed (1)
     Tests  3 passed (3)
  Start at  07:49:56
  Duration  736ms (transform 67ms, setup 0ms, collect 521ms, tests 25ms, environment 0ms, prepare 7ms)
```

getSavedListingIds



```
✓ test/saveListing/getSavedListingIds.test.js (3 tests) 25ms
  ✓ getSavedListingIds (3)
    ✓ should return empty array when no saved IDs exist 16ms
    ✓ should return saved listing IDs 4ms
    ✓ should return 500 on database error 3ms

Test Files  1 passed (1)
     Tests  3 passed (3)
  Start at  07:50:47
  Duration  760ms (transform 63ms, setup 0ms, collect 520ms, tests 25ms, environment 0ms, prepare 6ms)
```

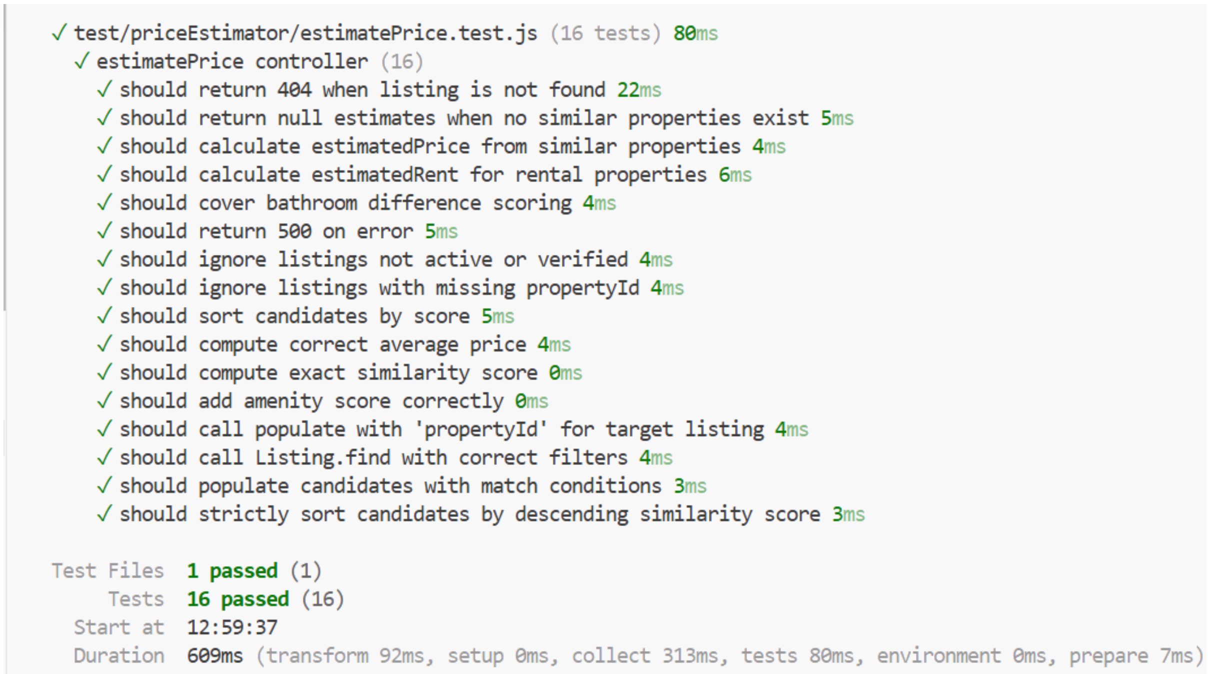
Coverage report:



```
 RUN  v4.0.13 C:/Users/NANDINI GADHVI/Desktop/testing/IT-314/Backend
      Coverage enabled with v8

 ✓ test/saveListing/getSavedListings.test.js (3 tests) 45ms
 ✓ test/saveListing/removeSavedListing.test.js (3 tests) 56ms
 ✓ test/saveListing/getSavedListingIds.test.js (3 tests) 49ms
 ✓ test/saveListing/saveUserListing.test.js (5 tests) 79ms

 Test Files  4 passed (4)
      Tests  14 passed (14)
   Start at  07:51:59
   Duration  1.37s (transform 323ms, setup 0ms, collect 3.71s, tests 229ms, environment 1ms, prepare 149ms)

 % Coverage report from v8
----------------------------|---------|----------|---------|---------|-------------------
File                        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------|---------|----------|---------|---------|-------------------
All files                   |     100 |      100 |     100 |     100 |
 savedListing.controller.js |     100 |      100 |     100 |     100 |
----------------------------|---------|----------|---------|---------|-------------------
```

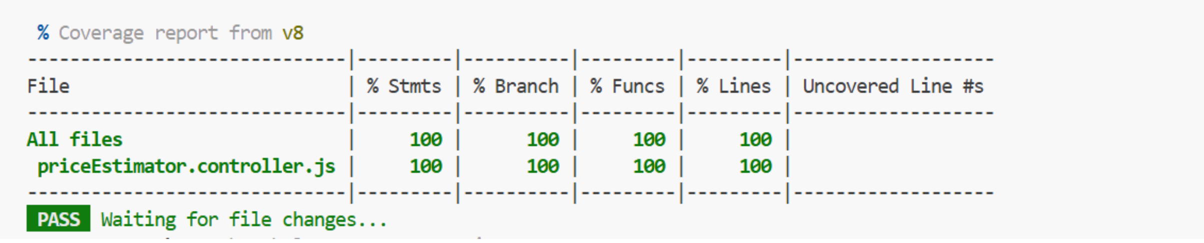
5. priceEstimator.controller.js

estimatePrice



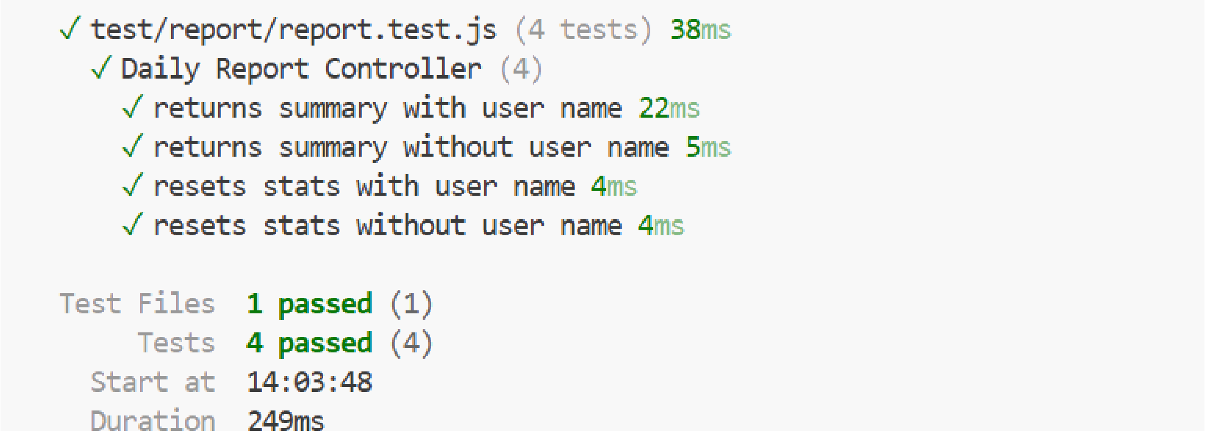
```
 ✓ test/priceEstimator/estimatePrice.test.js (16 tests) 80ms
   ✓ estimatePrice controller (16)
     ✓ should return 404 when listing is not found 22ms
     ✓ should return null estimates when no similar properties exist 5ms
     ✓ should calculate estimatedPrice from similar properties 4ms
     ✓ should calculate estimatedRent for rental properties 6ms
     ✓ should cover bathroom difference scoring 4ms
     ✓ should return 500 on error 5ms
     ✓ should ignore listings not active or verified 4ms
     ✓ should ignore listings with missing propertyId 4ms
     ✓ should sort candidates by score 5ms
     ✓ should compute correct average price 4ms
     ✓ should compute exact similarity score 0ms
     ✓ should add amenity score correctly 0ms
     ✓ should call populate with 'propertyId' for target listing 4ms
     ✓ should call Listing.find with correct filters 4ms
     ✓ should populate candidates with match conditions 3ms
     ✓ should strictly sort candidates by descending similarity score 3ms

 Test Files  1 passed (1)
      Tests  16 passed (16)
   Start at  12:59:37
   Duration  609ms (transform 92ms, setup 0ms, collect 313ms, tests 80ms, environment 0ms, prepare 7ms)
```

Coverage report:



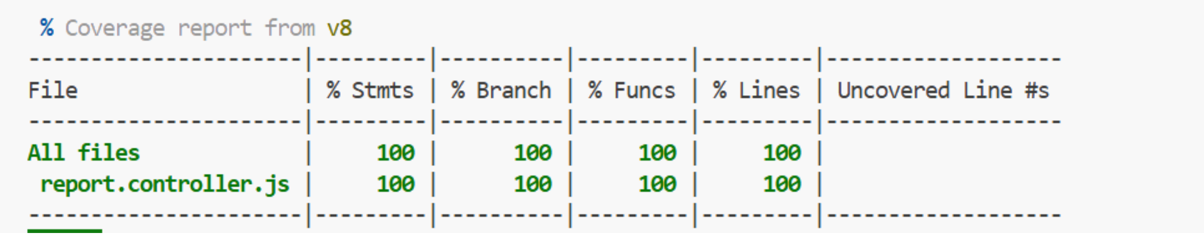
```
 % Coverage report from v8
-----------------------------|---------|----------|---------|---------|-------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------
All files                    |     100 |      100 |     100 |     100 |
 priceEstimator.controller.js |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------
PASS  Waiting for file changes...
```

6. **report.controller.js**



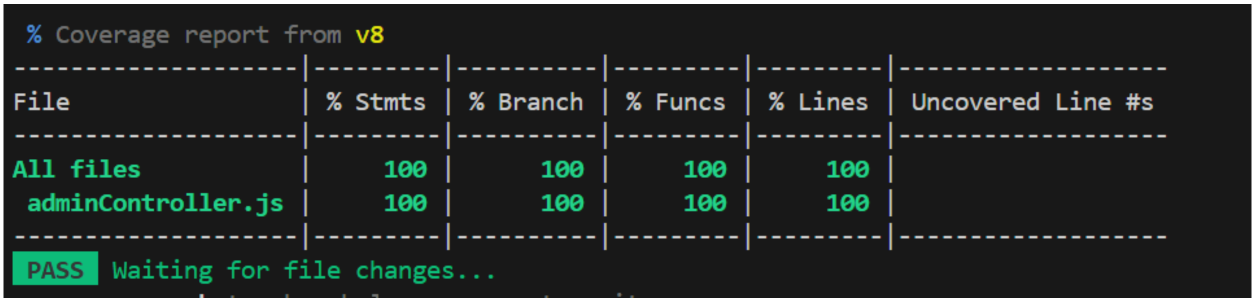
```
 ✓ test/report/report.test.js (4 tests) 38ms
   ✓ Daily Report Controller (4)
     ✓ returns summary with user name 22ms
     ✓ returns summary without user name 5ms
     ✓ resets stats with user name 4ms
     ✓ resets stats without user name 4ms

 Test Files  1 passed (1)
      Tests  4 passed (4)
   Start at  14:03:48
   Duration  249ms
```

Coverage report

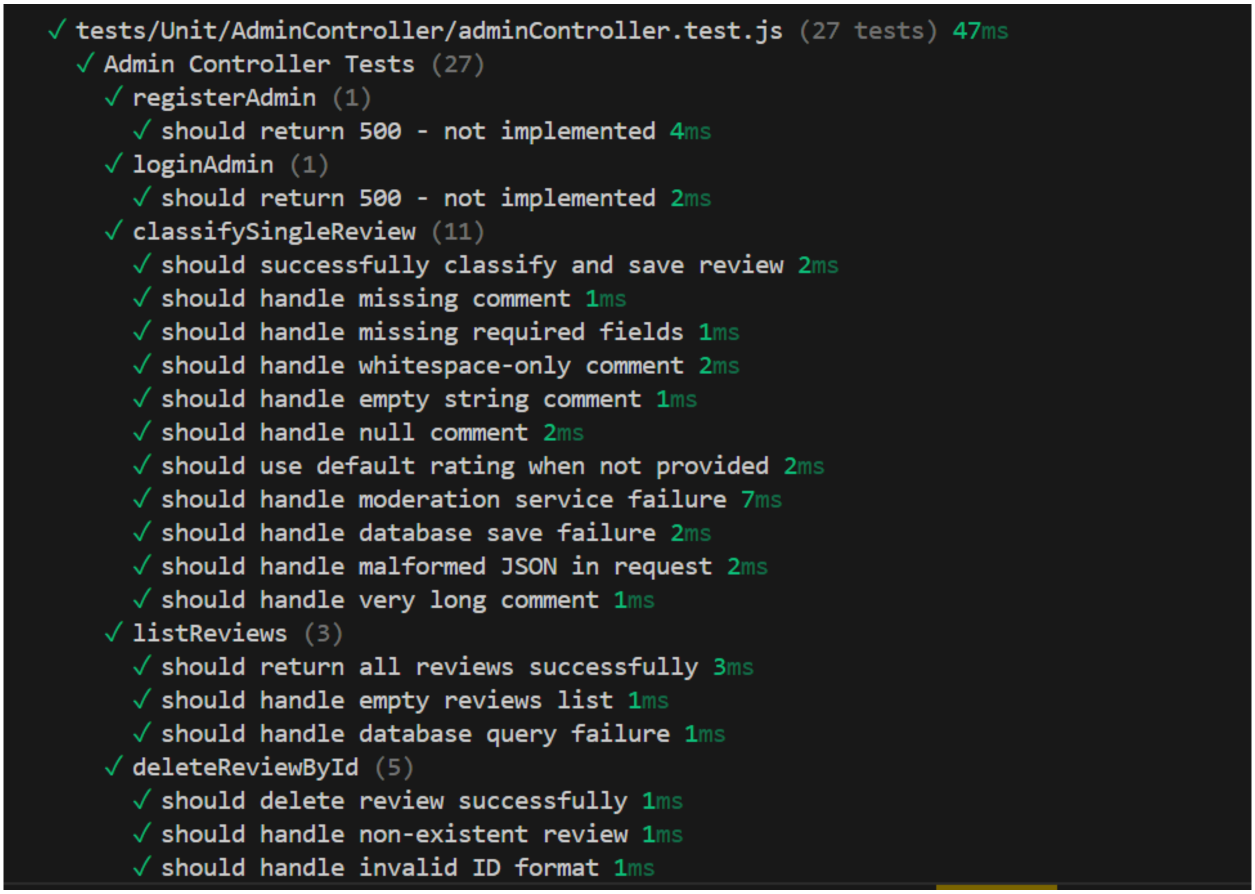


```
 % Coverage report from v8
----------------------|---------|----------|---------|---------|-------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------
All files             |     100 |      100 |     100 |     100 |
 report.controller.js |     100 |      100 |     100 |     100 |
----------------------|---------|----------|---------|---------|-------------------
```

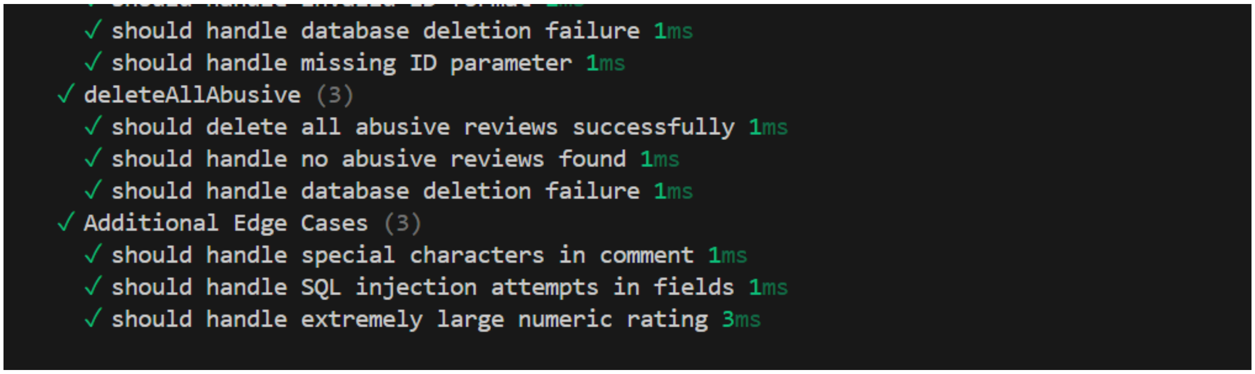
7. **Admin Controller:**



```
 % Coverage report from v8
--------------------|---------|----------|---------|---------|-------------------
File                | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------|---------|----------|---------|---------|-------------------
All files           |     100 |      100 |     100 |     100 |
 adminController.js |     100 |      100 |     100 |     100 |
--------------------|---------|----------|---------|---------|-------------------
PASS  Waiting for file changes...
```

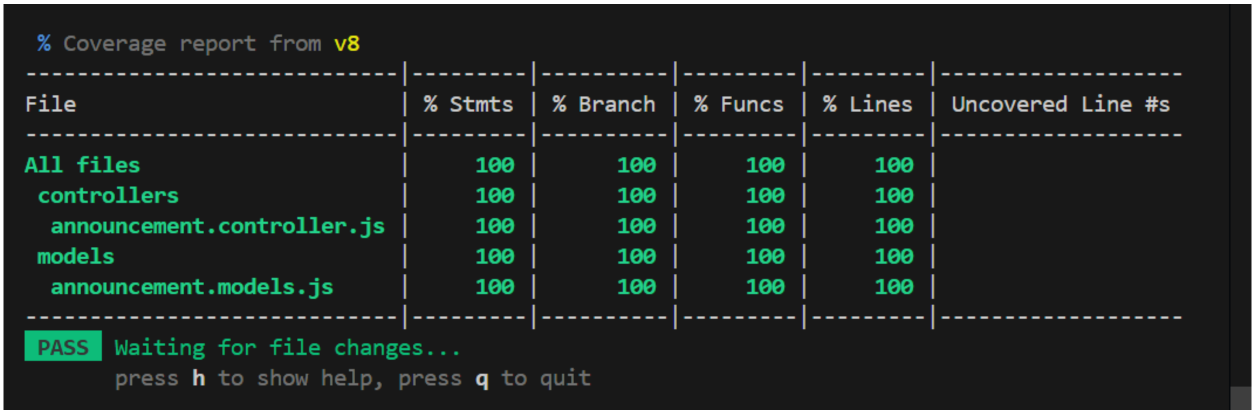


```
✓ tests/Unit/AdminController/adminController.test.js (27 tests) 47ms
  ✓ Admin Controller Tests (27)
    ✓ registerAdmin (1)
      ✓ should return 500 - not implemented 4ms
    ✓ loginAdmin (1)
      ✓ should return 500 - not implemented 2ms
    ✓ classifySingleReview (11)
      ✓ should successfully classify and save review 2ms
      ✓ should handle missing comment 1ms
      ✓ should handle missing required fields 1ms
      ✓ should handle whitespace-only comment 2ms
      ✓ should handle empty string comment 1ms
      ✓ should handle null comment 2ms
      ✓ should use default rating when not provided 2ms
      ✓ should handle moderation service failure 7ms
      ✓ should handle database save failure 2ms
      ✓ should handle malformed JSON in request 2ms
      ✓ should handle very long comment 1ms
    ✓ listReviews (3)
      ✓ should return all reviews successfully 3ms
      ✓ should handle empty reviews list 1ms
      ✓ should handle database query failure 1ms
    ✓ deleteReviewById (5)
      ✓ should delete review successfully 1ms
      ✓ should handle non-existent review 1ms
      ✓ should handle invalid ID format 1ms
```

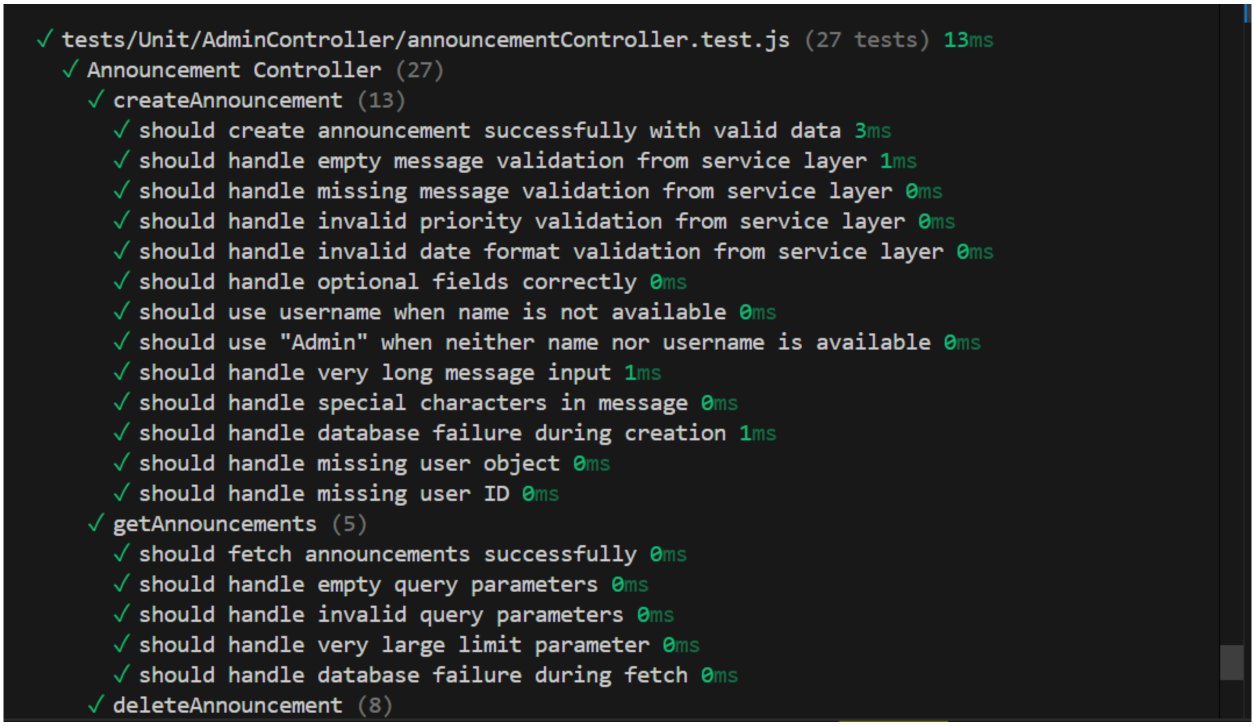


```
      ✓ should handle database deletion failure 1ms
      ✓ should handle missing ID parameter 1ms
    ✓ deleteAllAbusive (3)
      ✓ should delete all abusive reviews successfully 1ms
      ✓ should handle no abusive reviews found 1ms
      ✓ should handle database deletion failure 1ms
    ✓ Additional Edge Cases (3)
      ✓ should handle special characters in comment 1ms
      ✓ should handle SQL injection attempts in fields 1ms
      ✓ should handle extremely large numeric rating 3ms
```

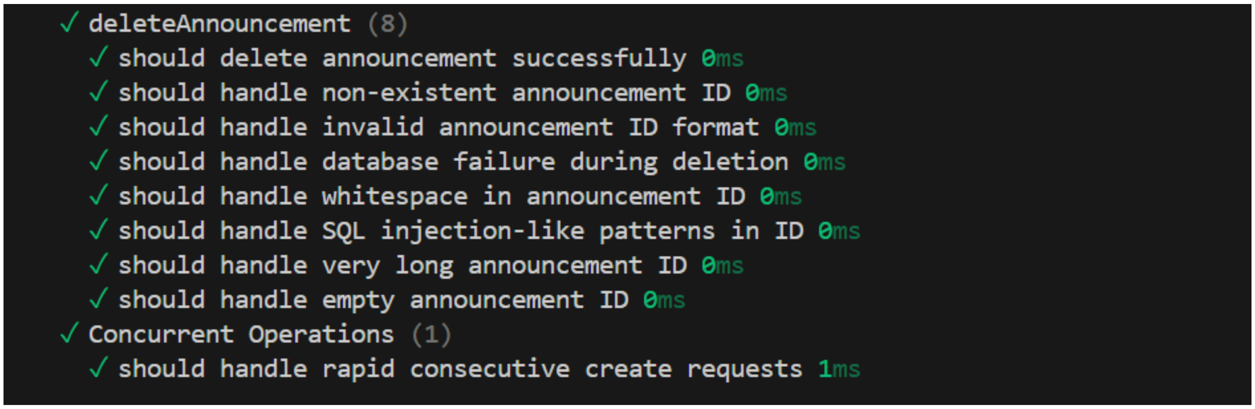
8. Announcement Controller



```
 % Coverage report from v8
-----------------------------|---------|----------|---------|---------|-------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------
All files                    |     100 |      100 |     100 |     100 |
 controllers                 |     100 |      100 |     100 |     100 |
  announcement.controller.js |     100 |      100 |     100 |     100 |
 models                      |     100 |      100 |     100 |     100 |
  announcement.models.js     |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------
 PASS  Waiting for file changes...
       press h to show help, press q to quit
```

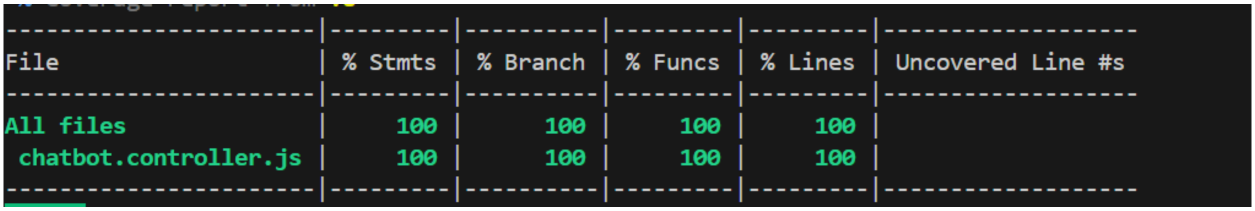


```
 ✓ tests/Unit/AdminController/announcementController.test.js (27 tests) 13ms
   ✓ Announcement Controller (27)
     ✓ createAnnouncement (13)
       ✓ should create announcement successfully with valid data 3ms
       ✓ should handle empty message validation from service layer 1ms
       ✓ should handle missing message validation from service layer 0ms
       ✓ should handle invalid priority validation from service layer 0ms
       ✓ should handle invalid date format validation from service layer 0ms
       ✓ should handle optional fields correctly 0ms
       ✓ should use username when name is not available 0ms
       ✓ should use "Admin" when neither name nor username is available 0ms
       ✓ should handle very long message input 1ms
       ✓ should handle special characters in message 0ms
       ✓ should handle database failure during creation 1ms
       ✓ should handle missing user object 0ms
       ✓ should handle missing user ID 0ms
     ✓ getAnnouncements (5)
       ✓ should fetch announcements successfully 0ms
       ✓ should handle empty query parameters 0ms
       ✓ should handle invalid query parameters 0ms
       ✓ should handle very large limit parameter 0ms
       ✓ should handle database failure during fetch 0ms
     ✓ deleteAnnouncement (8)
```

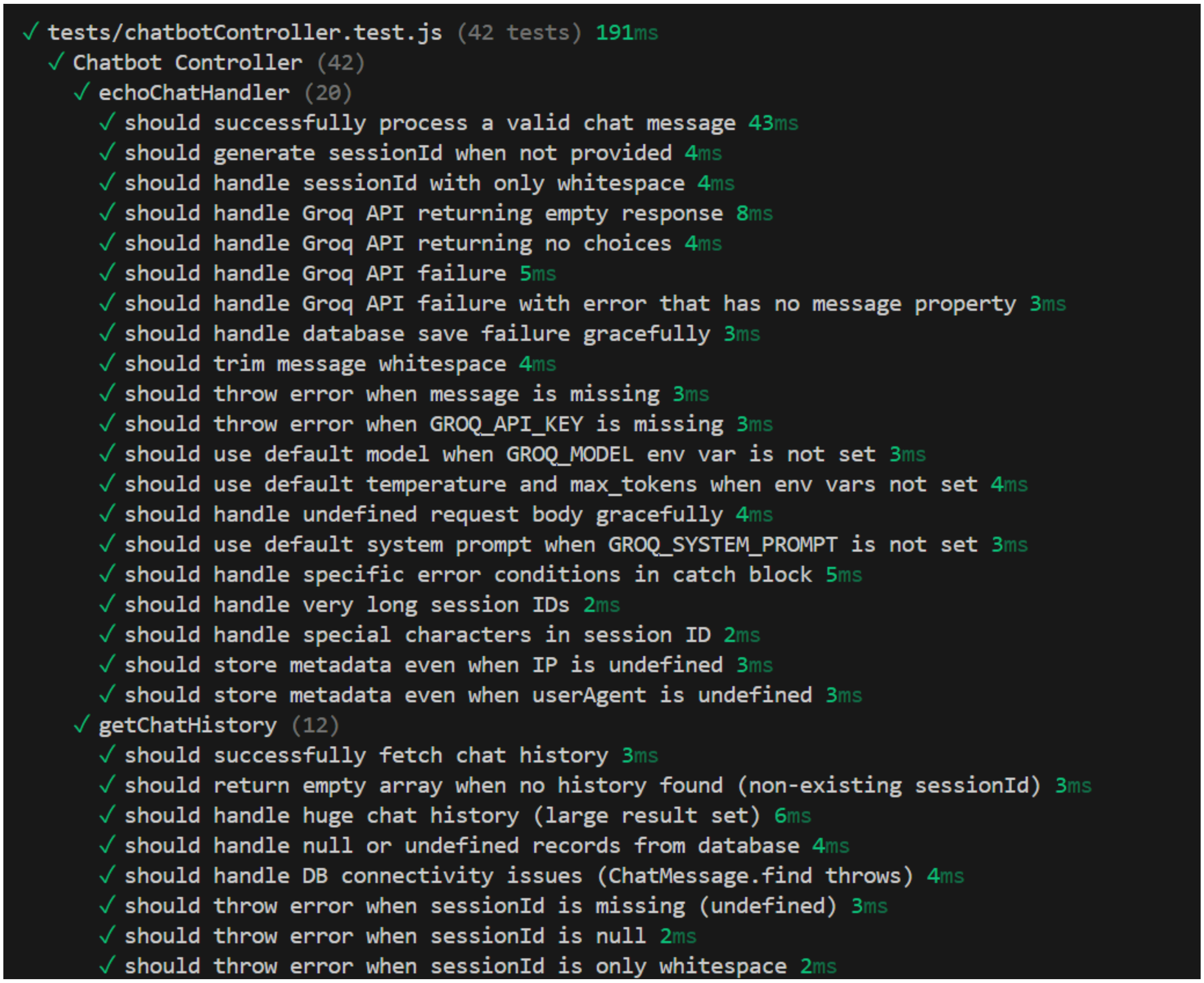


```
     ✓ deleteAnnouncement (8)
       ✓ should delete announcement successfully 0ms
       ✓ should handle non-existent announcement ID 0ms
       ✓ should handle invalid announcement ID format 0ms
       ✓ should handle database failure during deletion 0ms
       ✓ should handle whitespace in announcement ID 0ms
       ✓ should handle SQL injection-like patterns in ID 0ms
       ✓ should handle very long announcement ID 0ms
       ✓ should handle empty announcement ID 0ms
     ✓ Concurrent Operations (1)
       ✓ should handle rapid consecutive create requests 1ms
```

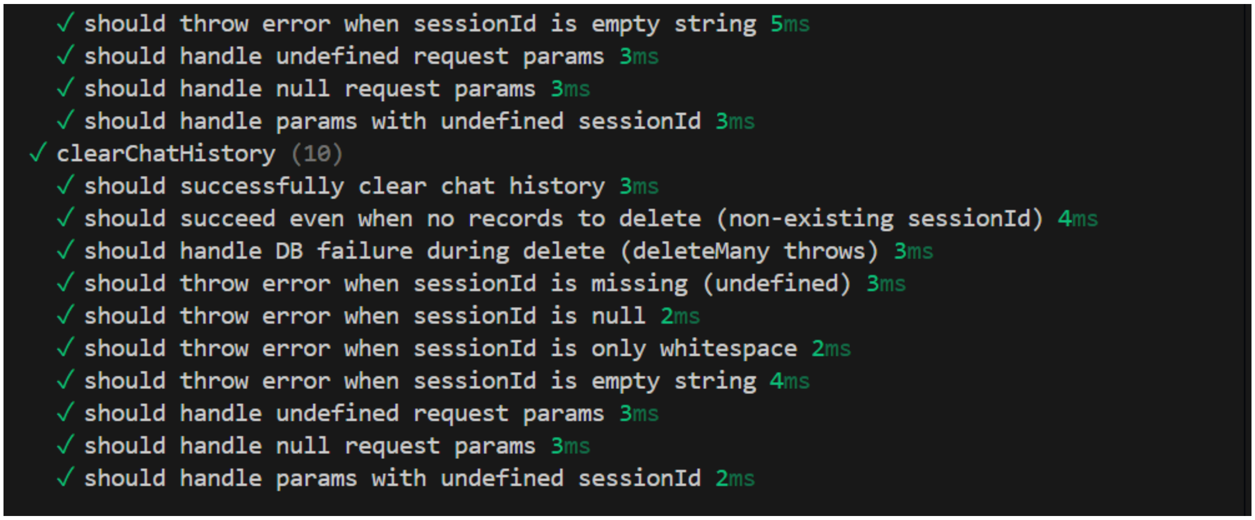
9. Chatbot Controller



```
-----------------------|---------|----------|---------|---------|-------------------
File                   | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------|---------|----------|---------|---------|-------------------
All files              |     100 |      100 |     100 |     100 |
 chatbot.controller.js |     100 |      100 |     100 |     100 |
-----------------------|---------|----------|---------|---------|-------------------
```

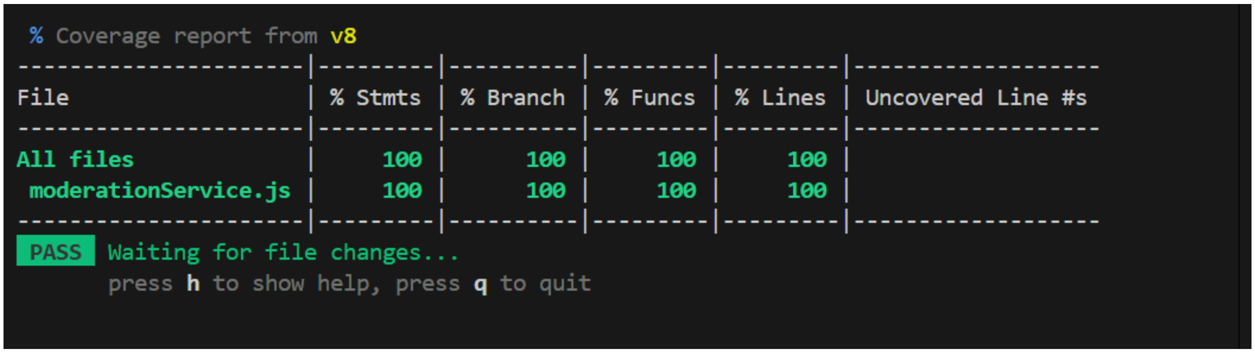


```
 ✓ tests/chatbotController.test.js (42 tests) 191ms
   ✓ Chatbot Controller (42)
     ✓ echoChatHandler (20)
       ✓ should successfully process a valid chat message 43ms
       ✓ should generate sessionId when not provided 4ms
       ✓ should handle sessionId with only whitespace 4ms
       ✓ should handle Groq API returning empty response 8ms
       ✓ should handle Groq API returning no choices 4ms
       ✓ should handle Groq API failure 5ms
       ✓ should handle Groq API failure with error that has no message property 3ms
       ✓ should handle database save failure gracefully 3ms
       ✓ should trim message whitespace 4ms
       ✓ should throw error when message is missing 3ms
       ✓ should throw error when GROQ_API_KEY is missing 3ms
       ✓ should use default model when GROQ_MODEL env var is not set 3ms
       ✓ should use default temperature and max_tokens when env vars not set 4ms
       ✓ should handle undefined request body gracefully 4ms
       ✓ should use default system prompt when GROQ_SYSTEM_PROMPT is not set 3ms
       ✓ should handle specific error conditions in catch block 5ms
       ✓ should handle very long session IDs 2ms
       ✓ should handle special characters in session ID 2ms
       ✓ should store metadata even when IP is undefined 3ms
       ✓ should store metadata even when userAgent is undefined 3ms
     ✓ getChatHistory (12)
       ✓ should successfully fetch chat history 3ms
       ✓ should return empty array when no history found (non-existing sessionId) 3ms
       ✓ should handle huge chat history (large result set) 6ms
       ✓ should handle null or undefined records from database 4ms
       ✓ should handle DB connectivity issues (ChatMessage.find throws) 4ms
       ✓ should throw error when sessionId is missing (undefined) 3ms
       ✓ should throw error when sessionId is null 2ms
       ✓ should throw error when sessionId is only whitespace 2ms
```

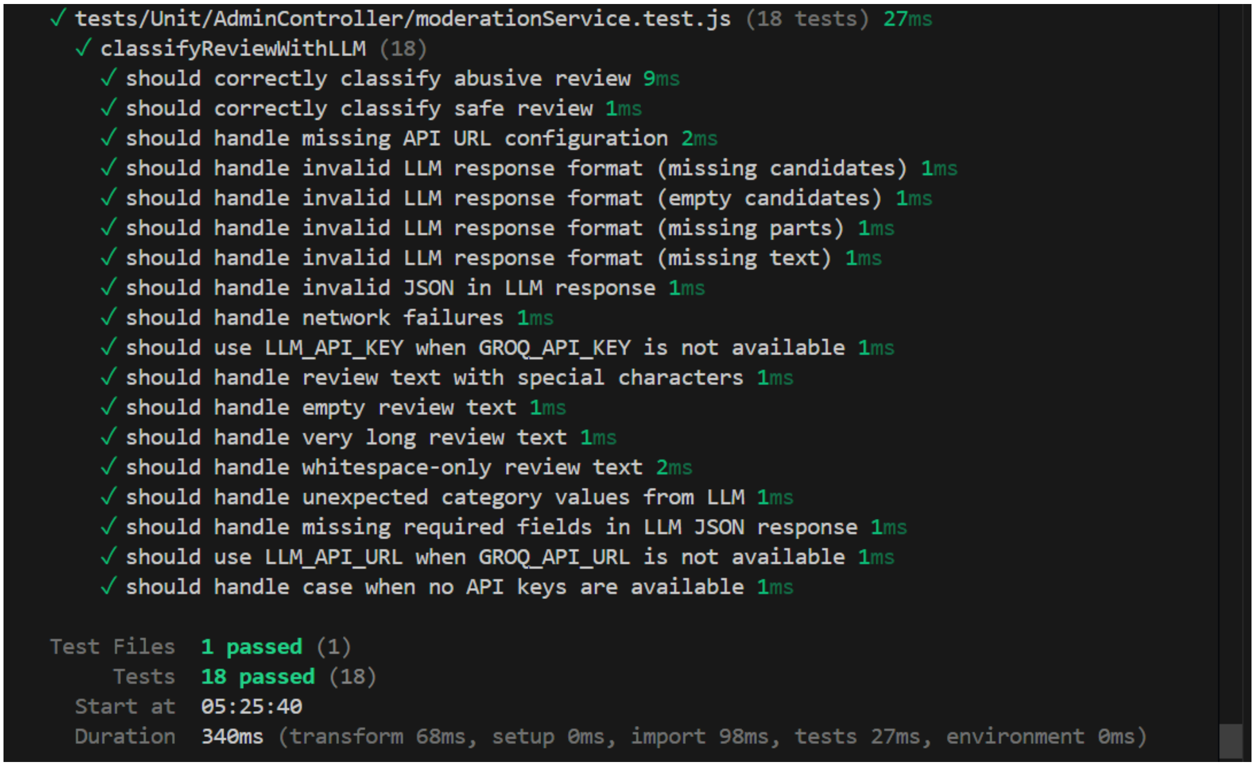


```
    ✓ should throw error when sessionId is empty string 5ms
    ✓ should handle undefined request params 3ms
    ✓ should handle null request params 3ms
    ✓ should handle params with undefined sessionId 3ms
  ✓ clearChatHistory (10)
    ✓ should successfully clear chat history 3ms
    ✓ should succeed even when no records to delete (non-existing sessionId) 4ms
    ✓ should handle DB failure during delete (deleteMany throws) 3ms
    ✓ should throw error when sessionId is missing (undefined) 3ms
    ✓ should throw error when sessionId is null 2ms
    ✓ should throw error when sessionId is only whitespace 2ms
    ✓ should throw error when sessionId is empty string 4ms
    ✓ should handle undefined request params 3ms
    ✓ should handle null request params 3ms
    ✓ should handle params with undefined sessionId 2ms
```

10. Moderation Service



```
 % Coverage report from v8
----------------------|---------|----------|---------|---------|-------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------
All files             |     100 |      100 |     100 |     100 |
 moderationService.js |     100 |      100 |     100 |     100 |
----------------------|---------|----------|---------|---------|-------------------
PASS  Waiting for file changes...
       press h to show help, press q to quit
```



```
 ✓ tests/Unit/AdminController/moderationService.test.js (18 tests) 27ms
   ✓ classifyReviewWithLLM (18)
     ✓ should correctly classify abusive review 9ms
     ✓ should correctly classify safe review 1ms
     ✓ should handle missing API URL configuration 2ms
     ✓ should handle invalid LLM response format (missing candidates) 1ms
     ✓ should handle invalid LLM response format (empty candidates) 1ms
     ✓ should handle invalid LLM response format (missing parts) 1ms
     ✓ should handle invalid LLM response format (missing text) 1ms
     ✓ should handle invalid JSON in LLM response 1ms
     ✓ should handle network failures 1ms
     ✓ should use LLM_API_KEY when GROQ_API_KEY is not available 1ms
     ✓ should handle review text with special characters 1ms
     ✓ should handle empty review text 1ms
     ✓ should handle very long review text 1ms
     ✓ should handle whitespace-only review text 2ms
     ✓ should handle unexpected category values from LLM 1ms
     ✓ should handle missing required fields in LLM JSON response 1ms
     ✓ should use LLM_API_URL when GROQ_API_URL is not available 1ms
     ✓ should handle case when no API keys are available 1ms

 Test Files  1 passed (1)
      Tests  18 passed (18)
   Start at  05:25:40
   Duration  340ms (transform 68ms, setup 0ms, import 98ms, tests 27ms, environment 0ms)
```