|  |
| --- |
| Pfizer Co. |
| E-Medicare |
| Asfand Khanzada 22K-4626  Munnazzar Shahzad 22K-4231 |

|  |
| --- |
| Alishba SHAHZAD  9-5-2025 |

## Authentication Endpoints Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Data | Steps | Expected Result | Actual Result | Status |
| TC\_AUTH\_01 | Return 401 for invalid credentials | { firstName: 'invaliduser', password: 'wrongpassword', role: 'doctor' } | 1. Send POST request to /api/login with payload | 401 Unauthorized; response contains 'msg' | 401 Unauthorized; response contains 'msg' | Pass |
| TC\_AUTH\_02 | Return 401 for missing credentials | {} | 1. Send POST request to /api/login with empty body | 401 Unauthorized; response contains 'msg' | 401 Unauthorized; response contains 'msg' | Pass |
| TC\_AUTH\_03 | Return 401 for missing firstName | { password: 'password123', role: 'doctor' } | 1. Send POST request to /api/login with missing firstName | 401 Unauthorized; response contains 'msg' | 401 Unauthorized; response contains 'msg' | Pass |
| TC\_AUTH\_04 | Return 401 for missing password | { firstName: 'testuser', role: 'doctor' } | 1. Send POST request to /api/login with missing password | 401 Unauthorized; response contains 'msg' | 401 Unauthorized; response contains 'msg' | Pass |
| TC\_AUTH\_05 | Return 401 for missing role | { firstName: 'testuser', password: 'password123' } | 1. Send POST request to /api/login with missing role | 401 Unauthorized; response contains 'msg' | 401 Unauthorized; response contains 'msg' | Pass |

## Doctor Endpoints Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Data | Steps | Expected Result | Actual Result | Status |
| TC\_DOC\_01 | 401 without authentication for getMyPatients | None | 1. Send GET request to /api/doctor/getMyPatients without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_DOC\_02 | Return patients list with valid token | Authorization: Bearer <token> | 1. Send GET request to /api/doctor/getMyPatients with valid token | 200 OK; response has 'count' (number) and 'patients' (array) with required fields | 200 OK; response has 'count' (number) and 'patients' (array) with required fields | Pass |
| TC\_DOC\_03 | 401 without authentication for getPatient | None | 1. Send GET request to /api/doctor/getPatient/1 without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_DOC\_04 | 404 for non-existent patient | Authorization: Bearer <token> | 1. Send GET request to /api/doctor/getPatient/999999 with valid token | 404 Not Found | 404 Not Found | Pass |
| TC\_DOC\_05 | 401 without authentication for addDiagnosis | { diagnosis: 'Test diagnosis' } | 1. Send POST request to /api/doctor/addDiagnosis/1 without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_DOC\_06 | Invalid diagnosis data returns 404 | {} | 1. Send POST request to /api/doctor/addDiagnosis/1 with valid token and empty body | 404 Not Found | 404 Not Found | Pass |

## Nurse Endpoints Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Data | Steps | Expected Result | Actual Result | Status |
| TC\_NUR\_01 | 401 without authentication for getPatients | None | 1. Send GET request to /api/nurse/getPatients without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_NUR\_02 | Return patients list with valid token | Authorization: Bearer <token> | 1. Send GET request to /api/nurse/getPatients with valid token | 200 OK; response is an array | 200 OK; response is an array | Pass |
| TC\_NUR\_03 | 401 without authentication for getPatient | None | 1. Send GET request to /api/nurse/getPatient/1 without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_NUR\_04 | 404 for non-existent nurse patient | Authorization: Bearer <token> | 1. Send GET request to /api/nurse/getPatient/999999 with valid token | 404 Not Found | 404 Not Found | Pass |
| TC\_NUR\_05 | 401 without authentication for getMyDosages | None | 1. Send GET request to /api/nurse/getMyDosages without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_NUR\_06 | Return dosages list with valid token | Authorization: Bearer <token> | 1. Send GET request to /api/nurse/getMyDosages with valid token | 200 OK; response is an array | 200 OK; response is an array | Pass |
| TC\_NUR\_07 | 401 without authentication for administerDosage | { dosageId: 1, notes: 'Test administration' } | 1. Send POST request to /api/nurse/administerDosage without Authorization header | 401 Unauthorized | 401 Unauthorized | Pass |
| TC\_NUR\_08 | Invalid dosage data returns 400 | { notes: 'Test administration' } | 1. Send POST request to /api/nurse/administerDosage with valid token and missing dosageId | 400 Bad Request | 400 Bad Request | Pass |

# Test Coverage

The test coverage includes validation of all key authentication endpoints, doctor and nurse-related functionalities. All CRUD operations related to patient data, diagnosis management, and dosage tracking are covered.  
Gaps in current test cases include missing tests for edge cases such as role-based restrictions, error boundary testing for rate limits, and handling of invalid token formats.

# Test Workflow

1. \*\*Authentication Testing\*\*:  
 - Validate login attempts using both valid and invalid credentials.  
 - Confirm correct system responses for missing fields or unauthorized access.  
  
2. \*\*Doctor Endpoint Testing\*\*:  
 - Ensure patient lists and individual patient data are only accessible with valid tokens.  
 - Validate diagnosis addition functionality.  
  
3. \*\*Nurse Endpoint Testing\*\*:  
 - Check for proper access to patient dosage data.  
 - Confirm that administering dosage requires authentication and valid input.  
  
4. \*\*Negative Testing\*\*:  
 - Verify that endpoints return proper status codes and messages for invalid input and unauthorized access.