# IT314 - Software engineering

## Group - 11

## Unit Testing - Backend

**All files**

**87.83%** Statements 3356/3821 **78.83%** Branches 540/685 **93.42%** Functions 71/76 **87.83%** Lines 3356/3821

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 | |
|---|---|---|---|---|---|---|---|---|
| backend | 96.02% | 169/176 | 78.12% | 25/32 | 100% | 1/1 | 96.02% | |
| backend/controllers | 97.87% | 1522/1555 | 91.07% | 296/325 | 100% | 31/31 | 97.87% | |
| backend/middlewares | 100% | 49/49 | 100% | 10/10 | 100% | 2/2 | 100% | |
| backend/models | 100% | 213/213 | 100% | 10/10 | 100% | 2/2 | 100% | |
| backend/routes | 100% | 101/101 | 100% | 5/5 | 100% | 0/0 | 100% | |
| backend/services | 79.24% | 359/453 | 61.44% | 51/83 | 91.66% | 11/12 | 79.24% | |
| backend/utils | 74.01% | 943/1274 | 65% | 143/220 | 85.71% | 24/28 | 74.01% | |

# Controllers

**All files backend/controllers**

**97.87%** Statements 1522/1555 **91.07%** Branches 296/325 **100%** Functions 31/31 **97.87%** Lines 1522/1555

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

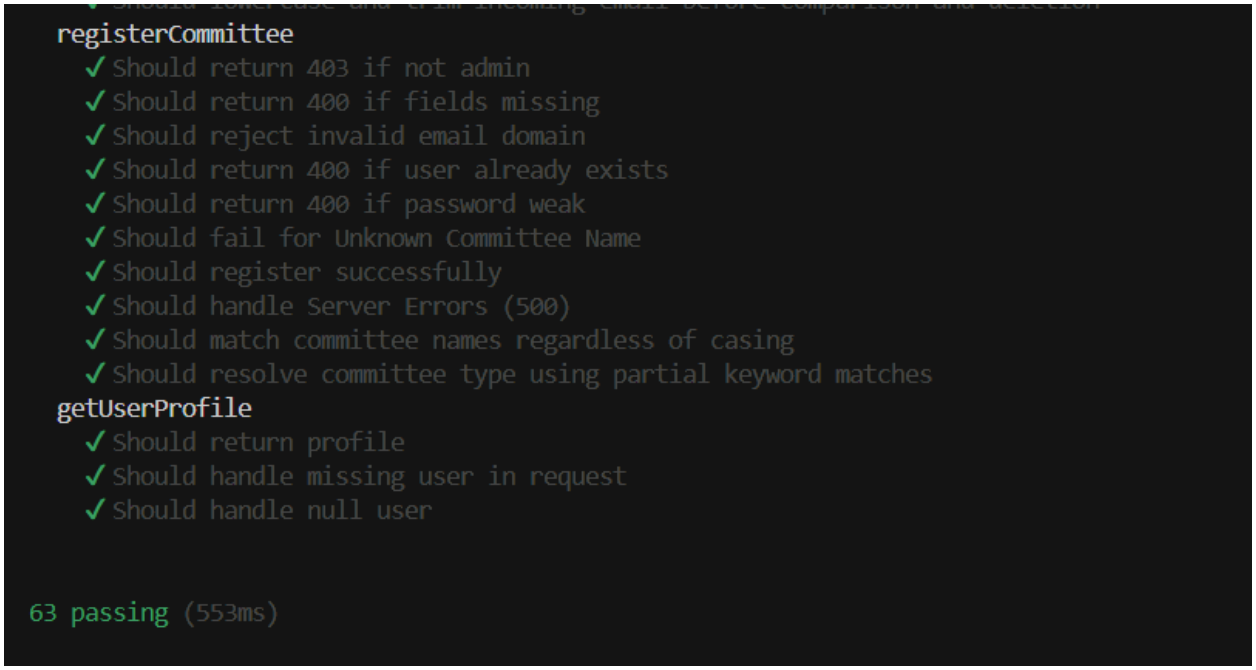
Filter:

| File | Statements | | Branches | | Functions | | Lines | |
|---|---|---|---|---|---|---|---|---|
| authController.js | 100% | 409/409 | 100% | 102/102 | 100% | 8/8 | 100% | |
| complaintController.js | 96.63% | 948/981 | 85.2% | 167/196 | 100% | 17/17 | 96.63% | |
| notificationController.js | 100% | 84/84 | 100% | 11/11 | 100% | 4/4 | 100% | |
| reportController.js | 100% | 81/81 | 100% | 16/16 | 100% | 2/2 | 100% | |

1. authController

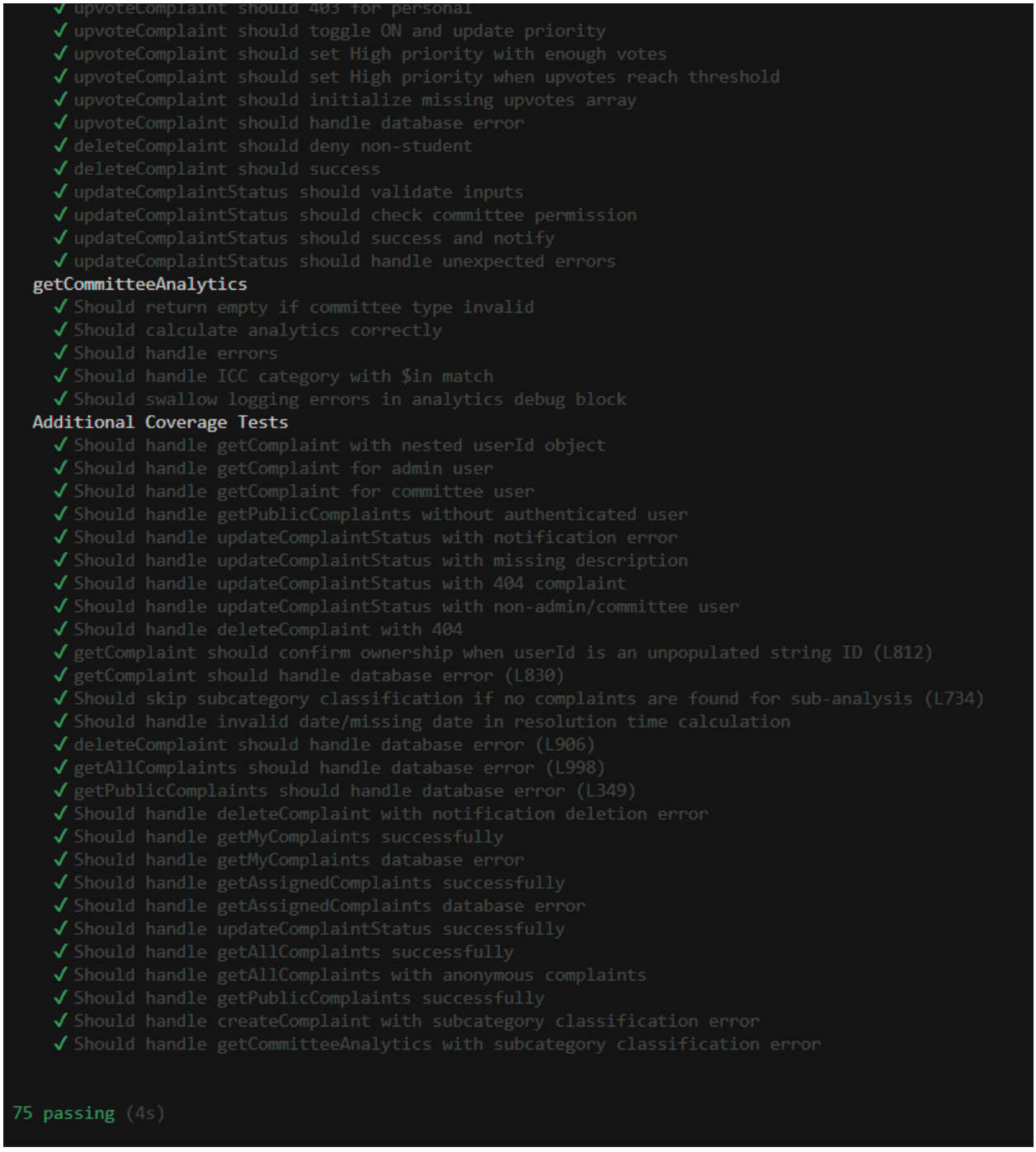
```
  Auth Controller 100% Coverage Test Suite
    loginUser
      ✓ Should return 400 if fields are missing
      ✓ Should return 401 if User Not Found
      ✓ Should return 401 if Password Incorrect
      ✓ Should return 403 if Role Mismatch
      ✓ Should login successfully
      ✓ Should handle Server Errors (500)
    changePassword
      ✓ Should return 400 if fields missing
      ✓ Should return 400 for Weak New Password
      ✓ Should return 404 if user not found
      ✓ Should return 401 if current password incorrect
      ✓ Should return 400 if New Password is same as Old
      ✓ Should change password successfully
      ✓ Should handle Server Errors (500)
    forgotPassword
      ✓ Should return 400 if email missing
      ✓ Should return 404 if email not found
Email configuration incomplete. Email sending will be skipped.
      ✓ Should attempt to send OTP
      ✓ Should send OTP and return success response (148ms)
      ✓ Should return 500 when sendOTPEmail reports failure
      ✓ Should handle DB errors (500)
    resetPassword
      ✓ Should return 400 if fields missing
      ✓ Should require email even when otp and password exist
      ✓ Should require otp even when email and password exist
      ✓ Should require new password even when email and otp exist
      ✓ Should return 400 if password too short
      ✓ Should allow password length of exactly six characters
      ✓ Should fail if OTP is invalid
      ✓ Should return 404 if User not found after OTP verify
      ✓ Should return 400 if New Password same as Old
```

```
  ✓ Should return 404 if User not found after OTP verify
  ✓ Should return 400 if New Password same as Old
  ✓ Should reset password successfully
  ✓ Should handle Server Errors (500)
registerStudent
  ✓ Should return 403 if not admin
  ✓ Should return 400 if fields missing
  ✓ Should reject invalid email domain
  ✓ Should return 400 if User already exists
  ✓ Should return 400 if Password weak
  ✓ Should register student successfully
  ✓ Should handle Server Errors (500)
deleteUser
  ✓ Should return 403 if not admin
  ✓ Should return 400 if email missing
  ✓ Should return 404 if user not found
  ✓ Should return 400 if role mismatch
  ✓ Should return 400 if trying to delete admin
  ✓ Should prevent Admin from deleting themselves
  ✓ Should prevent Admin from deleting themselves when target is not admin
  ✓ Should block admin from deleting own account via normalized email check
  ✓ Should delete user successfully
  ✓ Should handle Server Errors (500)
  ✓ Should trim whitespace from email before querying
  ✓ Should allow deletion when provided role matches the target role
  ✓ Should lowercase and trim incoming email before comparison and deletion
registerCommittee
  ✓ Should return 403 if not admin
  ✓ Should return 400 if fields missing
  ✓ Should reject invalid email domain
  ✓ Should return 400 if user already exists
  ✓ Should return 400 if password weak
  ✓ Should fail for Unknown Committee Name
  ✓ Should register successfully
```




2. ComplaintController

```
Complaint Controller 100% Coverage Tests
  Helpers
    ✓ normalizeCategory should map ICC variants
    ✓ resolveCommitteeCategory should map Hostels to Hostel Management
  createComplaint
    ✓ Should return 400 if title/desc missing
    ✓ Should use default invalid complaint message when error message is empty
    ✓ Should handle INVALID_COMPLAINT from AI
    ✓ Should handle Server Error (500)
    ✓ Should handle notification errors gracefully
    ✓ Should use fallback if AI crashes, create complaint, and trigger notifications
  Getters
    ✓ getMyComplaintsStats should return counts
    ✓ getMyComplaintsStats should handle errors
    ✓ getMyComplaints should fetch and sort
    ✓ getPublicComplaints should sort by upvotes and format anonymous
    ✓ getPublicComplaints should use createdAt tiebreaker and mask anonymous user
    ✓ getAssignedComplaints should deny non-committee
    ✓ getAssignedComplaints should return empty if no category map
    ✓ getAssignedComplaints should fetch and sort assigned
    ✓ getAssignedComplaintsStats should deny non-committee
    ✓ getAssignedComplaintsStats should return counts
    ✓ getAssignedComplaintsStats should return zeros if no category mapping
    ✓ getAssignedComplaintsStats should handle database errors
    ✓ getComplaint should return 404
    ✓ getComplaint should deny access
    ✓ getComplaint should success for owner
    ✓ getComplaint should mask anonymous owner details for other viewers
    ✓ getComplaint should not mask owner data when the owner views an anonymous complaint
    ✓ getComplaint should allow other students to view general complaints
    ✓ getComplaint should query with populate and select for nested relations
    ✓ getComplaint should handle missing user reference gracefully
    ✓ getAllComplaints should deny non-admin
    ✓ getAllComplaints should return all for admin
  Actions
    ✓ upvoteComplaint should 404
    ✓ upvoteComplaint should 403 for personal
    ✓ upvoteComplaint should toggle ON and update priority
    ✓ upvoteComplaint should set High priority with enough votes
```




## 3. notificationController

```
PS C:\Users\neel4\IT314_Project\Campus-Complaint-Management-System> npm run test

> campus-complaint-management-system@1.0.0 test
> mocha



  Notification Controller 100% Coverage Tests
    getNotifications
      ✓ Should fetch notifications and unread count successfully
      ✓ Should handle Server Errors (500)
    markNotificationRead
      ✓ Should return 404 if notification not found
      ✓ Should mark notification as read successfully
      ✓ Should handle Server Errors (500)
    markAllRead
      ✓ Should update all notifications successfully
      ✓ Should handle Server Errors (500)
    deleteNotification
      ✓ Should return 404 if notification not found
      ✓ Should delete notification and return new count
      ✓ Should handle Server Errors (500)


  10 passing (75ms)

PS C:\Users\neel4\IT314_Project\Campus-Complaint-Management-System>
```

## 4. reportController

```
PS C:\Users\neel4\IT314_Project\Campus-Complaint-Management-System> npm run test

> campus-complaint-management-system@1.0.0 test
> mocha



  Report Controller Unit Tests
    ✓ Should return 400 if committeeType is missing in query
    ✓ Should return 200 if committee type is invalid (Resolve returns null)
    ✓ Should generate PDF report successfully with analytics (92ms)
    ✓ Should handle database errors gracefully
    ✓ Should generate report with empty complaints array
    ✓ Should query complaints using mapped category and 30-day lookback
    ✓ Should compute status and priority aggregates accurately
    ✓ Should include report metadata text with underline header
    ✓ Should include committee metadata and section headers
    ✓ Should list only the first 10 recent complaints with numbering


  10 passing (2s)

PS C:\Users\neel4\IT314_Project\Campus-Complaint-Management-System>
```

# Models

**All files backend/models**

**100%** Statements `213/213` **100%** Branches `10/10` **100%** Functions `2/2` **100%** Lines `213/213`

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 | |
|---|---|---|---|---|---|---|---|---|
| Complaint.js | 100% | 103/103 | 100% | 1/1 | 100% | 0/0 | 100% | |
| Notification.js | 100% | 32/32 | 100% | 1/1 | 100% | 0/0 | 100% | |
| otpModel.js | 100% | 21/21 | 100% | 1/1 | 100% | 0/0 | 100% | |
| userModel.js | 100% | 57/57 | 100% | 7/7 | 100% | 2/2 | 100% | |

```
Complaint Model Unit Tests
  ✓ enforces required fields
  ✓ validates enum guarded fields and nested status history
  ✓ applies safe defaults for optional fields
  ✓ preserves status history metadata with timestamps and updater
  ✓ exposes the canonical category whitelist used for routing
  ✓ keeps priority and status enums synchronized with business logic
  ✓ stores automatic timestamps for auditing

Notification Model Unit Tests
  ✓ requires a user reference and message body
  ✓ applies defaults for optional fields
  ✓ rejects unsupported notification types
  ✓ documents the permitted notification type enum
  ✓ keeps relational references wired to User and Complaint models

OTP Model Unit Tests
  ✓ requires lowercase email and otp string
  ✓ sets expiration metadata on createdAt

User Model Unit Tests
  Validation Logic
    ✓ Should require committeeType if role is "committee"
    ✓ Should NOT require committeeType if role is "student"
  Schema metadata
    ✓ lists all allowed committee types for committee accounts
    ✓ locks the role enum to supported portals
    ✓ keeps base identity fields required and unique where needed
  matchPassword Method
    ✓ Should verify correct password using bcrypt
  Pre-save Hook (Password Hashing)
    ✓ Should hash password if it is modified
    ✓ Should NOT hash password if it is NOT modified
    ✓ Should handle hashing errors


23 passing (77ms)
```

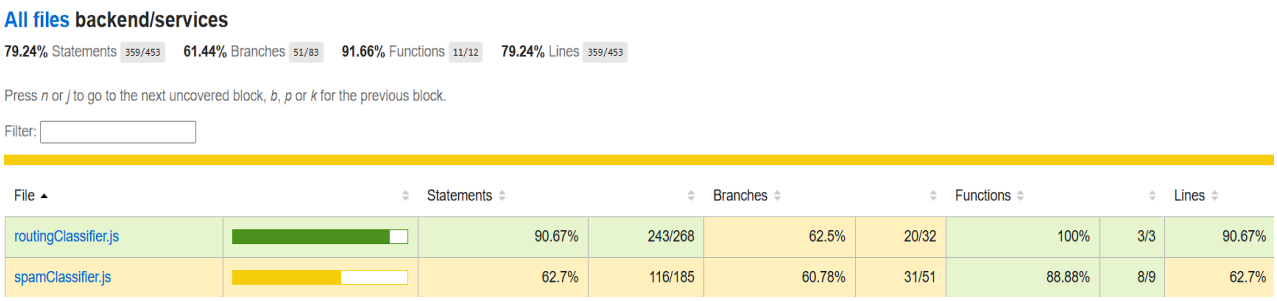
```
------------------|---------|----------|---------|---------|-------------------
File              | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------------|---------|----------|---------|---------|-------------------
All files         |     100 |      100 |     100 |     100 |
 Complaint.js     |     100 |      100 |     100 |     100 |
 Notification.js  |     100 |      100 |     100 |     100 |
 otpModel.js      |     100 |      100 |     100 |     100 |
 userModel.js     |     100 |      100 |     100 |     100 |
------------------|---------|----------|---------|---------|-------------------
PS C:\Users\neel4\IT314_Project\Campus-Complaint-Management-System>
```

# Middleware

| File ▲ | | Statements | | Branches | | Functions | | Lines |
|---|---|---|---|---|---|---|---|---|
| authMiddleware.js | | 100% | 49/49 | 100% | 10/10 | 100% | 2/2 | 100% |

```
Auth Middleware Unit Tests
  protect
    ✓ Should attach user to request and call next for valid Bearer token
    ✓ Should return 401 when token verification fails
    ✓ Should return 401 when Authorization header is missing
    ✓ Should return 401 when Authorization header is not Bearer scheme
  authorize
    ✓ Should allow request when user role is in allowed roles
    ✓ Should return 403 when user is missing
    ✓ Should return 403 when user role is not allowed


7 passing (35ms)
```

# Services




Reason for Lower Coverage : *Probabilistic, pattern-based logic and Unused Safeguards.*

- Classifiers rely on probabilities and pattern matching. Achieving 100% coverage would mean manually triggering every keyword match and every possible edge-case scoring scenario, which would require an unrealistic amount of test data.

- Defensive Code (Dead Code): These services include safety checks (for example, if (!text) return null). But in our actual flow, the controllers already validate inputs before calling the service. Because valid data is always sent during integration, these error-handling lines never get executed in tests resulting in lower coverage numbers.

# Utils

## All files backend/utils

**74.01%** Statements `943/1274` **65%** Branches `143/220` **85.71%** Functions `24/28` **74.01%** Lines `943/1274`

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 | |
|---|---|---|---|---|---|---|---|---|
| aiRouting.js | 69.12% | 741/1072 | 60.21% | 112/186 | 90% | 18/20 | 69.12% | |
| emailService.js | 100% | 105/105 | 83.33% | 15/18 | 100% | 3/3 | 100% | |
| fileUploadService.js | 100% | 82/82 | 100% | 14/14 | 50% | 2/4 | 100% | |
| generateToken.js | 100% | 15/15 | 100% | 2/2 | 100% | 1/1 | 100% | |

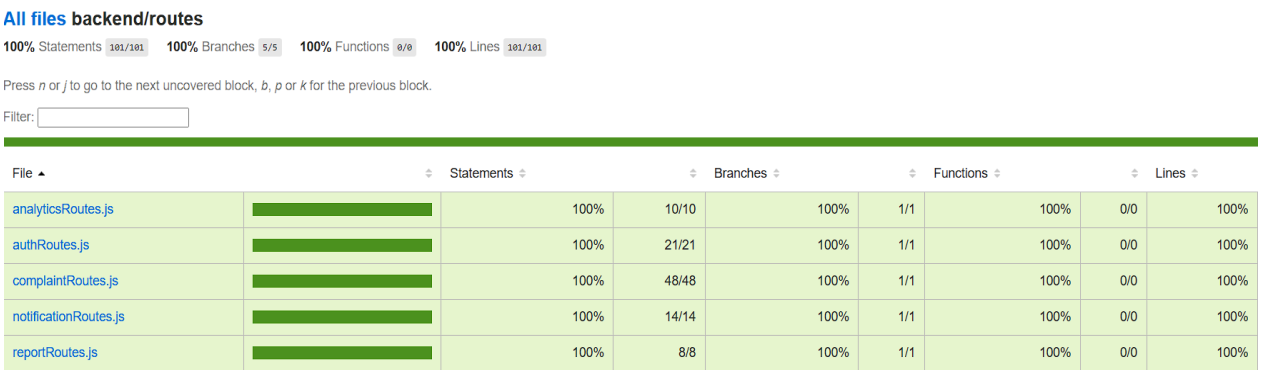
```
  Email Service Unit Tests
    Configuration Validation
      ✓ Should fail if EMAIL_USER is missing
      ✓ Should fail if EMAIL_PASS is missing
    Transporter Creation Logic
      ✓ Should create Gmail transporter if email is @gmail.com
      ✓ Should create Custom SMTP transporter if email is NOT @gmail.com
    sendOTPEmail
      ✓ Should send email successfully
      ✓ Should handle send errors
    sendStatusUpdateEmail
      ✓ Should send status update email successfully
      ✓ Should handle status update errors

  File Upload Service Unit Tests
    ✓ Should configure disk storage in E2E mode and enforce fileFilter rules (891ms)
    ✓ Should configure Cloudinary storage when not in E2E mode and credentials are present (72
ms)
CRITICAL ERROR: Cloudinary credentials are not configured.
    ✓ Should throw a clear error when Cloudinary credentials are missing (38ms)

  Generate Token Utility - 100% Coverage
    generateToken
      ✓ Should generate token with correct payload and options
      ✓ Should handle string userId
      ✓ Should handle numeric userId
      ✓ Should handle null userId
      ✓ Should handle undefined userId
      ✓ Should use JWT_SECRET from environment
      ✓ Should set expiration to 30 days
      ✓ Should handle jwt.sign throwing an error
      ✓ Should return the exact token from jwt.sign
      ✓ Should handle empty string userId
      ✓ Should handle object userId


  22 passing (1s)
```

```
backend/utils     |   74.01 |       65 |   85.71 |   74.01 |
 aiRouting.js     |   69.12 |    60.21 |      90 |   69.12 | ...1038,1040-1048
 emailService.js  |     100 |    83.33 |     100 |     100 | 23-24,78
 ...oadService.js |     100 |      100 |      50 |     100 |
 generateToken.js |     100 |      100 |     100 |     100 |
```

Reason for Lower Coverage of aiRouting.js : *External Process Dependency and Test Flags.*

- Python Wrapper Issue (Skipped Logic): This file is meant to start an external Python process. But running a Python-based ML model inside a fast JavaScript unit test suite would slow everything down.
- E2E_MODE Flag: To avoid this slowdown, tests usually set an environment variable (like E2E_MODE=true) or mock the function. When this flag is enabled, the code completely skips the part that launches the Python process and reads its output.
- Because of this, the coverage tool sees that the entire "Python Execution Block" (which is a major portion of the file) is never executed. This leads to a noticeably lower coverage percentage. The logic still exists in the codebase, but it is intentionally bypassed during testing to keep the test suite fast and stable.

# Routes



**All files backend/routes**

**100%** Statements 101/101 **100%** Branches 5/5 **100%** Functions 0/0 **100%** Lines 101/101

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 | |
|---|---|---|---|---|---|---|---|---|
| analyticsRoutes.js | 100% | 10/10 | 100% | 1/1 | 100% | 0/0 | 100% | |
| authRoutes.js | 100% | 21/21 | 100% | 1/1 | 100% | 0/0 | 100% | |
| complaintRoutes.js | 100% | 48/48 | 100% | 1/1 | 100% | 0/0 | 100% | |
| notificationRoutes.js | 100% | 14/14 | 100% | 1/1 | 100% | 0/0 | 100% | |
| reportRoutes.js | 100% | 8/8 | 100% | 1/1 | 100% | 0/0 | 100% | |

Report Routes Wiring
  ✓ registers the committee monthly report endpoint

Notification Routes Wiring
  ✓ enforces protection on all downstream routes
  ✓ maps HTTP verbs to controller handlers

Complaint Routes Wiring
  ✓ exports the configured router
  ✓ registers upload-enabled create endpoints
  ✓ protects list and insight endpoints
  ✓ registers mutation endpoints for complaints
  ✓ covers delete and detail endpoints

Auth Routes Wiring
  ✓ exposes a configured router instance
  ✓ registers public auth endpoints without guard middleware
  ✓ protects password change and profile routes
  ✓ protects admin-only management endpoints

Analytics Routes Wiring
  ✓ guards committee analytics endpoint by committee type

| File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s |
|---|---|---|---|---|---|
| All files | 87.83 | 78.83 | 93.42 | 87.83 | |
| backend | 96.02 | 78.12 | 100 | 96.02 | |
| server.js | 96.02 | 78.12 | 100 | 96.02 | ...53-154,168-169 |
| ...nd/controllers | 97.87 | 91.07 | 100 | 97.87 | |
| ...Controller.js | 100 | 100 | 100 | 100 | |
| ...Controller.js | 96.63 | 85.2 | 100 | 96.63 | ...20-425,436-440 |
| ...Controller.js | 100 | 100 | 100 | 100 | |
| ...Controller.js | 100 | 100 | 100 | 100 | |
| ...nd/middlewares | 100 | 100 | 100 | 100 | |
| ...Middleware.js | 100 | 100 | 100 | 100 | |
| backend/models | 100 | 100 | 100 | 100 | |
| Complaint.js | 100 | 100 | 100 | 100 | |
| Notification.js | 100 | 100 | 100 | 100 | |
| otpModel.js | 100 | 100 | 100 | 100 | |
| userModel.js | 100 | 100 | 100 | 100 | |
| backend/routes | 100 | 100 | 100 | 100 | |
| ...ticsRoutes.js | 100 | 100 | 100 | 100 | |
| authRoutes.js | 100 | 100 | 100 | 100 | |
| ...aintRoutes.js | 100 | 100 | 100 | 100 | |
| ...tionRoutes.js | 100 | 100 | 100 | 100 | |
| reportRoutes.js | 100 | 100 | 100 | 100 | |
| backend/services | 79.24 | 61.44 | 91.66 | 79.24 | |
| ...Classifier.js | 90.67 | 62.5 | 100 | 90.67 | ...37-240,242-244 |
| ...Classifier.js | 62.7 | 60.78 | 88.88 | 62.7 | ...52-160,176-177 |
| backend/utils | 74.01 | 65 | 85.71 | 74.01 | |
| aiRouting.js | 69.12 | 60.21 | 90 | 69.12 | ...1038,1040-1048 |
| emailService.js | 100 | 83.33 | 100 | 100 | 23-24,78 |
| ...oadService.js | 100 | 100 | 50 | 100 | |
| generateToken.js | 100 | 100 | 100 | 100 | |