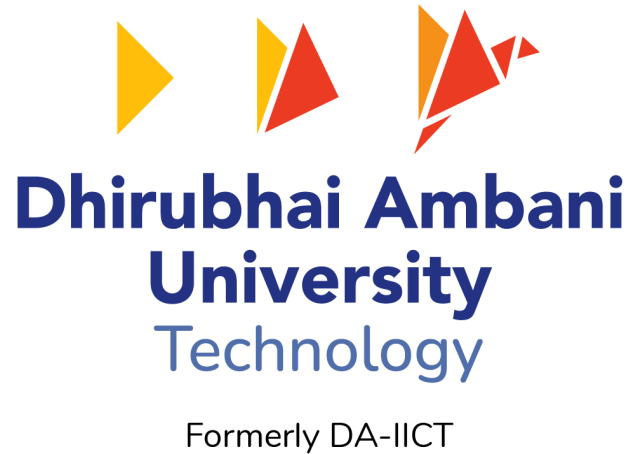


IT314 - Software Engineering

Unit Testing - G34



Group Members:

Student ID	Name
202301462	JOSHI YESHA SNEHAL
202301216	PATEL DIYA MUKESHKUMAR
202301475	PARMAR VIVEK MAHESHBHAI
202301432	JAS MEHTA
202301438	MAANAV GURUBAXANI
202301252	PRASHAM SHAH
202301478	PRAJAPATI SUJALKUMAR NAYANBHAI
202301470	MAHERIYA HARSH PRAKASHBHAI
202301440	VRAJ RONA K PARIKH
202301487	YUG TEJANI

PROFESSOR : DR. Saurabh Tiwari

Group Mentor : Kashvi Bhanderi

Group Leader : Maheriya Harsh Prakashbhai

1.Introduction

Unit testing is a software quality assurance practice in which individual components of the codebase are tested in isolation to verify that each unit functions as intended. By validating small, discrete pieces of functionality early in the development process, unit tests help identify defects quickly, reduce integration issues, and ensure long-term code stability.

This report summarizes the unit tests executed, their coverage, results, and any identified issues, providing a clear view of the system's reliability at the component level.

2.What is Unit Test?

- A unit test is a small piece of code that checks if a specific function or method in an application works correctly.
- It will work as the function inputs and verifying the outputs.
- These tests check that the code works as expected based on the logic the developer intended.
- We have used Jest as the test runner.

3.How to run unit test

- Run full test suite:
 - `npm test`
 - `npx jest`
- Run a specific test file:
 - `npx jest src/tests/some_test_file.test.ts`
- Run coverage:
 - `npm run test:coverage`
- Helpful flags:
 - `--runInBand` (serial)
 - `--detectOpenHandles` (diagnose hanging async operations)

Game Controllers and Middleware:

1. getGames

```
PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere> npm test -- src/tests/gameControllers_test/getGames.test.ts

> sportssphere-backend@1.0.0 test
> jest --detectOpenHandles --runInBand src/tests/gameControllers_test/getGames.test.ts

PASS src/tests/gameControllers_test/getGames.test.ts
Game Controllers - getGames.ts
  getGameById
    ✓ should return game by ID with populated fields and calendarLink as host is requesting (14 ms)
    ✓ should call next with AppError when game not found (7 ms)
    ✓ should NOT expose calendar link if booking not paid (9 ms)
    ✓ should expose calendar link if requester is approved player (6 ms)
    ✓ should NOT expose calendar link if user is not host or approved (5 ms)
    ✓ should call all populate chains with correct parameters (3 ms)
  getMyBookings
    ✓ should return all booking categories for logged-in user (8 ms)
    ✓ should return empty arrays when user has no bookings (3 ms)
    ✓ should include a cancelled game where user is ONLY an approvedPlayer (4 ms)
  getGames
    ✓ should return all open upcoming not-full games with no filters (7 ms)
    ✓ should include game when approvedPlayers < max (8 ms)
    ✓ should exclude game when approvedPlayers = max (11 ms)
    ✓ should not crash when playersNeeded.max is missing (5 ms)
    ✓ should filter by sport (case-insensitive) (5 ms)
    ✓ should filter by city (case-insensitive) (5 ms)
    ✓ should filter by venueName (case-insensitive) (5 ms)
    ✓ should filter by date range (startDate and endDate) (4 ms)
    ✓ should filter by startDate only and ensure upcoming games (4 ms)
    ✓ should filter by endDate only and ensure not past games (3 ms)
    ✓ should ensure only upcoming games when no date filters (7 ms)
    ✓ should filter by minPrice only (7 ms)
    ✓ should filter by maxPrice only (6 ms)
    ✓ should filter by price range (minPrice and maxPrice) (4 ms)
    ✓ should handle invalid price values gracefully (4 ms)
    ✓ should handle valid minPrice and invalid maxPrice (3 ms)
```

```

✓ should handle minPrice greater than maxPrice (2 ms)
✓ should filter by geolocation with default radius (3 ms)
✓ should filter by geolocation with custom radius (5 ms)
✓ should handle combined filters (sport, price, date) (4 ms)
✓ should ignore sport filter if not a string (3 ms)
✓ should ignore city filter if not a string (8 ms)
✓ should ignore venueName filter if not a string (2 ms)
✓ should not apply geo filter if lng is missing (7 ms)
✓ should not apply geo filter if lat is missing (2 ms)
✓ should not apply geo filter if both is missing (1 ms)
✓ should default radius to 5000 when radius <= 0 (1 ms)
✓ should default radius to 5000 when radius is negative (2 ms)
✓ should return empty games array when no games match filters (3 ms)
✓ should populate host and approvedPlayers fields (3 ms)
✓ should sort games by slot.startTime in ascending order (2 ms)
✓ should ignore empty string filters for sport, city, venueName (2 ms)

```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
getGames.ts	100	100	100	100	

Test Suites: 1 passed, 1 total

Tests: 41 passed, 41 total

Snapshots: 0 total

Time: 3.102 s

Ran all test suites matching /src\\tests\\gameControllers_test\\getGames.test.ts/i.

2. hostGame

Ran all test suites matching /src\\tests\\gameControllers_test\\getGames.test.ts/i.

PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere> npm test -- src/tests/gameControllers_test/hostGame.test.ts

> sportosphere-backend@1.0.0 test

> jest --detectOpenHandles --runInBand src/tests/gameControllers_test/hostGame.test.ts

PASS src/tests/gameControllers_test/hostGame.test.ts

Game Controllers - hostGame.ts

hostGame

```

✓ should successfully host a game with valid inputs (14 ms)
✓ should throw error when sport field is missing (4 ms)
✓ should throw error when venueId field is missing (5 ms)
✓ should throw error when subVenueId field is missing (8 ms)
✓ should throw error when timeSlotDocId field is missing (4 ms)
✓ should throw error when slotId field is missing (3 ms)
✓ should throw error when playersNeeded is missing (5 ms)
✓ should throw error when playersNeeded.min is missing (11 ms)
✓ should throw error when playersNeeded.max is missing (8 ms)
✓ should throw error when min players exceeds max players (4 ms)
✓ should throw error when venue is not found (5 ms)
✓ should throw error when subVenue is not found (5 ms)
✓ should throw error when sport is not available in subVenue (3 ms)
✓ should throw error when slot does not have valid price for sport (13 ms)
✓ should handle slot prices as object instead of Map (4 ms)
✓ should calculate approxCostPerPlayer correctly (7 ms)

```

cancelGame

```

✓ should successfully cancel a game when conditions are met (4 ms)
✓ should throw error when game is not found (2 ms)
✓ should throw error when non-host tries to cancel (4 ms)
✓ should throw error when trying to cancel booked game (4 ms)
✓ should throw error when trying to cancel less than 2 hours before start (8 ms)
✓ should allow cancellation exactly 2 hours before start (5 ms)

```

leaveGame

```

✓ should successfully leave a game when user is approved (4 ms)
✓ should throw error when game is not found (3 ms)

```

```

    ✓ should throw error when slot is already booked (9 ms)
    ✓ should throw error when user is not approved for the game (4 ms)
    ✓ should reopen game when status was Full and becomes not full (5 ms)
    ✓ should not change status if game was not Full (3 ms)
    ✓ should abort transaction and end session on error (4 ms)

-----|-----|-----|-----|-----|-----
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100 |    100   |    100   |    100   |
  hostGame.ts |    100 |    100   |    100   |    100   |
-----|-----|-----|-----|-----|-----
Test Suites: 1 passed, 1 total
Tests:       29 passed, 29 total
Snapshots:   0 total
Time:        2.583 s, estimated 4 s

```

3. joinRequestController

```

PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere> npm test -- src/tests/gameControllers_test/joinRequestController.test.ts

> sportsphere-backend@1.0.0 test
> jest --detectOpenHandles --runInBand src/tests/gameControllers_test/joinRequestController.test.ts

PASS src/tests/gameControllers_test/joinRequestController.test.ts
  Join Request Controllers - joinRequestController.ts
    createJoinRequest
      ✓ should successfully create a join request (11 ms)
      ✓ should throw error when game is not found (6 ms)
      ✓ should throw error when host tries to join their own game (3 ms)
      ✓ should throw error when game is not open (2 ms)
      ✓ should throw error when game has already started (4 ms)
      ✓ should throw error when join request is already pending (4 ms)
      ✓ should throw error when player is already approved (5 ms)
      ✓ should throw error when join request was previously rejected (5 ms)
      ✓ should check for time overlap before creating join request (8 ms)
    approveJoinRequest
      ✓ should successfully approve a join request (16 ms)
      ✓ should throw error when game is not found (3 ms)
      ✓ should throw error when non-host tries to approve (2 ms)
      ✓ should throw error when game is not open (5 ms)
      ✓ should throw error when game has already started (6 ms)
      ✓ should throw error when join request is not found (5 ms)
      ✓ should throw error when join request is not pending (4 ms)
      ✓ should throw error when game is already full (4 ms)
      ✓ should change game status to Full when capacity reached (3 ms)
      ✓ should not add player twice if already in approvedPlayers (4 ms)
      ✓ should check for time overlap for the player being approved (2 ms)

```

```
rejectJoinRequest
  ✓ should successfully reject a join request (3 ms)
  ✓ should throw error when game is not found (3 ms)
  ✓ should throw error when non-host tries to reject (5 ms)
  ✓ should throw error when game is not open (3 ms)
  ✓ should throw error when join request is not found (2 ms)
  ✓ should throw error when join request is not pending (2 ms)
cancelJoinRequest
  ✓ should successfully cancel a join request (8 ms)
  ✓ should throw error when game is not found (3 ms)
  ✓ should throw error when no join request found (2 ms)
  ✓ should throw error when trying to cancel approved request (11 ms)
  ✓ should throw error when trying to cancel rejected request (3 ms)
  ✓ should throw error when cancelling less than 2 hours before game start (4 ms)
  ✓ should allow cancellation exactly 2 hours before game start (5 ms)
  ✓ should remove join request from joinRequests array (4 ms)

-----|-----|-----|-----|-----|-----
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100 |    100   |    100   |    100   |
joinRequestController.ts |    100 |    100   |    100   |    100   |
-----|-----|-----|-----|-----|-----
Test Suites: 1 passed, 1 total
Tests:       34 passed, 34 total
Snapshots:   0 total
Time:        2.342 s, estimated 4 s
Ran all test suites matching /src\\tests\\gameControllers_test\\joinRequestController.test.ts/i.
```

4. postGame

```

gameControllers/postGame.ts" src/tests/gameControllers_test/postGame.test.ts
PASS src/tests/gameControllers_test/postGame.test.ts
Post Game Controllers - postGame.ts
  rateVenueAfterGame
    ✓ should successfully rate venue by host (10 ms)
    ✓ should successfully rate venue by approved player (5 ms)
    ✓ should throw error when rating is missing (3 ms)
    ✓ should throw error when rating is less than 1 (3 ms)
    ✓ should throw error when rating is greater than 5 (2 ms)
    ✓ should throw error when game is not found (11 ms)
    ✓ should throw error when game is not completed (4 ms)
    ✓ should throw error when user is not authorized to rate (3 ms)
    ✓ should throw error when venue is not found (7 ms)
    ✓ should update existing rating when user has already rated (8 ms)
    ✓ should correctly calculate average rating with multiple ratings (3 ms)
  completeGame
    ✓ should successfully complete game by host (3 ms)
    ✓ should return 404 when game is not found (3 ms)
    ✓ should return 403 when non-host tries to complete game (10 ms)
    ✓ should return 400 when game has not ended yet (5 ms)
    ✓ should allow completion when current time equals end time (3 ms)
    ✓ should return void explicitly (5 ms)

-----|-----|-----|-----|-----|-----
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100 |    100 |    100 |    100 |
postGame.ts |    100 |    100 |    100 |    100 |
-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       17 passed, 17 total
Snapshots:   0 total
Time:        3.28 s
Ran all test suites matching /src\\tests\\gameControllers_test\\postGame.test.ts/i.
❖ PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere>

```

Middleware

1. gameMiddleware

```
Ran all test suites.
PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere> npm test -- --coverage --collectCoverageFrom="src/middleware/gameMiddleware.ts" src/tests/middleware_test/gameMiddleware.test.ts
> jest --detectOpenHandles --runInBand --coverage --collectCoverageFrom=src/middleware/gameMiddleware.ts src/tests/middleware_test/gameMiddleware.test.ts

PASS src/tests/middleware_test/gameMiddleware.test.ts
  authorizeRoles() authorizeRoles method
    Happy Paths
      ✓ should call next() when user role is authorized (single role) (8 ms)
      ✓ should call next() when user role is among multiple allowed roles (2 ms)
      ✓ should call next() when user role is at the end of allowed roles array (2 ms)
    Edge Cases
      ✓ should return 403 when user role is not authorized (3 ms)
      ✓ should return 403 when allowed roles array is empty (3 ms)
      ✓ should return 403 when user role is an empty string (3 ms)
      ✓ should return 403 when user role is undefined (4 ms)
      ✓ should return 403 when req.user is missing (3 ms)
      ✓ should return 403 when req.user.role is null (4 ms)
      ✓ should call next() when allowed roles contains an empty string and user role is empty string (2 ms)
      ✓ should call next() when allowed roles contains undefined and user role is undefined (3 ms)
      ✓ should call next() when allowed roles contains null and user role is null (3 ms)

-----|-----|-----|-----|-----|-----
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100 |     100 |     100 |     100 |
gameMiddleware.ts |    100 |     100 |     100 |     100 |
-----|-----|-----|-----|-----|-----
Test Suites: 1 passed, 1 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        1.315 s
Ran all test suites matching /src\\tests\\middleware_test\\gameMiddleware.test.ts/i.
❖ PS C:\Users\M.P\OneDrive\Desktop\SE\SportSphere>
```

Admin Controllers:

1. adminAuthController

```
PASS src/tests/adminControllers_test/adminAuthController.test.ts
Admin Controllers - adminAuthController.ts
  Happy Paths
    ✓ should successfully login and return token + user when env credentials are correct (9 ms)
    ✓ should default username to "Admin" when ADMIN_USERNAME is not set (2 ms)
  Edge Cases & Error Handling
    ✓ should return 500 when ADMIN_EMAIL or ADMIN_PASSWORD are not configured (25 ms)
    ✓ should return 401 when credentials are invalid (wrong password) (2 ms)
    ✓ should return 401 when email or password is missing in request body (2 ms)
    ✓ should return 500 when JWT_SECRET is not set even if credentials are correct (5 ms)
    ✓ should handle unexpected errors and return 500 with "Server error" (16 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        0.604 s, estimated 1 s
Ran all test suites matching /src\\tests\\adminControllers_test\\adminAuthController.test.ts/i.
```

2. adminManagementController

```

PS C:\Users\lenovo\Desktop\software\SportSphere> npm test -- src/tests/adminControllers_test/adminManagementController.
> sportssphere-backend@1.0.0 test
> jest --detectOpenHandles --runInBand src/tests/adminControllers_test/adminManagementController.test.ts --no-coverage

PASS src/tests/adminControllers_test/adminManagementController.test.ts
Admin Controllers - adminManagementController.ts
  Happy Paths - Users
    ✓ should list users with pagination and search (6 ms)
    ✓ should get single user by id (1 ms)
    ✓ should delete user when exists (1 ms)
  Edge Cases & Error Handling - Users
    ✓ should return 404 when user not found in getUserById
    ✓ should return 404 when deleting non-existing user (1 ms)
  Happy Paths - Coaches
    ✓ should list coaches with details (1 ms)
    ✓ should delete coach and related coach detail (1 ms)
  Edge Cases & Error Handling - Coaches
    ✓ should return 404 when deleting non-coach or missing user (1 ms)
  Happy Paths - Venue Owners
    ✓ should list venue owners with their venues (2 ms)
    ✓ should delete venue owner when exists and role correct (1 ms)
  Edge Cases & Error Handling - Venue Owners
    ✓ should return 404 when deleting non-existing venue owner
    ✓ should return 404 when user role is not venue-owner (1 ms)
  Happy Paths - Overview Stats
    ✓ should return overview stats for users, owners, coaches and tickets (1 ms)
  Mutation Fixes - Deep Logic Coverage
    ✓ should use default pagination values (skip 0, limit 50) when query is empty (1 ms)
    ✓ should search users with case-insensitive Regex (1 ms)
    ✓ should warn but not fail if deleting coach detail throws error (2 ms)
    ✓ should return 404 if user exists but is NOT a coach (1 ms)
    ✓ should return 404 if user exists but is NOT a venue-owner (1 ms)
    ✓ should use default pagination for listCoaches (1 ms)
    ✓ should query statistics with exact filters (1 ms)

Test Suites: 1 passed, 1 total
Tests:       20 passed, 20 total
Snapshots:   0 total
Time:        1.333 s
Ran all test suites matching /src\\tests\\adminControllers_test\\adminManagementController.test.ts/i.
PS C:\Users\lenovo\Desktop\software\SportSphere>

```

3. ticketController

PASS src/tests/adminControllers_test/ticketController.test.ts

Ticket Controllers - ticketController.ts

Happy Paths - createTicket

- ✓ should create a ticket with uploaded files and return 201 (5 ms)
- ✓ should attach userName and userEmail from req.user if not provided (2 ms)

Edge Cases & Error Handling - createTicket

- ✓ should return 400 when required fields are missing (2 ms)
- ✓ should cleanup temp files and return 500 when Cloudinary upload fails (24 ms)
- ✓ should handle mongoose ValidationError from Ticket.create with 400 (5 ms)
- ✓ should return 500 on generic error from Ticket.create and cleanup temp files (6 ms)

Happy Paths - getMyTickets

- ✓ should return tickets for authenticated user (2 ms)

Edge Cases & Error Handling - getMyTickets

- ✓ should return 401 when user is not authenticated (1 ms)
- ✓ should return 500 on unexpected error (6 ms)

Happy Paths - getTicketById

- ✓ should return ticket when user owns it (3 ms)
- ✓ should return ticket when email matches userEmail (1 ms)

Edge Cases & Error Handling - getTicketById

- ✓ should return 400 for invalid ObjectId (1 ms)
- ✓ should return 404 when ticket not found (1 ms)
- ✓ should return 401 when no user on request (1 ms)
- ✓ should return 403 when user is not owner and email does not match (2 ms)
- ✓ should return 500 on unexpected error (7 ms)

Happy Paths - getAllTickets

- ✓ should return all tickets (2 ms)

Edge Cases & Error Handling - getAllTickets

- ✓ should return 500 on error (15 ms)

Happy Paths - replyToTicket

- ✓ should add reply and set status to Replied (2 ms)

Edge Cases & Error Handling - replyToTicket

- ✓ should return 400 when message is missing or empty (1 ms)
- ✓ should return 400 for invalid ticket id (1 ms)
- ✓ should return 404 when ticket not found (1 ms)
- ✓ should return 500 on unexpected error (4 ms)

Happy Paths - closeTicket

- ✓ should close ticket successfully (1 ms)

Edge Cases & Error Handling - closeTicket

- ✓ should return 400 for invalid ticket id (2 ms)
- ✓ should return 404 when ticket not found (1 ms)
- ✓ should return 500 on unexpected error (9 ms)

Mutation Fixes - Deep Logic Coverage

- ✓ should fail specifically if ONLY subject is missing (2 ms)
- ✓ should fail specifically if ONLY category is missing (1 ms)
- ✓ should correctly identify PDF and DOC file types (1 ms)
- ✓ should log error but continue if temp file cleanup fails during error handling (3 ms)
- ✓ should match userEmail case-insensitively in getMyTickets (2 ms)
- ✓ should correctly handle tickets created by deleted users (ticket.user is null) (1 ms)
- ✓ should default author to "admin" if req.user is missing during reply (1 ms)

Test Suites: 1 passed, 1 total

Tests: 34 passed, 34 total

Snapshots: 0 total

Time: 1.324 s, estimated 2 s

Ran all test suites matching /src\\tests\\adminControllers_test\\ticketController.test.ts/i.

Coverage:

```
PASS src/tests/adminControllers_test/adminManagementController.test.ts
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	99.15	93.33	100	99.06	
adminAuthController.ts	100	100	100	100	
adminManagementController.ts	100	88.88	100	100	99
ticketController.ts	98.54	93.44	100	98.27	47-48

Test Suites: 3 passed, 3 total
Tests: 61 passed, 61 total
Snapshots: 0 total
Time: 2.423 s
Ran all test suites matching /src\\tests\\adminControllers_test/i.

Booking Controllers

1. getCalendarLink

```
PASS src/tests/booking_test/getCalendarLink.test.ts
```

getCalendarLink Controller

- ✓ should return calendar link if user owns the booking (6 ms)
- ✓ should throw 404 error when booking is not found (2 ms)
- ✓ should throw 403 error when user does not own the booking (5 ms)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	0	0	0	0	

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 0.866 s, estimated 2 s
Ran all test suites matching /src\\tests\\booking_test\\getCalendarLink.test.ts/i.

Time Slot Controller

Generate Timeslots, getSlotsForSubVenueDate, UpdateSlotById, deleteSlotForDate

PASS src/tests/timeslot_test/timeslot_Controller.test.ts (48.023 s)

timeslotController.generateTimeSlots

- ✓ returns 400 if subVenue or date missing (27 ms)
- ✓ returns 400 when only subVenue missing (1 ms)
- ✓ returns 400 invalid date (3 ms)
- ✓ 400 invalid date - trailing characters (2 ms)
- ✓ returns 400 invalid subVenue id (2 ms)
- ✓ returns 404 when subVenue not found (2 ms)
- ✓ returns 400 when no sports (2 ms)
- ✓ returns 409 when slots exist (1 ms)
- ✓ creates 24 slots successfully (3 ms)
- ✓ each generated slot contains required fields (4 ms)
- ✓ creates correct startTime and endTime for first slot (hour 0) (2 ms)
- ✓ returns 500 on create error (1 ms)

timeslotController.getSlotsForSubVenueDate

- ✓ 400 when date missing (1 ms)
- ✓ 400 invalid date - leading characters (3 ms)
- ✓ 400 invalid date (1 ms)
- ✓ returns empty when not found (1 ms)
- ✓ returns timeSlot when found (2 ms)
- ✓ returns 500 on DB error (1 ms)

timeslotController.updateSlotById

- ✓ 400 invalid slotId (1 ms)
- ✓ 400 cannot modify startTime (1 ms)
- ✓ 404 parent not found (1 ms)
- ✓ 400 invalid subVenue (2 ms)
- ✓ 400 invalid sport pricing (3 ms)
- ✓ 400 negative price (3 ms)
- ✓ 400 invalid status (2 ms)
- ✓ 400 available without prices (2 ms)
- ✓ 400 invalid bookedForSport (3 ms)
- ✓ 200 block clears booking (2 ms)
- ✓ 200 successful update (2 ms)
- ✓ 500 when findOneAndUpdate returns null (1 ms)
- ✓ 500 DB error (2 ms)

timeslotController.deleteSlotsForDate

- ✓ 400 invalid date (2 ms)
- ✓ 404 nothing deleted (2 ms)
- ✓ 200 deletion success (1 ms)
- ✓ 500 DB error (2 ms)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	97.56	100	100	
...troller.ts	100	97.56	100	100	259

```
-----|-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:      35 passed, 35 total
Snapshots:  0 total
Time:       53.713 s
```

Venue and Subvenue controller

Createvenue, getvenue, getmyvenue, getvenuebyId, updatvenue, delteVenue, createsubvenue, getsubvenuebyvenue, updatesubvenue, deletesubvenue, ratevenue, updatrating

```
✓ handles internal error with 500 (2 ms)
rateVenue
✓ returns 400 if userId or rating missing (3 ms)
✓ returns 400 if rating out of range (3 ms)
✓ returns 404 if venue not found (1 ms)
✓ adds a new rating when user hasn't rated before (2 ms)
✓ updates existing user's rating (2 ms)
✓ handles internal error with 500 (2 ms)
getVenueRatings
✓ returns 404 if venue not found (1 ms)
✓ returns ratings when venue exists (2 ms)
✓ handles internal error with 500 (1 ms)
```

```

PASS src/tests/venue_test/venue_Controller.test.ts (84.033 s)
Venue Controller - Unit tests (combined)
  createVenue
    ✓ returns 401 if req.user is missing (unauthorized) (43 ms)
    ✓ creates a venue with image uploads and amenities string (7 ms)
    ✓ creates a venue with no files and no amenities (empty arrays) (4 ms)
    ✓ handles unexpected error and returns 500 (2 ms)
  getVenues
    ✓ returns venues list on success (2 ms)
    ✓ returns 500 if Venue.find throws (7 ms)
  getMyVenues
    ✓ returns 401 if no user (1 ms)
    ✓ returns venues and calendarLink when booking exists (4 ms)
    ✓ returns venues and null calendarLink when booking does not exist (1 ms)
    ✓ handles errors with 500 (8 ms)
  getVenueById
    ✓ returns 404 when not found (4 ms)
    ✓ returns 200 and venue when found (5 ms)
    ✓ handles DB error with 500 (4 ms)
  updateVenue
    ✓ updates venue with uploaded images and new location (10 ms)
    ✓ returns 404 when updating non-existing venue (2 ms)
    ✓ handles error with 500 (4 ms)
  deleteVenue
    ✓ returns 404 if venue not found (4 ms)
    ✓ deletes venue and its subvenues and returns success (4 ms)
    ✓ handles error with 500 (2 ms)
  createSubVenue
    ✓ returns 400 for invalid sports JSON string (2 ms)
    ✓ creates subvenue with parsed sports and updates venue sports (2 ms)
    ✓ handles internal error with 500 (2 ms)
  getSubVenuesByVenue
    ✓ returns subvenues array (1 ms)
    ✓ handles error with 500
  updateSubVenue
    ✓ returns 404 when subvenue not found (2 ms)
    ✓ updates subvenue and triggers venue sports update (4 ms)
    ✓ handles internal error with 500 (1 ms)
  deleteSubVenue
    ✓ returns 404 when subvenue not found (3 ms)
    ✓ deletes subvenue and updates venue sports (2 ms)
    ✓ handles internal error with 500 (1 ms)
  rateVenue
    ✓ returns 400 if userId or rating missing (3 ms)
    ✓ returns 400 if rating out of range (3 ms)
    ✓ returns 404 if venue not found (1 ms)
    ✓ adds a new rating when user hasn't rated before (2 ms)
    ✓ updates existing user's rating (2 ms)
    ✓ handles internal error with 500 (2 ms)

```

venueController.ts

98.35

87.93

94.11

98.11

```

-----|-----|-----|-----|-----|-----
Test Suites: 1 passed, 1 total
Tests:      39 passed, 39 total
Snapshots:  0 total
Time:       132.359 s

```


LLM based venue searching

```
PASS src/tests/venueSearch_test/aiVenueSearch.test.ts
aiVenueSearch Controller
  ✓ should return 400 if no question is provided (4 ms)
  ✓ should run pipeline when no $geoNear stage exists (2 ms)
  ✓ should move $geoNear to first if not at index 0 (2 ms)
  ✓ should not move $geoNear if it is already first (1 ms)
  ✓ should return success = true with correct count and data
  ✓ should return 500 if getPipelineFromNL throws (51 ms)
  ✓ should return 500 if Venue.aggregate().exec throws (5 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        6.427 s
Ran all test suites matching /src\\tests\\venueSearch_test\\aiVenueSearch.test.ts/i.
```

Venue wise Dashboard for summarization

```
PASS src/tests/venuesDashboard_test/venueDashboardController.test.ts

getVenueOwnerDashboarddashboard_test/venueDashboardController.
  ✓ should return 400 for invalid venueId (7 ms)rdController.
  ✓ should return 401 if user is not authenticated (4 ms)ler.
  ✓ should return 404 if venue not found (3 ms)ardController.
  ✓ should return 403 if user is not owner or admin (1 ms)er.
  ✓ should return 500 if Groq LLM returns invalid JSON (6 ms)
  ✓ should return 200 with computed dashboard + AI insights (happy path) (3 ms)
  ✓ should set revenueChangePercent = 0 when previous revenue is 0 (3 ms)
  ✓ should handle case when there are no bookings (6 ms)ller.
  ✓ should successfully extract JSON even when Groq returns text around JSON (2 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        5.983 s
Ran all test suites matching /src\\tests\\venuesDashboard_test\\venueDashboardControll
er.test.ts/i.
PS C:\Sujal\Software Engineering Project\Dummy4\SportSphere> 
```

Tests for the authcontrollers, authmiddleware,errorhandlermiddleware, verificationmiddleware, bookings and payments-

File	Statements	Branches	Functions	Lines
stripeWebhook.ts	<div><div></div></div> 93.42%	71/76	86.66%	26/30

authMiddleware.ts	<div><div></div></div> 97.4%	75/77	79.41%	27/34	100%	2/2	97.4%	75/77
errorHandler.ts	<div><div></div></div> 100%	12/12	90.9%	10/11	100%	1/1	100%	11/11
gameMiddleware.ts	<div><div></div></div> 100%	6/6	100%	1/1	100%	2/2	100%	5/5
multer.ts	<div><div></div></div> 92.3%	12/13	33.33%	1/3	100%	2/2	92.3%	12/13
validation.ts	<div><div></div></div> 100%	11/11	100%	2/2	100%	3/3	100%	11/11

File	Statements	Branches	Functions	Lines
bookingCleanup.ts	<div><div></div></div> 95.31%	61/64	100%	21/21

File	Statements	Branches	Functions	Lines
directBooking.ts	<div><div></div></div> 90.76%	59/65	85.71%	18/21
gameBooking.ts	<div><div></div></div> 78.18%	43/55	64.28%	9/14
getCalendarLink.ts	<div><div></div></div> 100%	10/10	100%	2/2
getMyVenueBooking.ts	<div><div></div></div> 100%	11/11	100%	22/22
retryPayment.ts	<div><div></div></div> 97.05%	66/68	89.65%	26/29
verifyPayment.ts	<div><div></div></div> 96.15%	25/26	87.5%	7/8

```
Test Suites: 24 passed, 24 total
Tests:       271 passed, 271 total
Snapshots:   0 total
Time:        12.259 s, estimated 13 s
Ran all test suites matching /src\/tests\/(auth_test|booking_test|integration|services|middleware_test|payment_test)/i.
```

Coach Controllers

1. Accept Booking request

```
(Use node --trace-warnings ... to show where the warning was created)
PASS src/tests/coach_test/acceptBookingRequest.test.ts
  acceptBookingRequest - controller unit tests
    ✓ returns failure when a non-coach attempts to accept a booking (14 ms)
    ✓ returns failure when the booking cannot be found (3 ms)
    ✓ returns failure when the booking is not in pending state (3 ms)
    ✓ returns failure when the referenced slot is missing (3 ms)
    ✓ returns failure when the slot is already booked (2 ms)
    ✓ accepts booking and marks slot as booked when all validations pass (3 ms)
    ✓ propagates database errors from CoachBooking lookup (6 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        2.747 s, estimated 4 s
Ran all test suites matching /src\\tests\\coach_test\\acceptBookingRequest.test.ts/i.
```

2. Create Coach Slot

```
lots.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "createCoachSlot - controller unit tests"
PASS src/tests/coach_test/createCoachSlots.test.ts
  createCoachSlot - controller unit tests
    ✓ returns failure when a non-coach attempts to create a slot (7 ms)
    ✓ returns failure when required slot fields are missing (1 ms)
    ✓ returns failure when a slot already exists for the same date and time (1 ms)
    ✓ creates and persists a new slot when input is valid (10 ms)
    ✓ propagates errors when CoachSlot lookup fails (3 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        7.569 s
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\createCoachSlots.test.ts/
tching "createCoachSlot - controller unit tests".
```

3. Create or update Coach details

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\createOrUpdateCoach tests matching "createOrUpdateCoachDetail".
```

```
Ran all test suites matching /d:\SE_my\SE_12\SportSphere\src\tests\coach_test\deleteCoachPhoto.test.ts
tching "deleteCoachPhoto - controller unit tests".
```

```
Ran all test suites matching /d:\SE_my\SE_12\SportSphere\src\tests\coach_test\deleteCoachSlots.test.t
atching "deleteCoachSlot - controller unit tests".
```

6. Get all coaches

```
s.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getAllCoaches - controller unit tests"
```

```
PASS src/tests/coach_test/getAllCoaches.test.ts
```

```
getAllCoaches - controller unit tests
```

- ✓ returns list of coaches with merged basic and detail information (10 ms)
- ✓ returns empty array when no coaches are found (2 ms)
- ✓ propagates errors when the user lookup fails (4 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 3 passed, 3 total
```

```
Snapshots: 0 total
```

```
Time: 8.427 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getAllCoaches.test.ts matching "getAllCoaches - controller unit tests".
```

7. Get coach bookings

```
ingRequest.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getCoachBookingRequests - controller unit tests"
```

```
PASS src/tests/coach_test/getCoachBookingRequest.test.ts
```

```
getCoachBookingRequests - controller unit tests
```

- ✓ returns failure when requester is not a coach (13 ms)
- ✓ returns populated booking requests for the coach (2 ms)
- ✓ returns an empty array when the coach has no booking requests (2 ms)
- ✓ propagates errors when CoachBooking lookup throws (5 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 4 passed, 4 total
```

```
Snapshots: 0 total
```

```
Time: 8.558 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getCoachBookingRequest.test.ts matching "getCoachBookingRequests - controller unit tests".
```

8. Get Coach Profile

```
ile.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getCoachProfile - Fixed Tests"
```

```
PASS src/tests/coach_test/getCoachProfile.test.ts
```

```
getCoachProfile - Fixed Tests
```

- ✓ returns error when coach not found (10 ms)
- ✓ returns error when user is not a coach (1 ms)
- ✓ returns coach profile when details exist (2 ms)
- ✓ returns coach profile with defaults when details are missing (1 ms)
- ✓ throws when a database error occurs (4 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 5 passed, 5 total
```

```
Snapshots: 0 total
```

```
Time: 9.194 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getCoachProfile.test.ts matching "getCoachProfile - Fixed Tests".
```

9. Get Coach Slots

```
s.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getCoachSlots - controller unit tests"
PASS src/tests/coach_test/getCoachSlots.test.ts
  getCoachSlots - controller unit tests
    ✓ returns available, unbooked slots for the given coach ID (8 ms)
    ✓ returns an empty array when the coach has no available slots (2 ms)
    ✓ propagates database errors from CoachSlot lookup (5 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        8.893 s
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getCoachSlots.test.ts/i
ing "getCoachSlots - controller unit tests".
```

10. Get My Coach bookings (Player's Side)

```
okings.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getMyCoachBookings - controller unit tests"
PASS src/tests/coach_test/getMyCoachBookings.test.ts
  getMyCoachBookings - controller unit tests
    ✓ returns failure when the requester is not a player (7 ms)
    ✓ returns populated bookings for the player (2 ms)
    ✓ returns an empty array when no bookings are found (1 ms)
    ✓ propagates errors when CoachBooking lookup throws (4 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        7.827 s
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getMyCoachBookings.test.ts/i with
matching "getMyCoachBookings - controller unit tests".
```

11. Get My coach details

```
tails.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getMyCoachDetails"
PASS src/tests/coach_test/getMyCoachDetails.test.ts
  getMyCoachDetails
    ✓ returns error when called by non-coach (7 ms)
    ✓ returns null and a helpful message when no details exist (2 ms)
    ✓ returns existing coach details (1 ms)
    ✓ propagates database errors (4 ms)
```

```
Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        8.029 s
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getMyCoachDetails.test.ts/i
atching "getMyCoachDetails".
```

12. Get My slots

```
est.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "getMySlots - controller unit tests"
```

```
PASS src/tests/coach_test/getMySlots.test.ts
```

```
getMySlots - controller unit tests
```

- ✓ returns failure when the requester is not a coach (8 ms)
- ✓ returns the coach's unbooked slots ordered by date/time (2 ms)
- ✓ returns an empty array when the coach has no available slots (2 ms)
- ✓ propagates errors when CoachSlot lookup throws (4 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 4 passed, 4 total
```

```
Snapshots: 0 total
```

```
Time: 8.07 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\getMySlots.test.ts/i with  
"getMySlots - controller unit tests".
```

13. Reject Booking request

```
gReques.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "rejectBookingRequest - controller unit tests"
```

```
PASS src/tests/coach_test/rejectBookingReques.test.ts
```

```
rejectBookingRequest - controller unit tests
```

- ✓ returns failure when a non-coach attempts to reject a booking (10 ms)
- ✓ returns failure when the booking cannot be found (2 ms)
- ✓ returns failure when the booking is not pending (2 ms)
- ✓ rejects the booking and saves a provided rejection reason (3 ms)
- ✓ rejects the booking with a default reason when none is provided (2 ms)
- ✓ propagates database errors from CoachBooking lookup (4 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 6 passed, 6 total
```

```
Snapshots: 0 total
```

```
Time: 8.265 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\rejectBookingReques.test.ts/i wi  
matching "rejectBookingRequest - controller unit tests".
```

14. Request Coach Booking

```
Booking.test.ts" -c "d:/SE_my/SE_12/SportSphere/jest.config.js" -t "requestCoachBooking - controller unit tests"
```

```
PASS src/tests/coach_test/requestCoachBooking.test.ts
```

```
requestCoachBooking - controller unit tests
```

- ✓ returns failure when a non-player attempts to request a booking (7 ms)
- ✓ returns failure when slotId is not provided (1 ms)
- ✓ returns failure when the specified slot does not exist (1 ms)
- ✓ returns failure when the slot is already booked (2 ms)
- ✓ returns failure when the player already has a pending or accepted booking for the slot (2 ms)
- ✓ creates and persists a booking request when input is valid (9 ms)
- ✓ propagates database errors from CoachSlot lookup (4 ms)

```
Test Suites: 1 passed, 1 total
```

```
Tests: 7 passed, 7 total
```

```
Snapshots: 0 total
```

```
Time: 8.254 s
```

```
Ran all test suites matching /d:\\SE_my\\SE_12\\SportSphere\\src\\tests\\coach_test\\requestCoachBooking.test.ts/i with tests  
matching "requestCoachBooking - controller unit tests".
```

Final Coach Side Coverage :

```
PS D:\SE_my\SE_12> cmd /c "cd /d d:\SE_my\SE_12\SportSphere && npx jest src/tests/coach_test --coverage --collectCoverageFrom
=src/controllers/coachController.ts --runInBand --detectOpenHandles"
(node:3220) ExperimentalWarning: CommonJS module C:\Program Files\nodejs\node_modules\npm\node_modules\debug\src\node.js is l
oading ES Module C:\Program Files\nodejs\node_modules\npm\node_modules\supports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS src/tests/coach_test/createOrUpdateCoachDetail.test.ts
PASS src/tests/coach_test/getCoachProfile.test.ts
PASS src/tests/coach_test/rejectBookingRequest.test.ts
PASS src/tests/coach_test/requestCoachBooking.test.ts
PASS src/tests/coach_test/createCoachSlots.test.ts
PASS src/tests/coach_test/deleteCoachSlots.test.ts
PASS src/tests/coach_test/getMyCoachDetails.test.ts
PASS src/tests/coach_test/deleteCoachPhoto.test.ts
PASS src/tests/coach_test/getMyCoachBookings.test.ts
PASS src/tests/coach_test/getCoachBookingRequest.test.ts
PASS src/tests/coach_test/getMySlots.test.ts
PASS src/tests/coach_test/getCoachSlots.test.ts
PASS src/tests/coach_test/acceptBookingRequest.test.ts
PASS src/tests/coach_test/getAllCoaches.test.ts

-----|-----|-----|-----|-----|-----
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100 |    92.78 |    100 |    100 |
coachController.ts |    100 |    92.78 |    100 |    100 | 26-29,141,144,171
-----|-----|-----|-----|-----|-----

Test Suites: 14 passed, 14 total
Tests:       70 passed, 70 total
Snapshots:   0 total
Time:        10.464 s, estimated 11 s
Ran all test suites matching /src\\tests\\coach_test/i.
PS D:\SE_my\SE_12>
```