# Database Manager Testing Report

## 1. Overview

This report documents the testing process and results for the `db\_manager.py` module of the Hospital Management System. The tests ensure database connectivity, table creation, data integrity, authentication, and security measures.

## 2. Testing Environment

The tests were conducted in a local development environment with the following setup:

• Operating System: Windows 10  
• Database: SQLite 3  
• Python Version: 3.x  
• Testing Framework: unittest  
• Dependencies: sqlite3, hashlib, logging

## 3. Test Cases

### 3.1 Database Connection Test

Objective: Verify that the database connection is established successfully.

Expected Result: Connection should be successful and return a valid connection object.

Actual Result: Connection successful.

### 3.2 Table Creation Test

Objective: Ensure that all required tables are created successfully.

Expected Result: Tables should be present in the database.

Actual Result: All tables were created successfully.

### 3.3 User Authentication Test

Objective: Validate that users can log in using correct credentials and fail with incorrect ones.

Expected Result: Valid users should authenticate successfully; invalid users should be rejected.

Actual Result: Authentication worked correctly.

### 3.4 Data Insertion Test

Objective: Verify that user records and medical records can be inserted correctly.

Expected Result: Data should be inserted without errors and retrievable.

Actual Result: Data insertion and retrieval successful.

### 3.5 Security Test (Password Hashing)

Objective: Ensure that passwords are hashed before storage.

Expected Result: Passwords should be stored in hashed format.

Actual Result: Password hashing verified.

### 3.6 Error Handling Test

Objective: Check how the system handles database connection failures.

Expected Result: Errors should be logged and handled gracefully.

Actual Result: Proper error handling observed.

## 4. Conclusion

The tests confirmed that the `db\_manager.py` module operates as expected, with successful database interactions, authentication mechanisms, and security measures. No critical issues were found. Regular testing is recommended to maintain system integrity.