Event Management System - Bug Report

Bug Report & Defect Analysis

Document Information

Project Name	**Event Management System Testing**
Document Type	Bug Report & Defect Analysis
Version	1.0
Author	Bassam Ashraf
Date	October 04, 2025
Review Status	Draft

Executive Summary

Defect Overview

Total Bugs Found	**3**
Critical	0
High	2
Medium	1
Low	0
Resolution Status	All bugs identified with proposed solutions

Impact Assessment

All identified bugs are backend-related affecting API functionality and compliance with technical specifications. No frontend or user interface bugs were discovered during comprehensive testing.

GitHub Repository:

- Link of Repo: https://github.com/Bassam0Ashraf/Event_Management_System_Testing.git

Bug Reports

BUG-001: JWT Token Generation Lacks Uniqueness

Bug ID	**BUG-001**
Title	JWT Token Generation Lacks Uniqueness
Severity	High
Priority	High
Status	Identified
Reporter	Bassam Ashraf
Date Found	October 04, 2025
Component	Backend Authentication
File Location	routes/auth.js

Bug Description:

The JWT token generation process creates static tokens that do not change between login sessions, causing API test failures and potential security issues with session management.

Environment Details:

Server	**Node.js Express Application**
Database	MariaDB
Testing Tool	Postman API Collection
Browser	Not applicable (backend issue)
OS	Windows

Steps to Reproduce:

1. Start the backend server
2. Send POST request to /api/auth/login with valid credentials
3. Note the generated JWT token
4. Logout user
5. Login again with same credentials
6. Compare the new JWT token with previous token
7. Observe that tokens are identical

Expected Behavior:

Each login session should generate a unique JWT token with different timestamps or session identifiers to ensure proper session management and security.

Actual Behavior:

The same JWT token is generated for multiple login sessions, preventing proper session differentiation and causing API test automation failures due to token expiration issues.

Test Evidence (excerpt):

Example of identical tokens across sessions.

```
First Login Token:  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
Second Login Token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." (identical)
```

Root Cause Analysis

The JWT token generation code does not include dynamic elements like login timestamp, causing identical payloads to generate identical tokens.

Current Code (Incorrect):

```
const token = jwt.sign(
{userId: user.id, isAdmin: user.isAdmin},
process.env.JWT_SECRET,
{expiresIn: '1h'}

);
```

Proposed Solution

```
const token = jwt.sign(
  {
    userId: user.id,
    isAdmin: user.isAdmin,
    loginTime: Date.now()
  },
  process.env.JWT_SECRET,
  { expiresIn: '1h' }
);
```

Business Impact

- High: API test automation failures
- Security Risk: Potential session hijacking vulnerability
- User Experience: May affect concurrent user sessions
- Development: Blocks automated testing workflow

Testing Impact

- Postman API test collection fails due to token reuse issues
- Automated regression testing blocked
- Manual testing requires workarounds

BUG-002: User Profile Endpoint Returns 404 Error

Bug ID	**BUG-002**
Title	User Profile Endpoint Returns 404 Error
Severity	High
Priority	High
Status	Identified
Reporter	Bassam Ashraf
Date Found	October 04, 2025
Component	Backend API Routing
File Location	app.js

Bug Description:

The user profile API endpoint is incorrectly configured without a leading slash, causing 404 errors when clients attempt to retrieve user profile information.

Environment Details:

Server	**Node.js Express Application**
Testing Tool	Postman
HTTP Method	GET
Expected Endpoint	/api/users/profile
Actual Endpoint	api/users/profile (missing leading slash)

Steps to Reproduce:

1. Obtain valid JWT authentication token
2. Send GET request to /api/users/profile with Authorization header
3. Observe 404 Not Found error response
4. Check server route configuration
5. Confirm missing slash in endpoint definition

Expected Behavior:

GET request to (**/api/users/profile**) with valid authentication should return user profile data with 200 OK status.

Actual Behavior:

Server returns 404 Not Found error because the endpoint is not properly registered with Express router.

Root Cause Analysis

Express.js route definition missing leading slash character.

Current Code (Incorrect):

```
app.get('api/users/profile', async (req, res) => {
// Profile logic
|});
```

Proposed Solution

```
app.get('/api/users/profile', async (req, res) => {
// Profile logic
});
```

Business Impact

- High: User profile functionality completely broken
- API Compliance: Violates REST API specification
- User Experience: Users cannot access profile information
- Documentation: API documentation becomes inaccurate

Testing Impact

- API test case TC-019: Get User Profile fails
- Profile-related test scenarios blocked
- End-to-end user journey testing affected

BUG-003: Duplicate Email Registration Returns 500 Instead of 400

Bug ID	**BUG-003**
Title	Duplicate Email Registration Returns 500 Instead of 400
Severity	Medium
Priority	Medium
Status	Identified
Reporter	Bassam Ashraf
Date Found	October 04, 2025
Component	Backend Authentication
File Location	routes/auth.js

Bug Description:

When attempting to register a user with an email that already exists in the system, the API returns HTTP status code 500 (Internal Server Error) instead of the specified 400 (Bad Request) as documented in the API specification.

Environment Details:

Server	**Node.js Express Application**
Database	MariaDB with user email uniqueness constraint
Testing Tool	Postman API Collection
HTTP Method	POST
Endpoint	/api/auth/register

Steps to Reproduce:

1. Register a new user with email test@example.com and confirm 201 Created
2. Attempt to register another user with the same email test@example.com
3. Observe response status code and message
4. Compare with API specification requirements

Expected Behavior:

Status Code: 400 Bad Request
Error Message: **{“error": "Email already exists.”}**

Actual Behavior:

Status Code: 500 Internal Server Error
Error Message: **"Something went wrong!"**

Root Cause Analysis

Error handling code returns generic 500 error instead of specific 400 error for duplicate email scenario.

Current Code (Incorrect):

```
// In registration endpoint error handling

catch (error) {
res.status(500).send("Something went wrong!");|
}
```

Proposed Solution

```
catch (error) {
if (error.code === 'ER_DUP_ENTRY' ||  error.message.includes('email'))
{
   res.status(400).json({ "error": "Email already exists." });

} else {
   res.status(500).send("Something went wrong!");
}
}
```

Business Impact

- Medium: API specification non-compliance
- Client Integration: Front-end error handling may not work correctly
- User Experience: Generic error messages provide poor user feedback
- API Documentation: Discrepancy between docs and implementation

Testing Impact

- API validation tests fail schema verification
- Error handling test cases return unexpected results
- API specification compliance testing blocked

Recommendations and Next Steps

Immediate Actions Required

1. Fix BUG-001 and BUG-002 (High Priority)
   - Critical for API functionality
   - Required for test automation success

2. Address BUG-003 (Medium Priority)
   - Improve API specification compliance
   - Enhance error handling consistency

Bug Fix Verification Plan

BUG-001 Verification Steps

1. Apply proposed JWT token fix
2. Run Postman authentication test collection
3. Verify unique tokens generated for each login
4. Confirm API test automation success

BUG-002 Verification Steps

1. Add leading slash to profile endpoint route
2. Test GET /api/users/profile with valid token
3. Verify 200 OK response with user data
4. Run complete API test suite

BUG-003 Verification Steps

1. Implement specific error handling for duplicate emails
2. Test registration with existing email
3. Verify 400 status code and correct error message
4. Update API test validations