Software Test Plan (STP) - Academic Performance Analytics System

Project: Academic Performance Analytics System (APAS)
Version: 1.0
Authors: QA Team - Diya D Bhat, Dhanya Prabhu, Delisha Riyona Dsouza, Eshwar R A.
Team Lead - Diya D Bhat
Date: 2-11-2025
Status: Sample / Draft

1. Introduction

Purpose: This test plan defines the approach, scope, objectives, resources, and responsibilities for testing the Academic Performance Analytics System (APAS). The purpose of this document is to ensure that all functional, non-functional, and security requirements specified in the SRS are validated through systematic testing. The QA team will use this plan as a roadmap to prepare, execute, and report on all test activities, ensuring that APAS meets its quality standards before release.

Scope: Testing will cover the following APAS features:

- **Authentication & Access Control**: Institutional SSO, role-based dashboards (student, instructor, admin).
- **Data Ingestion**: Import grades, attendance, and assignments through API and CSV.
- **Dashboards**: Visualization of academic data for students, instructors, and administrators.
- **Predictive Analytics**: Risk scoring of students based on academic performance.
- **Reporting & Exports**: PDF/CSV downloads, anonymized data exports.
- **Notifications**: Email and in-app alerts for at-risk students.

Exclusions (out of scope):

- Advanced ML model development beyond baseline predictive analysis.
- Infrastructure management (server hosting, deployment automation).

References: APAS SRS v1.0, UML Design Diagrams v1.0, Institution QA Guidelines, WCAG 2.1 AA (Web Content Accessibility Guidelines, for accessibility compliance), Institutional Data Privacy Policy (covers student data & PII protection).

Definitions: APAS (Academic Performance Analytics System), SRS (Software Requirements Specification), RTM (Requirements Traceability Matrix), RBAC (Role-Based Access Control), SSO (Single Sign-On), PII (Personally Identifiable Information)

2. Test Items

- Authentication & Access Control
- Data Ingestion (API + CSV upload)
- Student Dashboard
- Instructor Dashboard
- Administrator Dashboard
- Predictive Analytics Module
- Reporting & Export Module
- Notification System
- Audit & Logging

3. Features to be Tested

Features mapped to SRS requirement IDs:

Functional requirements to be tested:

- APAS-F-001: User authentication via SSO/local login
- APAS-F-002: Role-based access control (student/instructor/admin)
- APAS-F-010: Data ingestion via LMS API
- APAS-F-011: Data ingestion via CSV upload
- APAS-F-020: Instructor dashboard – course KPIs and analytics
- APAS-F-021: Student dashboard – personal progress and risk score
- APAS-F-022: Admin dashboard – department/institution reports
- APAS-F-030: Predictive risk score calculation per student
- APAS-F-040: Export dashboards to PDF/CSV
- APAS-F-050: Email alerts for at-risk students
- APAS-F-051: In-app notifications for students/instructors

Non-functional requirements to be tested:

- APAS-NF-001: UI response time ≤ 2s
- APAS-NF-003: System availability ≥ 99.5%
- APAS-NF-005: Data encryption (TLS + AES-256)

4. Features Not to be Tested

- **Advanced ML Models** – Testing of advanced predictive analytics beyond the baseline risk score is excluded (research scope, future phase).
- **Infrastructure & Deployment** – Server hosting, cloud provisioning, and CI/CD automation are not part of QA testing (handled by DevOps).
- **Third-Party LMS Production Integrations** – Only staging/test APIs will be validated; live production LMS integrations are excluded.
- **UI Customization & Legacy Browsers** – Cosmetic branding changes and unsupported browsers (e.g., IE, outdated OS) are out of scope.
- **External Notification Delivery** – Email/SMS delivery reliability outside APAS (SMTP/institution servers) will not be tested.

5. Test Approach / Strategy

Levels:

- **Unit Testing :** Each module such as authentication, data ingestion, dashboards, reporting, predictive analytics will be tested independently by developers using dummy and edge-case datasets.
- **Integration Testing :** Interfaces between modules like, ingestion, database, dashboards, ML risk scoring etc. will be validated. API and CSV upload flows will also be tested.
- **System Testing :** End-to-end workflows will be tested to ensure that APAS functions correctly as a whole.
- **Acceptance Testing (UAT) :** Instructors and administrators will validate system usability, accuracy of predictions, and reporting against acceptance criteria defined in the SRS.

Types:

- **Functional Testing :** Validate all functional requirements from SRS v1.0, including authentication, dashboards, reporting, notifications, and predictive analytics.
- **Regression Testing :** Re-run critical test cases after bug fixes or new feature additions to ensure existing functionality is not broken.
- **Performance Testing :** Evaluate response time (<3 seconds for dashboard queries), scalability (up to 50,000 student records), and concurrent usage (≥2000 users).
- **Usability & Accessibility Testing :** Verify that dashboards are intuitive, role-based, and compliant with WCAG 2.1 AA (screen-reader support, keyboard navigation, high-contrast mode, scalable text).
- **Security Testing :** Ensure secure login (SSO), role-based access control (RBAC), TLS 1.2+ for all communications, encrypted student data at rest, and anonymization in reports.
- **Data Quality Testing :** Verify correctness and integrity of ingested data (e.g., grades not corrupted, attendance records match source).
- **Predictive Model Validation :** Evaluate accuracy of the risk prediction model (minimum acceptable AUC/accuracy threshold defined in SRS).

Entry Criteria: Stable build delivered by development team, test environment configured

with database and APIs, anonymized test data sets available, test plan and test cases reviewed and approved.
Exit Criteria: 100% of planned test cases executed, no critical or high severity defects open, all acceptance criteria satisfied.

5.1 Security Validation

- **Authentication & Access Control**: Verify Single Sign-On (SSO), Role-Based Access Control (RBAC), and session timeouts. Unauthorized users must not access restricted dashboards.
- **Data Protection & Privacy**: Confirm TLS 1.2+ for all communications, encryption of sensitive data at rest, and anonymization of report exports to prevent exposure of PII.
- **Input & Data Validation**: Test CSV/API ingestion with negative cases (fuzzing, malformed inputs) to prevent SQL injection, XSS, and corruption.
- **Audit & Logging**: Ensure all activities are logged (user, action, timestamp) without exposing credentials or PII. Logs must be append-only and admin-restricted.
- **Vulnerability Testing**: Perform penetration testing on authentication flows and dashboards, covering OWASP Top 10 risks (SQL Injection, XSS, CSRF, etc.).

6. Test Environment

Hardware:

- Server hosting APAS web app and database (Linux/Windows server)
- Client devices: PC, laptop, tablet, smartphone with modern browsers
- Optional peripherals: Printers for PDF report generation

Software:

- APAS web application v1.0
- Relational database (PostgreSQL/MySQL)
- LMS API sandbox (e.g., Moodle/Canvas test environment)
- Authentication service (SSO/OAuth2 provider)

Tools:

- Selenium (UI automation testing for dashboards and login)
- Postman (API testing for LMS integration & data ingestion)
- JMeter (performance/load testing for concurrency and response time)
- Jira (defect tracking & test case management)
- pytest/unittest (unit and integration test automation)
- SonarQube (code quality/security checks)

Test Data:

- Dummy student accounts (with varying grades, attendance, and assignments)
- Instructor/admin accounts with different role privileges
- Sample CSV files (valid and invalid data formats)
- Synthetic datasets for predictive analytics testing (risk score validation)
- Large dataset (50k+ students) for batch job performance tests

7. Test Schedule

Test Case Design (14-Sep-2025 to 20-Sep-2025)

- Document different scenarios (valid, invalid, edge cases).
- By 20-Sep, have the test cases reviewed and finalized for use.

Environment Setup (21-Sep-2025 to 23-Sep-2025)

- Set up the required tools (IDE, databases, blockchain/ML frameworks if applicable).
- Configure test data and project environment on local/system setup.
- Ensure the system is ready to execute test cases.

Test Execution Phase 1 (24-Sep-2025 to 10-Oct-2025)

- Begin executing test cases.
- Note down results, identify bugs or issues, and re-run after small fixes.
- Maintain a defect log for documentation.

Test Execution Phase 2 (11-Oct-2025 to 20-Oct-2025)

- Continue with remaining and advanced test cases.
- Cover integration testing, boundary cases, and project-specific validations.
- Aim to reach at least 90–95% test case completion.

User Acceptance Testing (UAT) 21-Oct-2025 to 31-Oct-2025)

- Share the project with peers/mentor for review.
- Collect feedback and fix any identified issues.
- Prepare for demonstration based on the feedback loop.

Final Stabilization & Report Preparation (01-Nov-2025 to 10-Nov-2025)

- Refine the code, test results, and fix last-minute issues.
- Prepare project report, documentation, and screenshots of test cases.
- Practice for viva/presentation.

Project Submission & Closure (11-Nov-2025 to 15-Nov-2025)

- Submit the final project report and source code

8. Test Deliverables

- **Test Plan Document** – Defines the scope, objectives, features to be tested/not tested, schedule, resources, and responsibilities.
- **Test Case Document** – Detailed test cases with steps, input data, and expected vs. actual results.
- **Test Data Sets** – Sample student data, commit logs, and input datasets used for functional and ML-related testing.
- **Defect/Issue Log** – A record of all identified bugs, their severity, resolution status, and retest outcomes.
- **Test Execution Report** – Documentation of executed test cases, pass/fail status, and coverage statistics.
- **User Acceptance Testing (UAT) Feedback** – Peer/mentor review notes and acceptance sign-off.
- **Final Test Summary Report** – Consolidated report summarizing test activities, results, issues resolved, and project quality assessment.
- **Supporting Evidence** – Screenshots, logs, and outputs collected during test runs for documentation and viva presentation.

9. Roles and Responsibilities

Role	***Name***	***Responsibility***
QA Lead	Diya D Bhat	Prepare plan, coordinate execution
Test Engineer	Dhanya Prabhu	Design & execute test cases, log defects
Developer	Eshwar R A	Support defect fixes and triage
Product Owner	Delisha Riyona Dsouza	Approve test results, sign-off readiness

10. Risks and Mitigation

Risk	***Mitigation***
LMS API integration delays or instability	Develop mock API stubs for testing; engage LMS vendors early for sandbox access; maintain fallback CSV import functionality

Predictive ML model accuracy below acceptable threshold	Establish baseline accuracy metrics early; prepare alternative algorithms (logistic regression, decision trees); validate with historical data
Large dataset performance degradation (50k+ students)	Implement incremental testing with smaller datasets; optimize database queries and indexing; conduct load testing in phases
SSO/OAuth2 integration complexity and delays	Implement local authentication fallback; coordinate with institutional IT early; prepare test accounts and certificates
Data privacy compliance violations (PII exposure)	Implement data anonymization early in development; conduct regular security audits; establish clear data handling procedures
Test environment downtime affecting schedule	Maintain backup cloud-based test environment; implement automated environment setup scripts; establish SLA with infrastructure team
Insufficient or poor quality test data	Generate synthetic datasets early; coordinate with registrar for anonymized sample data; create data generation scripts

11. Assumptions & Dependencies

Dependencies:

- LMS API sandbox environments (Moodle/Canvas/Blackboard) will be available and stable for integration testing
- Institutional SSO/OAuth2 provider will be configured and accessible for authentication testing
- PostgreSQL/MySQL database server will be provisioned and configured for test environment
- Test datasets containing anonymized student records (grades, attendance, assignments) will be provided by registrar office
- SMTP server configuration will be available for email notification testing
  Python ML libraries (pandas, scikit-learn, numpy) and web frameworks will be installed and configured
- Network connectivity and firewall rules will allow API communication between APAS and external systems

Assumptions:

- Institutional data privacy policies and procedures are already established and documented

- Baseline accuracy requirements for predictive analytics model are defined (minimum 70% AUC)
- Browser compatibility testing will focus on modern browsers (Chrome 90+, Firefox 88+, Edge 90+)
- Test execution will occur during business hours when technical support is available
- The development team will provide stable builds with adequate lead time before test execution phases
- Test data will remain static during active test execution to ensure consistent results

12. Suspension & Resumption Criteria

Suspend testing if:

- A critical security vulnerability was discovered that exposes Personally Identifiable Information or allows unauthorised access
- Test environment unavailable for >8 hours during scheduled test execution periods
- Build stability issues blocking >40% of planned test cases due to application crashes or data corruption
- LMS API integration completely non-functional preventing data ingestion testing for >24 hours
- Database corruption or data integrity issues affecting baseline test datasets
- Predictive model producing completely invalid results (accuracy <50% or runtime errors)

Resume testing if:

- Security patches applied and verified through penetration testing or security audit
- Test environment restored and validated with smoke tests passing
- Build stability confirmed with <10% test case failure rate due to environmental issues
- API connectivity restored and validated through basic integration tests
- Database restored from backup with data integrity verification completed
- ML model issues resolved and baseline accuracy requirements met (>70% AUC on validation dataset)
- Formal approval received from QA Lead and Product Owner to resume test activities

13. Test Case Management & Traceability

The Requirements Traceability Matrix (RTM) ensures mapping of SRS requirements to test cases, design components, and testing activities. It establishes bidirectional traceability to verify that every requirement has corresponding test cases and every test case traces back to requirements.

Traceability Coverage

Metric	Value	Status

Total Requirements	15 + 3 Security	✓ Complete
Test Cases Defined	15	✓ Linked
Forward Traceability (Req → Test)	100%	✓ Verified
Backward Traceability (Test → Req)	100%	✓ Verified

Requirements Traceability Matrix (RTM)

Requirement ID	Test Case ID	Description
APAS-F-001	TC-Auth-01	Valid credentials login
APAS-F-001	TC-Auth-02	Invalid credentials handling
APAS-F-002	TC-RBAC-01	Role-based access control
APAS-F-010	TC-Ingest-01	API data ingestion
APAS-F-011	TC-Ingest-02	CSV data ingestion
APAS-F-020	TC-Dash-01	Instructor dashboard KPIs
APAS-F-021	TC-Dash-02	Student dashboard view
APAS-F-022	TC-Dash-03	Admin dashboard aggregation
APAS-F-030	TC-ML-01	Risk score calculation
APAS-F-040	TC-Report-01	Export to PDF/CSV
APAS-F-050	TC-Alert-01	Email alerts for at-risk students
APAS-F-051	TC-Alert-02	In-app notifications
APAS-NF-001	TC-Perf-01	Dashboard response time ≤ 2s
APAS-NF-003	TC-Reliab-01	System availability ≥ 99.5%
APAS-NF-005	TC-Sec-01	Data encryption & TLS

14. Test Metrics & Reporting

Metric	Definition / Formula	Current Status (After Epics 1–5)	Remarks
% Test Cases Executed	(Number of test cases executed ÷ test cases planned till now) × 100	14 / 16 = 87.5 %	All planned tests executed for Epics 1–6
% Passed / Failed	(Number of passed or failed test cases ÷ Executed cases) × 100	Passed = 12 (87.5 %) Failed = 2 (12.5 %)	Minor UI & data-sync issues logged

Defect Density	Number of defects ÷ Total test cases	4 / 16 = 0.25 defects per test case	Within acceptable range
Defect Aging	Average time open before resolution	Avg = 1.5 days	All critical defects resolved within sprint
Requirement Coverage	(Number of requirements tested ÷ Total requirements) × 100	45 / 45 = 100 %	Functional coverage achieved for Epics 1–5

Reports:

- Daily execution status

Daily progress was tracked across all 8 epics. Functional testing for Epics 1–5 is nearly complete, with 16 test cases executed — 15 passed and 1 showed a minor backend–frontend sync issue. No major blockers were found. CI/CD pipeline tests were monitored, though some build failures occurred due to missing dependencies and unmerged frontend branches.

- Final Test Summary Report

Testing progress is around 70% complete, with unit testing for Epics 1–5 successfully validated. Integration, system, and acceptance testing for all 8 epics are yet to be done. CI/CD pipeline integration is under verification, with build failures due to dependency and configuration issues. Once resolved and merged, final regression and deployment validation will follow.

15. Approvals

Role	Name	Signature / Date
QA Lead	Diya D Bhat	Diya D Bhat / 2-11-2025
Dev Lead	Eshwar R A	Eshwar R A / 2-11-2025
Product Owner	Delisha Riyona Dsouza	Delisha Riyona Dsouza / 2-11-2025
Test Developer	Dhanya Prabhu	Dhanya Prabhu / 2-11-2025

16. Test Cases

This section provides sample test cases for the APAS System. Each test case includes a unique identifier, description, preconditions, input data, expected results, and postconditions.

Test Case ID	Test Scenario / Description	Preconditions	Test Steps / Input Data	Expected Result	Postconditions / Remarks
TC-Auth-01	Validate successful login via SSO/local credentials (APAS-F-001)	User registered with valid institutional credentials	1. Open APAS login page 2. Enter valid username/ password 3. Click Login	User authenticated and redirected to dashboard based on role	Session token generated; login recorded in Audit Log
TC-Auth-02	Validate invalid login attempt handling (APAS-F-001)	User registered with invalid credentials	1. Enter incorrect username/password 2. Click Login each time	Error message “Invalid credentials” displayed; user remains on login page	No session created. Attempt recorded in log
TC-RBAC-01	Verify role-based access restriction (APAS-F-002)	Student, Instructor, Admin accounts available	1. Login as each role 2. Try accessing other roles’ pages	Access allowed only to authorized dashboards	RBAC verified; unauthorized access blocked
TC-Ingest-01	Ingest data via LMS API (APAS-F-010)	API token configured; valid JSON payload	1. POST request /ingest/api 2. Provide valid student records	HTTP 200 OK; data stored in database	Records visible in dashboards

TC-Ingest-02	Ingest data via CSV upload (APAS-F-011)	Instructor logged in; valid CSV file	1. Upload CSV file 2. Submit upload	“Upload successful” message; rows saved to DB	Audit log updated
TC-Dash-01	Verify Instructor dashboard KPIs (APAS-F-020)	Instructor account with course data	1. Open Instructor Dashboard	KPIs (avg grade, attendance %) display correctly	Matches DB values
TC-Dash-02	Verify Student dashboard performance view (APAS-F-021)	Student account with academic records	1. Login as Student 2. View Dashboard	Grades, attendance trend, risk score displayed correctly	Data consistent with backend
TC-Dash-03	Verify Admin dashboard aggregation (APAS-F-022)	Admin account active; department data present	1. Open Admin Dashboard	Department summary and analytics displayed	Aggregated data matches DB reports
TC-ML-01	Verify risk score calculation (APAS-F-030)	Dataset available for model run	1. Trigger risk analysis 2. Check scores	Risk scores (0–1) generated successfully	Results stored for dashboards
TC-Report-01	Export dashboards to PDF/CSV (APAS-F-040)	Dashboard loaded for user	1. Click Export 2. Choose PDF or CSV	File downloaded; content matches dashboard	Export record logged
TC-Alert-01	Verify email alert for at-risk students	SMTP configured; students with risk > threshold	1. Run risk analysis 2. Check email notifications	Email alerts sent to faculty / students	Alert status logged

	(APAS-F-050)				
TC-Alert-02	Verify in-app notifications (APAS-F-051)	User dashboard loaded with alerts	1. Login 2. Open Notifications panel	Alert message displayed in dashboard	Marked as read after view
TC-Perf-01	Verify dashboard response time ≤ 2 s (APAS-NF-001)	Load-test setup with 100 users	1. Run JMeter test 2. Measure response times	90th percentile ≤ 2 seconds	Meets NFR target
TC-Reliab-01	Verify system availability (APAS-NF-003)	Monitoring tool active	1. Observe uptime for 30 days	Availability ≥ 99.5% (excl. maintenance)	SLA compliance verified
TC-Sec-01	Verify data encryption and TLS (APAS-NF-005)	HTTPS enabled; DB encryption on	1. Access system via HTTPS 2. Check DB storage	Only TLS connections; encrypted records	Security criteria met