**Project Test Plan for Course Management System**

**1. Introduction**

**Project Overview:** The Course Management System is a web-based application designed to help universities manage courses, student registrations, student schedules, and student information. The system includes functionalities for registration, enrollment/withdraw, grade tracking, and scheduling.

**Objective:** The purpose of this test plan is to outline the testing strategy, resources, schedule, and scope for the Course Management System to ensure a high-quality release.

**2. Testing Teams**

**Test Manager:** Responsible for overall test strategy, planning, and communication with stakeholders.

**Test Engineers:** Perform test case design, execution, and report defects. Specialize in different testing types (e.g., functional, integration, system testing).

**Automation Engineers:** Develop and mantain automated test scripts for regression testing.

**QA Analysts:** Analyze all test results, and ensure coverage, and perform risk assessment.

**3. Test Scope**

**In-Scope:** Testing will cover course creation, enrollment process, schedule management, user authentication, and data security features (password hashing).

**Out-of-Scope:** Third-party integrations not related to the core functionality of the CMS. (Not applicable for this assignment??)

**4. Exit Criteria**

**Quality Metrics:**

* 95% of all test cases must pass.
* All critical and major defects must be addressed.
* Code coverage of at least 89%.
* Sign-off: The QA team, Test Manager, and Project Manager must sign off on testing completion based on the achievement of quality metrics.

**5. Test Estimated Effort**

**Effort Estimation:**

* Test Planning: 1 week
* Test Design: 1 weeks
* Test Execution: 3 weeks (parallel)
* Test Automation: 2 weeks (parallel)
* Test Closure: 1 week
* Resource Allocation: 2 Test Engineers, 1 Automation Engineer, 1 QA Analyst, Test Manager.

**6. Incident Reporting Strategy**

**Incident Reporting Process:**

* Defects will be reported using JIRA.
* Each defect will be categorized by severity such as Critical, Major, Minor, and the priorities High, Medium, Low.
* The workflow will involve logging, fixing, retesting, and closing alldefects.

**Severity and Priority Levels:**

* **Critical:** System crashes, data loss, or security breaches.
* **Major:** Functional errors that affect major features without workarounds.
* **Minor:** Minor UI issues or functional bugs with workarounds.

**7. Defect Classification**

* **Functional Defects:** Issues related to the core functionalities like course creation, enrollment, and schedule management.
* **UI/UX Issues:** Problems related to user interface inconsistencies or usability issues.
* **Performance Issues:** Defects related to slow system response times or load issues.
* **Security Vulnerabilities:** Any issues related to data security, authentication, or unauthorized access.

**8. Management Strategies for Configuration**

**Version Control:** All test scripts, data, and documentation will be stored in GitHub for version control.

**Environment Management:** The test environment will mirror the production environment as closely as possible, including databases and configurations.

**Tool Integration:** Configuration management tools will be integrated with Jenkins for continuous integration and deployment.

**9. Tools Used Throughout the Testing Lifecycle**

**Test Management Tool:** TestRail for managing test cases, tracking progress, and reporting.

**Automation Tool:** Selenium WebDriver for automating regression test cases.

**Defect Tracking Tool:** JIRA for logging, tracking, and managing defects.

**Performance Testing Tool:** JMeter for testing system performance under load.

**Version Control Tool:** GitHub for managing test scripts and version control.

**10. References**

Spillner, A., Linz, T., & Schaefer, H. (2014). Software testing foundations: A study guide  
for the certified tester exam (4th ed.). <https://platform.virdocs.com/read/567697/95/#/4/4>

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning. [Course Text](https://platform.virdocs.com/read/2348054/11/#/4/2[ch06]/6/4[ch06-sect1-002]/10/4/2)