**TEST PLAN DEVELOPMENT**

**1. TEST PLAN TABLE**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature** | **Description** | **Details/Example** | **Test Owner** | **Test Case Ref** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| Student Management | Complete student lifecycle operation | Add new students, update records, delete students, search functionality | QA Engineer | TC-STU-001 to TC-STU-012 | All student operations execute successfully with data integrity | SQL injection vulnerabilities found in search, basic CRUD operation functional | FAIL | Core business functionality compromised by security issues |
| Course Management | Course creation and maintenance | Create courses, modify details, remove courses, search courses | QA Engineer | TC-CRS001 to TC-CRS-010 | Course data maintained accurately with proper validation | Basic functionality working, search vulnerable to SQL injection | PARTIAL | CRUD operation OK but security vulnerabilities present |
| Database Integrity | Data relationships and constraint | Foreign key enforcement, unique constraints, data consistency | QA Engineer | TC-DB-001 to TC-DB008 | All database constraints enforced, no orphaned records | Foreign keys disabled, constraint not enforced, orphaned records possible | FAIL | Critical data integrity issue confirmed by tests |
| Security | Vulnerability prevention and data protection | SQL injection prevention, input validation, error handling | QA Engineer | TC-SEC-001 to TC-SEC-006 | System rejects malicious inputs, protects sensitive data | SQL injection confirmed, input validation gaps identified | FAIL | High severity security vulnerabilities |
| User Interface | GUI functionality and user experience | Form validation, theme switching, responsive design | QA Engineer | TC-UI-001 to TC-UI-008 | Interface responds correctly, themes apply consistently | Manual testing required, theme system operational | PENDING | Core UI features need security intergration |
| Data Export | Export functionality for reporting | CSV export for students and courses | QA Engineer | TC-EXP-001 to TCexp-004 | Files generate with correct format and compete data | Basic export functional, data integrity concerns | PARTIAL | Export works but relies on vulnerable data |

**2. RISKS AND MITIGATION TABLE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Risk Description** | **Impact** | **Likelihood** | **Mitigation Strategy** | **Owner** | **Test Status** |
| R01 | SQL injection vulnerabilities in search functionality | Critical | High | Implement parameterized queries, input sanitization and security testing | Development Team | Confirmed by test\_sql\_injection\_student search |
| R02 | Data integrity loss from missing foreign key constraints | High | High | Enable foreign key support, implement cascade operations, add validation | Database Architect | CONFIRMED by test\_foreign\_key\_constraints FAILED |
| R03 | Application crashes from unhandled exceptions | High | Medium | Implement comprehensive error handling, input validation and graceful degradation | Development Team | Manual Review – Mixed error handling patterns found |
| R04 | Performance issues with large dataset operations | Medium | Medium | Optimize database queries, implement pagination, add indexing | Performance Engineer | PENDING – Not tested with large datasets |
| R05 | Inconsistent user experience across application tabs | Medium | Low | Implementing centralized state management for themes and data | UI Developer | Manual Review – Theme sync implemented |
| R06 | Data corruption from invalid user inputs | High | Medium | Implement server side validation, data type checking and boundary testing | QA Engineer | CONFIRMES BY test\_input\_validation\_gaps |
| R07 | Test environment configuration issues | Medium | Low | Maintain separate test database, automated environment setup | DevOps Engineer | RESOLVED – Test environment working |

**3. TEST SCENARIOS TABLE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Scenario ID** | **Scenario** | **Description** | **Priority** | **Pre-Conditions** | **Expected Result** | **Actual Result** | **Status** |
| TS01 | Complete student registration | End to end student enrolment process | High | System running, courses available | Student registered with all details saved correctly | Registration functional but data integrity compromised | PARTIAL |
| TS02 | Course creation and management | Full course lifecycle management | High | Valid course data available | Course created, updated and deleted successfully | Basic CRUD operations working, search vulnerable | PASS |
| FAILTS03FAIL | Advanced search operations | Test search with various criteria and edge cases | Medium | Test data populated in system | Accurate and relevant search results returned | SQL injection vulnerabilities found in search | FAIL |
| TS04 | SQL injection attack prevention | Security testing for injection vulnerabilities | Critical | Application n deployed and running | System safely handles malicious inputs with data exposure | SQL injection vulnerabilities confirmed | FAIL |
| TS05 | Data export functionality | CSV export for students and courses | Medium | Data exists in system | Complete and formatted CSV files generated | Basic export functional, relies on vulnerable data | PARTIAL |
| TS06 | User interface theme management | Light/dark mode switching consistency | Low | Application UI loaded | Themes apply correctly across all application sections | Theme system operational and synchronized | PASS |
| TS07 | Error handling and validation | System behaviour with invalid inputs and edge cases | High | Application running | Graceful error messages and system stability maintained | Mixed error handling, validation gaps | PARTIAL |
| TS08 | Database relationship and integrity | Foreign key and data consistency validation | High | Related data exists in tables | Data relationships maintained, constraints enforced | Foreign keys not enforced , integrity compromised | FAIL |

**4. TEST CASES TABLE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario ID** | **Test Case Description** | **Pre Conditions** | **Test Steps** | **Expected Result** | **Actual Result** | **Status** |
| TC-STU-001 | TS01 | Register new student with valid data | Course CS101 exists | 1. Navigate to Students tab  2. Fill all required fields  3. Select course from dropdown  4. Click Add button | Students added successfully, confirmation message shown | Student Added but data integrity not enforce | PARTIAL |
| TC-STU-002 | TS01 | Register student with duplicate student number | Student S1234567 exists | 1. Navigate to Students tab  2. Enter existing student number  3. Fill other fields  4. Click Add button | Error message displayed, student not added | Duplicate prevention working correctly | PASS |
| TC-STU-003 | TS03 | Search students by name | Multiple students exists | 1. Navigate to Students tab  2. Enter student name in search  3. Click Search button | Matching students displayed in results | Search functional but vulnerable to SQL injection | FAIL |
| TC-CRS-001 | TS02 | Add new course with valid data | System running | 1. Navigate to courses tab  2. Enter course details  3. Click Add button | Course added successfully, appears in course list | Course creation working correctly | PASS |
| TC-CRS-002 | TS02 | Add course with duplicate course code | Course TEST101 exists | 1. Navigate to Courses tab  2. Enter duplicate course code  3. Click Add button | Error message shown, course not added | Duplicate prevention working | PASS |
| TC-SEC-001 | TS04 | SQL injection test in student search | System running | 1. Navigate to Students tab  2. Enter SQL injection payload in search  3. Click Search button | System returns empty results, no error or data exposure | Vulnerability confirmed – returns data | FAIL |
| TC-SEC-002 | TS04 | SQL injection test in course search | System running | 1. Navigate to Course tab  2. Enter malicious SQL in search  3. Click Search button | Safe handling, no database errors or crashes | Vulnerability confirmed – returns data | FAIL |
| TC-DB-001 | TS08 | Verify foreign key constraint enforcement | Courses and students exist | 1. Attempt to delete course with enrolled students  2. Check database relationships | Constraints prevents deletion or cascades properly | Foreign keys disabled – deletion allowed | FAIL |
| TC-UI-001 | TS06 | Switch between light and dark themes | Application running | 1. Click theme toggle button  2. Verify all screens  3. Switch back | Theme changes consistently across all UI elements | Theme switching working correctly | PASS |

5. **TRACEABILITY MATRIX TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Linked Test Scenarios** | **Linked Test Cases** | **Linked Bug Reports** | **Status** |
| REQ-001 | System shall allow adding new students with validation | TSO1 | TC-STU-001, TC-STU-002 | BR001 | PARTIAL |
| REQ-002 | System shall allow managing courses with unique codes | TS02 | TC-CRS-001, TC-CRS-002 | BR002 | PASS |
| REQ-003 | System shall provide search functionality for students and courses | TS03 | TC-STU-003 | BR003 | FAIL |
| REQ-004 | System shall prevent SQL injection attacks | TS04 | TC-SEC-001, TC-SEC-002 | BR004 | FAIL |
| REQ-005 | System shall maintain data integrity with foreign keys | TS08 | TC-DB-001 | BR005 | FAIL |
| REQ-006 | System shall support theme switching | TS06 | TC-UI-001 | BR006 | PASS |
| REQ-007 | System shall export data to CSV format | TS05 | TC-EXP-001 | BR007 | PARTIAL |
| REQ-008 | System shall handle errors gracefully | TS07 | TC-ERR-001 | BR008 | PARTIAL |

**6. TRACEABILITY MATRIX TABLE EXAMPLE 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Test Case ID | Test Case Description | Bug ID | Bug Description | Status |
| REQ-001 | Users can add new students with validation | TC-STU-001 | Register student with valid data | BR001 | Student email validation not working | Open |
| REQ-002 | Users can manage courses with unique codes | TC-CRS-001 | Add new course with valid data | BR002 | Course code case sensitivity issue | Open |
| REQ-003 | Users can search for students and courses | TC-STU-003 | Search students by name | BR003 | Search returns partial matches only | Open |
| REQ-004 | System prevents SQL injection attacks | TC-SEC-001 | SQL injection test in student search | BR004 | Search vulnerable to SQL injection | Open |
| REQ-005 | System maintains data integrity | TC-DB-001 | Verify foreign key constraints | BR005 | Foreign keys not enforced | Open |
| REQ-006 | System supports theme switching | TC-UI-001 | Switch between themes | BR006 | Theme not syncing between tabs | Open |

**7. BUGS REPORTS TABLE**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Title** | **Description** | **Severity** | **Priority** | **Steps to Reproduce** | **Expected Result** | **Actual result** | **Status** | **Assigned To** |
| BR001 | Student email validation not working | System accepts invalid emails format during student registration | Medium | P2 | 1. Navigate to Students tab  2. Enter invalid  3. Click Add button | System should reject invalid email format | System accepts invalid email | Open | Developer A |
| BR002 | Course code case sensitivity issue | Duplicate course codes allowed with different cases | Medium | P2 | 1. Add course “CS101”  2. Add course “cs101”  3. Check database | System should treat course codes as insensitive | Both courses added as separate records | Open | Developer B |
| BR003 | Search returns incomplete results | Student search only matches exact names | Low | P3 | 1. Add student “Jonathan Smith”  2. Search for “John”  3. Check results | Should return partial matches | No result returned for partial match | Open | Developer C |
| BR004 | SQL injection incomplete results | Search fields vulnerable to SQL injection attacks | Critical | P1 | 1. Navigate to any search field  2. Enter: ‘;DROP TABLE students; --  3. Execute search | System should safely handle input | Potential data loss risk | Open | Security Team |
| BR005 | Foreign key constraints not enforced | Can delete courses with enrolled students | High | P1 | 1. Create course with enrolled students  2. Attempt to delete course  3. Check database | Should prevent deletion or cascade delete students | Course deleted, students orphaned | Open | Database Team |
| BR006 | Theme not synchronized between tabs | Theme change in one tab does not affect other tabs | Low | P3 | 1. Open both Students and Courses tabs  2. Change theme in Students tab  3. Check Courses tab | Both tabs should update theme simultaneously | Only current tab updates theme | Open | UI Developer |

**TESTING SCHEDULE**

**Week 1: Test Preparation (COMPLETED)**

* Test environment setup
* Test data preparation
* Test case review and approval

**Week 2: Functionality Testing (COMPLETED)**

* Student management testing
* Course management testing
* Database operations testing

**Week 3: Security and Integration Testing (COMPLETED)**

* Security vulnerability testing
* Integration workflow testing
* User interface testing

**Week 4: Regression and Final Validation (IN PROGRESS)**

* Bug fix verification (PEDNING)
* Registration testing (COMPLETED)
* Final test report preparation (COMLETED)

**TEST SUMMARY METRICS**

|  |  |  |
| --- | --- | --- |
| Metric | Value | Assessment |
| Total Test Cases | 26 | Comprehensive Coverage |
| Test Cases Executed | 14 | 54% Coverage |
| Test Cases Passed | 13 | 93% success rate of executed |
| Critical Issue Found | 3 | High risk to production |
| Security Vulnerabilities | 2 | Immediate remediation required |
| Date Integrity Issues | 1 | Critical database concern |
| Overall Test Status | FAIL | Not production ready |