AI Detection Tool

DevOps Sprint Plan & Implementation Guide UK Higher Education AI Content Detection System

---

Executive Summary

This document outlines a comprehensive 12-week DevOps sprint plan for developing, testing, and deploying an AI Detection Tool designed specifically for UK Higher Education institutions. The tool analyzes written work to identify AI-generated content and assesses critical analysis quality against UK HE levels 4-8.

Project Overview

Objectives

1. Develop a Python-based AI detection engine with 85%+ accuracy
2. Create an intuitive web interface for educators and administrators
3. Implement UK HE level-specific analysis (Levels 4-8)
4. Establish comprehensive testing and quality assurance processes
5. Deploy production-ready system with monitoring and support

Technology Stack

Component	Technology
Backend	Python 3.11+, Flask 3.0
Frontend	HTML5, CSS3, JavaScript (ES6+)
Database	PostgreSQL 15
Caching	Redis 7.0
Web Server	Nginx 1.24, Gunicorn
Container	Docker, Kubernetes

Component	Technology
CI/CD	GitHub Actions, Jenkins
Monitoring	Prometheus, Grafana, ELK Stack
Testing	pytest, Selenium, JMeter

Sprint Plan (12 Weeks)

Sprint 0: Pre-Sprint Risk Mitigation & Prerequisites

Duration: 1-2 Weeks (before Sprint 1 begins)

All identified risks must be addressed with concrete actions **before** development starts. This sprint ensures the team enters Week 1 with a secure foundation, clear boundaries, and reduced exposure.

1. Scope Creep Prevention (Risk: High)

Action	Owner	Completion Criteria
Define and sign off a formal Product Requirements Document (PRD) with all stakeholders	Product Owner	PRD approved and baselined
Establish a Change Control Board (CCB) with defined escalation process	Project Manager	CCB charter signed
Create a prioritized and frozen Sprint 1-2 backlog — no items added without CCB approval	Scrum Master	Backlog locked in project management tool
Agree Definition of Done (DoD) for each sprint	Full team	DoD documented and visible
Schedule fortnightly stakeholder alignment sessions for the full 12 weeks	Project Manager	Calendar invites sent

2. Data Privacy Breach Prevention (Risk: Critical)

Action	Owner	Completion Criteria
Complete a Data Protection Impact Assessment (DPIA) per GDPR Article 35	Data Protection Officer	DPIA submitted and reviewed
Register the project with the institution's Data Protection Officer	Project Manager	Registration confirmed
Define data classification scheme (what is PII, what is not)	DPO / Lead Dev	Classification document approved

Action	Owner	Completion Criteria
Draft and approve a data retention and deletion policy	DPO	Policy signed off
Create an Incident Response Plan with roles, escalation paths, and communication templates	Security Lead	IRP documented and tabletop-tested
Establish encryption standards: TLS 1.3 in transit, AES-256 at rest — documented in architecture decision record	Lead Dev / Security	ADR logged
Conduct a Threat Modelling session (STRIDE or equivalent) on the proposed architecture	Security Lead	Threat model documented, mitigations logged

3. False Positive Mitigation (Risk: High)

Action	Owner	Completion Criteria
Source and curate a validation dataset: minimum 500 samples across HE levels 4-8, balanced AI-generated vs human-written	ML Lead	Dataset versioned and stored
Define accuracy, precision, recall, and F1 targets per HE level	ML Lead / Product Owner	Metrics documented in PRD
Establish a confidence interval display strategy (e.g., "likely AI-generated" vs "uncertain") to set educator expectations	UX Lead / Product Owner	UI wireframes approved
Plan a feedback loop mechanism so educators can flag incorrect results for model retraining	ML Lead	Feedback workflow designed
Identify and document known limitations (e.g., short texts, multilingual content, paraphrasing tools)	ML Lead	Limitations page drafted

4. Performance Risk Mitigation (Risk: Medium)

Action	Owner	Completion Criteria
Define performance budgets: page load <2s, analysis <5s for 5000 words, 500+ concurrent users	Lead Dev	Budgets in non-functional requirements
Select and provision a scalable architecture from day one (Docker + K8s with auto-scaling)	DevOps Lead	Architecture Decision Record logged
Establish a performance testing cadence — load tests run at the end of every sprint, not just Sprint 10	QA Lead	Testing schedule in sprint ceremonies
Set up basic monitoring (Prometheus + Grafana) in the dev environment from Sprint 1	DevOps Lead	Dashboards live in dev

5. Integration Delay Mitigation (Risk: Medium)

Action	Owner	Completion Criteria
Define all API contracts (OpenAPI/Swagger specs) before any code is written	Lead Dev / Frontend Lead	API specs committed to repo
Agree on database schema v1 and document migration strategy	Lead Dev / DBA	Schema ERD reviewed and approved
Set up contract testing framework (e.g., Pact) in CI pipeline	QA Lead	Contract tests running in CI
Identify all third-party integrations (LMS, institutional SSO) and confirm access/ credentials	Project Manager	Integration inventory complete

6. Resource Constraint Mitigation (Risk: Medium)

Action	Owner	Completion Criteria
Confirm team allocation and availability for the full 12 weeks	Project Manager	Resource plan signed off
Identify bus-factor risks and establish cross-training pairs	Tech Lead	Pairing schedule created
Secure budget approval for external consulting (security audit, accessibility audit)	Project Manager	Budget confirmed
Set up development environments, accounts, and access for all team members	DevOps Lead	All devs can build and run locally

Sprint 0 Exit Criteria

Before proceeding to Sprint 1, **all** of the following must be confirmed:

Risk Mitigation - [ ] PRD signed off by all stakeholders - [ ] DPIA completed and submitted - [ ] Incident Response Plan documented and tested - [ ] API contracts defined and committed - [ ] Performance budgets documented - [ ] All team members have working dev environments - [ ] Change Control Board established - [ ] Monitoring infrastructure provisioned in dev - [ ] Threat model completed

Testing Readiness (see Testing Strategy — Prerequisites) - [ ] All test tooling installed and configured in CI pipeline (pytest, Selenium, JMeter, OWASP ZAP, Pact, axe-core, Bandit) - [ ] Test environments provisioned (local, CI, staging) - [ ] Validation dataset curated and versioned (650+ documents across all categories) - [ ] Quality gates defined, committed to repo, and enforced as blocking CI checks - [ ] Test data factories created for core models - [ ] Defect management process agreed and tooling configured - [ ] Test reporting dashboards live - [ ] Sprint testing cadence communicated to full team

UX/UI Readiness (see UX/UI — Prerequisites) - [ ] User research completed (5+ educator interviews) - [ ] Personas validated and signed off by stakeholders - [ ] Competitor analysis documented - [ ] Design system defined (colours, typography,

spacing, grid, icons) - [ ] Low-fidelity wireframes approved for all pages - [ ] Interactive prototype usability-tested with 3+ educators - [ ] High-fidelity mockups approved for Sprint 3-4 pages - [ ] Accessibility standards documented and communicated to all developers

Production Readiness (see Production — Prerequisites) - [ ] Cloud provider selected and accounts provisioned (dev, staging, production) - [ ] IaC repository created with base modules committed - [ ] Network architecture documented and approved - [ ] DR plan documented with RPO/RTO targets, test-restore verified - [ ] CI/CD pipeline skeleton running (build + lint + test) - [ ] TLS certificates and domain DNS configured

Monitoring & Maintenance Readiness (see Monitoring & Maintenance — Prerequisites) - [ ] Prometheus + Grafana provisioned in dev, scraping app metrics - [ ] ELK stack provisioned in dev, indexing logs - [ ] PagerDuty configured with on-call schedules and test page verified - [ ] SLIs, SLOs, and error budget policy approved - [ ] Logging and tracing standards documented - [ ] All 12 must-have runbooks written and committed - [ ] Cloud budget alerts configured - [ ] `#incidents` and `#alerts` Slack channels live with integrations

Risk Management Readiness (see Risk Management — Prerequisites) - [ ] Risk governance structure established (Risk Manager, CCB, steering committee) - [ ] Risk assessment framework documented (likelihood/impact matrix, response policy) - [ ] Risk register created with all 9 risks scored, owned, and mitigations assigned - [ ] Early warning indicators and contingency plans defined for all critical/high risks - [ ] First steering committee risk briefing delivered - [ ] Risk review cadence added to sprint ceremonies calendar

Success Criteria Readiness (see Success Criteria — Prerequisites) - [ ] All success criteria KPIs agreed and signed off by steering committee - [ ] "Adoption" and "target department" precisely defined - [ ] Baseline manual review time measured (10+ educator survey) - [ ] Analytics infrastructure provisioned and capturing events - [ ] Survey templates created (micro-survey, SUS, post-pilot, NPS) - [ ] Gate review criteria documented (Gates 1-4) - [ ] Success dashboard templated - [ ] First academic integrity committee presentation scheduled

---

Sprint 1-2: Foundation & Core Development

Duration: Weeks 1-2

Week	Task	Deliverables
Week 1	Environment Setup	- Setup development environment - Configure version control (Git) - Setup CI/CD pipeline (GitHub Actions) - Create project documentation structure
Week 1	Core Engine Development	- Implement AIDetector class - Develop phrase detection algorithms - Create perplexity calculation module - Build structural analysis functions

Week	Task	Deliverables
Week 2	HE Level Integration	- Implement HE level characteristics (4-8) - Develop critical analysis assessment - Create vocabulary analysis module - Build scoring algorithms
Week 2	Unit Testing	- Write unit tests for core functions - Achieve 80%+ code coverage - Setup automated testing pipeline

Sprint 3-4: API & Web Interface

Duration: Weeks 3-4

Week	Task	Deliverables
Week 3	Flask API Development	- Create REST API endpoints (including `/api/v1/analyse/file`, `/api/v1/analyse/text`, `/api/v1/upload/validate`) - Implement file upload handling (multipart, MIME validation via python-magic, ClamAV scanning, 25 MB limit, filename sanitisation) - Build batch upload endpoint (`/api/v1/batch/upload`, max 50 files / 500 MB) - Setup error handling and logging
Week 3	Database Integration	- Design results storage schema - Implement PostgreSQL connection - Create data persistence layer - Build query optimization
Week 4	Frontend Development	- Design responsive UI/UX - Implement HTML/CSS interface - Create JavaScript analysis client - Build results visualization
Week 4	API Testing	- Write API integration tests - Perform load testing - Test file upload limits (25 MB single, 50 files/500 MB batch, rate limiting, all file types, malware rejection, scanned PDF rejection) - Validate all error responses against File Upload Specification error messages

Sprint 5-6: Advanced Features & Testing

Duration: Weeks 5-6

Week	Task	Deliverables
Week 5	PDF Report Generation	- Implement ReportLab integration - Design professional report template - Add charts and visualizations - Create export functionality

Week	Task	Deliverables
Week 5	Batch Processing	- Implement queue system - Add asynchronous processing - Build progress tracking - Create batch result aggregation
Week 6	User Management	- Implement authentication system - Create role-based access control - Build user dashboard - Add activity logging
Week 6	Security Hardening	- Implement input validation - Add CSRF protection - Configure SSL/TLS - Perform security audit

Sprint 7-8: Integration & UAT

Duration: Weeks 7-8

Week	Task	Deliverables
Week 7	System Integration	- Integrate all modules - End-to-end testing - Performance optimization - Bug fixing and refinement
Week 7	Documentation	- Complete API documentation - Write user manual - Create admin guide - Produce video tutorials
Week 8	UAT Preparation	- Setup UAT environment - Create test scenarios - Prepare test data - Train UAT team
Week 8	User Acceptance Testing	- Conduct UAT sessions - Gather user feedback - Log issues and improvements - Validate against requirements

Sprint 9-10: Production Preparation

Duration: Weeks 9-10

Week	Task	Deliverables
Week 9	Infrastructure Setup	- Configure production servers - Setup load balancers - Configure database replication - Implement backup systems

Week	Task	Deliverables
Week 9	Monitoring & Logging	- Setup application monitoring (Prometheus) - Configure log aggregation (ELK stack) - Implement alerting system - Create monitoring dashboards
Week 10	Performance Testing	- Conduct load testing (1000+ concurrent users) - Perform stress testing - Test failover scenarios - Optimize bottlenecks
Week 10	Deployment Automation	- Configure automated deployment - Setup rollback procedures - Create deployment checklist - Test disaster recovery

Sprint 11-12: Production Launch & Stabilization

Duration: Weeks 11-12

Week	Task	Deliverables
Week 11	Soft Launch	- Deploy to production - Enable for pilot users - Monitor system performance - Provide immediate support
Week 11	Training & Onboarding	- Conduct staff training sessions - Distribute documentation - Setup support channels - Create FAQ resources
Week 12	Full Launch	- Enable for all users - Monitor metrics and usage - Address emerging issues - Gather user feedback
Week 12	Sprint Retrospective	- Review project outcomes - Document lessons learned - Plan future enhancements - Celebrate success

---

Testing Strategy

Testing Prerequisites (Sprint 0)

All testing infrastructure, tooling, and standards must be established **before Sprint 1 begins**. No code should be merged without passing through the quality gates defined here.

Test Tooling & Framework Setup

Tool / Framework	Purpose	Setup Owner	Completion Criteria
pytest	Unit and integration testing	Lead Dev	Config committed, sample test passing in CI
pytest-cov	Code coverage reporting	Lead Dev	Coverage thresholds enforced in CI (fail <80%)
Selenium + WebDriver	End-to-end browser testing	QA Lead	Grid configured for Chrome, Firefox, Safari, Edge
JMeter	Load and stress testing	QA Lead	Test plans templated, baseline scripts committed
OWASP ZAP	Automated security scanning	Security Lead	ZAP baseline scan integrated into CI pipeline
Pact	Contract testing (API consumers/providers)	Lead Dev	Pact broker running, example contract committed
Faker / Factory Boy	Test data generation	QA Lead	Factories for all core models created
Allure / pytest-html	Test reporting and dashboards	QA Lead	Reports auto-generated on every CI run
axe-core	Accessibility testing	Frontend Lead	axe integrated into E2E test suite
Bandit	Python static security analysis	Security Lead	Bandit running on every PR in CI

Test Environment Strategy

Environment	Purpose	Refresh Cadence	Data
Local Dev	Developer unit/ integration testing	Continuous	Synthetic fixtures
CI (GitHub Actions)	Automated test suite on every PR	Per commit/PR	Ephemeral containers, seeded test data
Staging	Integration, E2E, performance, security testing	Weekly rebuild from production-like snapshot	Anonymised production-like data
UAT	User acceptance testing (Weeks 7-8)	Frozen during UAT windows	Curated educator-provided samples
Pre-Production	Final validation before release	Per release candidate	Mirror of production data (anonymised)

Test Data Requirements

Dataset	Minimum Size	Description	Owner
AI-Generated Samples	250 documents	Texts generated by GPT-4, Claude, Gemini, Llama across HE levels 4-8	ML Lead
Human-Written Samples	250 documents	Genuine student submissions across HE levels 4-8 (anonymised, consented)	Academic Liaison
Mixed/ Paraphrased Samples	100 documents	AI-generated text edited by humans, or human text run through paraphrasing tools	ML Lead
Edge Cases	50 documents	Very short texts (<200 words), non-English fragments, heavily cited work, code-heavy submissions	QA Lead
Batch Test Set	500 files	Mixed formats (.docx, .pdf, .txt, .rtf, .odt) for batch upload and processing testing. Must include: oversized files (>25 MB), scanned PDFs, empty files, <50 word files, >50,000 word files, legacy .doc files, non-UTF-8 .txt files, files with special characters in filenames	QA Lead

Quality Gate Definitions

Every PR and release must pass through defined quality gates. No exceptions without CCB approval.

Gate	When Applied	Pass Criteria	Enforced By
PR Gate	Every pull request	Unit tests pass, coverage >=80%, linting clean, Bandit clean, Pact contracts valid	GitHub Actions (blocking)
Sprint Gate	End of each sprint	All sprint acceptance criteria met, no P1/P2 bugs open, integration tests pass	QA Lead sign-off
Security Gate	Sprints 6, 8, 10	OWASP ZAP scan clean (no high/ critical), Bandit clean, manual pen test findings resolved	Security Lead sign-off
Performance Gate	Sprints 4, 7, 10	Response time <2s (p95), analysis <5s (p95), 500+ concurrent users sustained, zero errors under load	QA Lead + DevOps sign-off
Accessibility Gate	Sprints 4, 8	axe-core zero critical/serious violations, manual screen reader test pass, keyboard navigation complete	Frontend Lead + Accessibility reviewer

Gate	When Applied	Pass Criteria	Enforced By
UAT Gate	Sprint 8	80%+ test scenarios passed, no critical defects, stakeholder sign-off obtained	Product Owner sign-off
Release Gate	Sprints 11, 12	All other gates passed, rollback tested, monitoring confirmed, IRP rehearsed	Full team sign-off

---

Test Types & Coverage

Test Type	Scope & Coverage
Unit Tests	- 80%+ code coverage - Test all detection algorithms - Validate scoring functions - Mock external dependencies
Integration Tests	- API endpoint testing - Database interaction validation - File upload/processing flows - Authentication & authorization
End-to-End Tests	- Complete user workflows - Cross-browser compatibility (Chrome, Firefox, Safari, Edge) - Mobile responsiveness - Report generation
Performance Tests	- Load testing: 1000 concurrent users - Stress testing: identify breaking points - Analysis speed benchmarks - Database query optimization
Security Tests	- Penetration testing - OWASP Top 10 validation - SQL injection prevention - XSS vulnerability checks
Accessibility Tests	- WCAG 2.1 AA compliance (axe-core automated) - Screen reader testing (NVDA, VoiceOver) - Keyboard-only navigation - Colour contrast validation
UAT	- Real user scenarios - Accuracy validation with known samples - Usability testing - Stakeholder sign-off

Testing Cadence Per Sprint

Testing is **not deferred** to later sprints. Each sprint has defined testing activities that must complete before the sprint closes.

Sprint	Unit Tests	Integration Tests	E2E Tests	Performance Tests	Security Tests	Accessibility Tests
Sprint 1-2	All new code	DB + core engine	—	Baseline benchmarks	Bandit on CI	—
Sprint 3-4	All new code	API + DB + file upload	Quick Analysis flow	API response times	Bandit + input validation	axe-core on all pages
Sprint 5-6	All new code	Batch + queue + auth	Batch + Report Export flows	Batch processing under load	OWASP ZAP scan + CSRF validation	axe-core + keyboard nav
Sprint 7-8	Regression suite	Full system integration	All user flows, cross-browser	Load test (500 users)	Full pen test	Full WCAG audit
Sprint 9-10	Regression suite	Staging environment validation	Staging E2E	Load test (1000+ users), stress test, failover	OWASP ZAP on staging, secrets scan	—
Sprint 11-12	Regression suite	Production smoke tests	Production E2E (pilot)	Production load validation	Production security scan	Post-launch spot check

Defect Management

Severity	Definition	SLA (Fix Time)	Sprint Impact
P1 — Critical	System unusable, data loss, security breach	4 hours	Sprint halted until resolved
P2 — High	Major feature broken, no workaround	24 hours	Must fix within current sprint
P3 — Medium	Feature partially broken, workaround exists	Current sprint or next	Prioritised in backlog
P4 — Low	Cosmetic, minor UX issue	Next sprint	Added to backlog

Test Reporting & Metrics

The following metrics are tracked and reviewed at every sprint retrospective:

Metric	Target	Red Flag
Code Coverage (unit)	>= 80%	< 75%
Test Pass Rate (CI)	>= 98%	< 95%
Mean Time to Fix (P1/P2)	< 8 hours	> 24 hours
Flaky Test Rate	< 2%	> 5%
Defect Escape Rate (found in UAT/Prod vs total)	< 10%	> 20%
Automated vs Manual Test Ratio	>= 80% automated	< 60% automated
Regression Suite Run Time	< 15 minutes	> 30 minutes

Testing Strategy Exit Criteria (Sprint 0)

Before Sprint 1, the following must be confirmed:

- [ ] All test tooling installed and configured in CI pipeline
- [ ] Test environments provisioned (local, CI, staging)
- [ ] Validation dataset curated and versioned (650+ documents)
- [ ] Quality gate definitions committed to repo and enforced in CI
- [ ] Test data factories created for core models
- [ ] Defect management process agreed and tooling configured (e.g., GitHub Issues labels)
- [ ] Test reporting dashboards live
- [ ] Sprint testing cadence communicated to full team

UX/UI Requirements

UX/UI Prerequisites (Sprint 0)

All design foundations must be established **before Sprint 1 begins**. No frontend code should be written without approved wireframes, a design system, and validated user personas.

User Personas & Research

Persona	Role	Key Goals	Pain Points	HE Level Focus
Dr Sarah Chen	Module Leader	Quickly check suspicious submissions, generate evidence for misconduct panels	Time-poor, handles 200+ submissions per module, needs clear yes/no guidance	Levels 5-6

Persona	Role	Key Goals	Pain Points	HE Level Focus
Prof James Okafor	Programme Director	Review department-wide trends, batch-analyse cohort submissions	Needs aggregated views, comparative data across modules, exportable reports	Levels 6-8
Emily Torres	Academic Integrity Officer	Investigate flagged cases, produce formal reports for disciplinary hearings	Needs audit trail, detailed evidence breakdown, legally defensible outputs	All levels
Dr Raj Patel	New Lecturer	Understand what AI-generated work looks like, learn to set better assessments	Unfamiliar with AI tools, needs educational context alongside detection results	Level 4
IT Admin (Alex)	System Administrator	Manage users, monitor system health, handle access requests	Needs clear admin dashboard, bulk user management, integration with institutional SSO	N/A

Action	Owner	Completion Criteria
Conduct user research interviews with 5+ educators across at least 3 HE levels	UX Lead	Interview notes and affinity map completed
Validate personas with academic stakeholders	UX Lead / Product Owner	Personas signed off
Map user journeys for each persona (happy path + error paths)	UX Lead	Journey maps documented and reviewed
Conduct competitor analysis (Turnitin AI, GPTZero, Originality.ai)	UX Lead	Comparison matrix with strengths/weaknesses

Design System & Component Library

Component	Standard	Owner	Completion Criteria
Colour Palette	Primary, secondary, semantic (success/ warning/error/info), WCAG AA contrast ratios verified	UX Lead	Palette documented with hex/RGB values and contrast ratios
Typography	System font stack (fallback-safe), heading hierarchy (h1-h4), body/caption sizes, minimum 16px body	UX Lead	Type scale documented
Spacing & Grid	8px base unit, 12-column responsive grid, breakpoints at 320px / 768px / 1024px / 1440px	UX Lead	Grid spec documented

Component	Standard	Owner	Completion Criteria
Component Library	Buttons, inputs, selects, modals, cards, tables, alerts, progress indicators (linear + circular), file upload zone (single drag-and-drop), batch upload zone (multi-file drag-and-drop), file list table with per-file status/actions, toast notifications	Frontend Lead	Components built in HTML/CSS, documented with usage guidelines
Iconography	Consistent icon set (e.g., Lucide, Phosphor), minimum 24x24 touch target	UX Lead	Icon set selected and documented
Data Visualisation	Chart styles for confidence scores, HE level breakdowns, batch summaries (colour-blind safe palettes)	UX Lead	Chart style guide with accessible colour variants

Wireframes & Prototyping

Deliverable	Fidelity	Tool	Review Process	Completion Criteria
Information Architecture (sitemap)	—	Miro / FigJam	Full team review	IA approved by Product Owner
Low-fidelity wireframes (all pages)	Low	Figma / Balsamiq	UX Lead + Product Owner	Wireframes covering all user flows
Interactive prototype (key flows)	Medium	Figma	Usability testing with 3+ educators	Prototype tested, feedback incorporated
High-fidelity mockups	High	Figma	Stakeholder sign-off	Mockups approved for all pages
Responsive variants	High	Figma	Frontend Lead review	Mobile (320px), tablet (768px), desktop (1440px) variants

---

Design Principles

1. **Simplicity:** Minimize clicks to complete analysis (target: 3 clicks or fewer)
2. **Accessibility:** WCAG 2.1 AA compliance for inclusive design
3. **Clarity:** Clear visual hierarchy and intuitive information architecture
4. **Responsiveness:** Mobile-first design supporting devices 320px+
5. **Performance:** Page load under 2 seconds, analysis results under 5 seconds
6. **Trust & Transparency:** Always show confidence levels, never present results as absolute — educators make the final judgement
7. **Progressive Disclosure:** Show summary results first, allow drill-down into detailed analysis on demand

8. **Error Prevention:** Validate inputs before submission, provide clear feedback on unsupported file types or sizes

Key User Flows

User Flow	Steps
Quick Analysis	1. Navigate to homepage 2. Paste text OR upload file 3. Select HE level 4. Click Analyze 5. View results instantly
Batch Analysis	1. Navigate to batch processing 2. Upload multiple files 3. Configure analysis settings 4. Submit batch 5. Download aggregated results
Report Export	1. Complete analysis 2. Review results 3. Select export format (PDF/JSON) 4. Download with one click
Admin: User Management	1. Navigate to admin dashboard 2. View/search users 3. Add/edit/deactivate accounts 4. Assign roles (Admin/Educator/Reviewer) 5. Confirm changes
Error Recovery	1. User encounters error (upload fails, analysis times out) 2. Clear error message with reason displayed 3. Actionable suggestion shown (retry, reduce file size, contact support) 4. One-click retry available

File Upload Specification

File upload is a core user interaction across both single and batch analysis. The following defines the complete upload experience, constraints, validation, and error handling.

Supported File Types

File Type	MIME Type	Extension	Max Size	Notes
Microsoft Word	`application/vnd.openxmlformats-officedocument.wordprocessingml.document`	`.docx`	25 MB	Primary academic submission format. `.doc` (legacy) not supported — prompt user to save as `.docx`
PDF	`application/pdf`	`.pdf`	25 MB	Text-based PDFs only. Scanned/ image-only PDFs rejected with clear message to use OCR first
Plain Text	`text/plain`	`.txt`	10 MB	UTF-8 encoding assumed. Non-UTF-8 files re-encoded with warning
Rich Text	`application/rtf`	`.rtf`	15 MB	Converted to plain text server-side before analysis
OpenDocument Text	`application/vnd.oasis.opendocument.text`	`.odt`	25 MB	Common in open-source environments used by some institutions

Explicitly unsupported (with user-facing message): `.doc` (legacy Word), `.pages` (Apple), `.html`, `.epub`, image files (`.jpg`, `.png`), spreadsheets, presentations, archives (`.zip`, `.rar`)

Document Size Limits

Constraint	Limit	Behaviour When Exceeded
Max file size (single upload)	25 MB	Upload rejected client-side before transmission. Error: "File exceeds the 25 MB limit. Please reduce the file size or split into smaller documents."
Max file size (batch per file)	25 MB per file	Individual file flagged in file list. Other files in batch proceed normally
Max word count (per document)	50,000 words	Analysis proceeds but with warning: "This document exceeds 50,000 words. Analysis may take longer than usual." Timeout extended to 30 seconds
Min word count (per document)	50 words	Analysis proceeds but with warning: "This document contains fewer than 50 words. Results may be less reliable for very short texts."
Recommended word count	500 - 5,000 words	Optimal accuracy range. No warning displayed
Max batch size (file count)	50 files per batch	Upload rejected if >50 files selected. Error: "Maximum 50 files per batch. Please split your submission into multiple batches."
Max batch size (total)	500 MB total	Upload rejected if total exceeds 500 MB. Error: "Total batch size exceeds 500 MB. Please reduce the number or size of files."

Single File Upload (Homepage — Sprint 4)

Component	Specification
Upload zone	Combined drag-and-drop zone + click-to-browse button. Dashed border, icon, "Drag a file here or click to browse" label
Drag-and-drop	Supported on Homepage (not just Batch page). Drop zone highlights on drag-over (border colour change, background tint). Visual feedback: "Drop file to upload"
Text paste alternative	Large text area adjacent to / tabbed with upload zone. Labelled "Paste text directly". Auto-detects word count on paste. Mutually exclusive with file upload (selecting one clears the other)
File preview	After selection: file name, file size, file type icon displayed. "Remove" button to clear selection
Progress indicator	Circular progress spinner during upload + "Uploading..." label. For files >5 MB: percentage progress bar
Client-side validation	Before upload: file type checked against allowed list, file size checked against 25 MB limit. Instant error display (no server round-trip)
Server-side validation	After upload: MIME type verified (not just extension), file scanned for malware (ClamAV or equivalent), text extracted and word count checked

Component	Specification
Keyboard accessible	Upload zone focusable via Tab. Enter/Space opens file browser. Drop zone announces state changes to screen readers
Mobile behaviour	Upload zone triggers native file picker. Camera/photo library options suppressed (document types only). Full-width button: "Upload Document"

Batch Upload (Batch Page — Sprint 5)

Component	Specification
Drag-and-drop zone	Large drop zone (minimum 200px height). Accepts multiple files simultaneously. "Drag files here or click to browse" with supported formats listed
Multi-file selection	File browser opens with `multiple` attribute enabled. Shift-click and Ctrl-click selection supported natively
File list	Sortable table showing: file name, file type (icon), file size, word count (after server-side extraction), status (queued / uploading / processing / complete / error)
Per-file status indicators	Queued: grey clock icon. Uploading: blue progress bar. Processing: blue spinner. Complete: green tick. Error: red cross with tooltip showing error reason
Per-file actions	"Remove" button per file (before submission). "Retry" button per file (after error). "View result" link per file (after completion)
Batch progress	Overall progress bar: "X of Y files processed". Estimated time remaining displayed after first file completes
Batch settings	HE level selector (applies to all files, or per-file override). Option: "Use same HE level for all files" (default) or "Set per file"
Submit behaviour	"Start Analysis" button disabled until >=1 valid file uploaded. Confirmation if >20 files: "You are about to analyse X files. This may take approximately Y minutes. Continue?"
Background processing	After submission, user can navigate away. Notification banner: "Batch analysis in progress — X of Y complete". Results available on Batch Results page
Partial failure handling	If some files fail: batch continues. Failed files shown with error reason. "Retry failed files" button at batch level

File Validation & Error Messages

Validation	Check Point	Error Message	UX Behaviour
Unsupported file type	Client-side (immediate)	"**{filename}** is not a supported file type. Please upload a .docx, .pdf, .txt, .rtf, or .odt file."	File not added to upload queue. Toast notification (auto-dismiss 8s)

Validation	Check Point	Error Message	UX Behaviour
File too large	Client-side (immediate)	"**{filename}** ({filesize} MB) exceeds the 25 MB limit. Please reduce the file size or split into smaller documents."	File not added to upload queue. Toast notification
Legacy .doc format	Client-side (immediate)	"**{filename}** is a legacy Word format (.doc). Please open the file in Word and save as .docx, then try again."	File not added to upload queue. Toast notification with help link
Batch too many files	Client-side (immediate)	"You selected {count} files, but the maximum is 50 per batch. Please reduce the number of files."	Excess files not added. First 50 shown in file list
Batch total too large	Client-side (after selection)	"Total batch size ({totalsize} MB) exceeds the 500 MB limit. Please remove some files to continue."	Submit button disabled. Files highlighted that could be removed
MIME type mismatch	Server-side	"**{filename}** appears to be a different file type than its extension suggests. Please check the file and re-upload."	File marked as error in file list. Retry available
Scanned/ image PDF	Server-side (text extraction)	"**{filename}** appears to be a scanned document with no extractable text. Please use OCR software to convert it to a text-based PDF first."	File marked as error. Help link to OCR guidance
Empty document	Server-side (text extraction)	"**{filename}** contains no readable text. Please check the file is not empty or corrupted."	File marked as error
Below min word count	Server-side (post-extraction)	"**{filename}** contains only {count} words. Results may be less reliable for texts under 50 words."	Warning banner on results (not blocking). Analysis proceeds
Above max word count	Server-side (post-extraction)	"**{filename}** contains {count} words (over 50,000). Analysis will proceed but may take longer."	Warning banner. Extended timeout applied
Malware detected	Server-side (scan)	"**{filename}** could not be processed for security reasons. Please check the file and try again, or contact support."	File rejected. No detail given to user (security). Logged as P2 alert
Upload network failure	Client-side (during transfer)	"Upload failed for **{filename}**. Please check your internet connection and try again."	Retry button displayed. File remains in queue

Validation	Check Point	Error Message	UX Behaviour
Server processing timeout	Server-side	"Analysis of **{filename}** timed out. This can happen with very large or complex documents. Please try again or contact support."	Retry button. Logged for investigation

Upload API Endpoints

Endpoint	Method	Purpose	Auth Required	Rate Limit
`/api/v1/analyse/text`	POST	Submit plain text for analysis	Yes	100 req/min per user
`/api/v1/analyse/file`	POST	Upload single file for analysis	Yes	10 files/min per user
`/api/v1/batch/upload`	POST	Upload multiple files for batch processing	Yes	5 batches/hour per user
`/api/v1/batch/{batch_id}/status`	GET	Check batch processing status	Yes	60 req/min per user
`/api/v1/batch/{batch_id}/results`	GET	Retrieve batch results	Yes	30 req/min per user
`/api/v1/batch/{batch_id}/retry`	POST	Retry failed files in a batch	Yes	10 req/hour per user
`/api/v1/upload/validate`	POST	Client-side pre-validation (file type, size check without full upload)	Yes	100 req/min per user

Request format for `/api/v1/analyse/file`: - Content-Type: `multipart/form-data` - Fields: `file` (binary), `he_level` (integer, 4-8), `options` (JSON, optional)

Request format for `/api/v1/batch/upload`: - Content-Type: `multipart/form-data` - Fields: `files[]` (multiple binary), `he_level` (integer, 4-8 — default for all), `file_he_levels` (JSON map of filename→level, optional override)

Upload Security

Measure	Implementation
File type verification	Server-side MIME type check via `python-magic` (libmagic). Extension-only checks insufficient
Malware scanning	All uploaded files scanned with ClamAV before processing. Quarantined if detected
Filename sanitisation	Strip path components, special characters, and Unicode normalise. Generate internal UUID filename. Original name stored in metadata only

Measure	Implementation
Temporary storage	Uploaded files stored in encrypted temporary directory (`/tmp/uploads/{uuid}/`). Deleted after processing or within 1 hour (whichever is sooner)
File size enforcement	Nginx `client_max_body_size` set to 30 MB (buffer above 25 MB limit). Flask `MAX_CONTENT_LENGTH` set to 25 MB. Gunicorn `--limit-request-line` configured
Content-Type enforcement	`multipart/form-data` required. Reject requests with unexpected content types
Anti-virus signature updates	ClamAV signature database updated daily (automated via `freshclam`)

Page Inventory & Layout Requirements

Page	Primary Purpose	Key Components	Priority
Homepage / Dashboard	Entry point, quick analysis	Text paste area OR drag-and-drop file upload zone (mutually exclusive, tabbed), file preview with name/size/type, HE level selector (4-8), recent analyses list	P1 — Sprint 4
Results View	Display analysis outcome	Confidence score (visual gauge), category breakdown, phrase highlights, HE level assessment, export button	P1 — Sprint 4
Batch Upload	Multi-file processing	Large drag-and-drop zone (multi-file), file list table (name, type, size, word count, status, per-file actions), overall progress bar with ETA, HE level selector (global or per-file), batch settings, submit/retry controls	P1 — Sprint 5
Batch Results	Aggregated outcomes	Summary statistics, sortable/filterable table, bulk export	P1 — Sprint 5
PDF Report Preview	Pre-export review	Report layout preview, customisation options (include/exclude sections), download button	P2 — Sprint 5
User Dashboard	Personal history	Analysis history (paginated), saved reports, account settings	P2 — Sprint 6
Admin Dashboard	System management	User management table, system health metrics, activity logs, role management	P2 — Sprint 6
Login / Auth	Authentication	Email/password, MFA challenge, SSO button, password reset	P1 — Sprint 6
Help / Documentation	Self-service support	Searchable FAQ, user guide, video tutorials, contact support	P3 — Sprint 7

Page	Primary Purpose	Key Components	Priority
Error Pages	Graceful failure	404, 500, maintenance pages with clear messaging and navigation back	P2 — Sprint 4

Accessibility Requirements (WCAG 2.1 AA)

Accessibility is **not optional** and must be built in from the first line of frontend code. The following standards are non-negotiable.

Perceivable

Requirement	Standard	Validation Method
Colour contrast (normal text)	Minimum 4.5:1 ratio	axe-core automated check
Colour contrast (large text)	Minimum 3:1 ratio	axe-core automated check
Non-text content	All images, icons, charts have meaningful alt text	Manual review + axe-core
Colour not sole indicator	Status/results never communicated by colour alone — use icons, labels, patterns	Manual review
Text resize	Content readable and functional at 200% zoom	Manual browser zoom test
Captions	Video tutorials include closed captions	Manual review

Operable

Requirement	Standard	Validation Method
Keyboard navigation	All interactive elements reachable and operable via keyboard	Manual tab-through test
Focus indicators	Visible focus ring on all interactive elements (minimum 2px, high contrast)	Manual + axe-core
Skip navigation	"Skip to main content" link on every page	Manual review
Touch targets	Minimum 44x44px for all interactive elements on mobile	Design review + manual test
No keyboard traps	User can always tab out of any component (modals include close on Escape)	Manual test
Motion	Animations respect `prefers-reduced-motion` media query	Code review

Understandable

Requirement	Standard	Validation Method
Language attribute	`lang="en-GB"` set on HTML element	axe-core
Error identification	Form errors identified in text, linked to the field, not colour-only	Manual + axe-core
Consistent navigation	Navigation structure identical across all pages	Manual review
Plain language	Results and guidance written at reading age 14 or below (Flesch-Kincaid)	Content review
Abbreviations	All acronyms (HE, RBAC, MFA) expanded on first use per page	Content review

Robust

Requirement	Standard	Validation Method
Valid HTML	No parsing errors in W3C validator	CI automated check
ARIA usage	ARIA roles/labels used correctly, not redundantly	axe-core + manual review
Screen reader compatibility	Tested with NVDA (Windows) and VoiceOver (macOS)	Manual testing (Sprints 4, 8)

Responsive Design Breakpoints

Breakpoint	Target Device	Layout Adjustments
320px — 767px	Mobile (portrait)	Single column, stacked cards, hamburger menu, full-width inputs, bottom-anchored CTA
768px — 1023px	Tablet	Two-column layout, collapsible sidebar, inline form elements
1024px — 1439px	Laptop / Small desktop	Full sidebar navigation, three-column results view, inline batch table
1440px+	Large desktop	Max-width container (1280px centred), expanded data tables, side-by-side comparison views

UX/UI Cadence Per Sprint

UX/UI work is **not confined to Sprint 4**. Design, testing, and refinement occur throughout.

Sprint	UX/UI Activities
Sprint 0	User research, personas, competitor analysis, design system, wireframes, prototype usability testing
Sprint 1-2	No frontend — UX Lead refines high-fidelity mockups based on prototype feedback, prepares component specs
Sprint 3-4	Build component library, implement Homepage + Results View + Error Pages, first axe-core pass, educator demo session
Sprint 5-6	Implement Batch Upload/Results, PDF Preview, User Dashboard, Admin Dashboard, Login/Auth pages, keyboard nav audit
Sprint 7-8	Full cross-browser testing, screen reader testing, usability testing with 5+ educators (all personas), iterate on feedback
Sprint 9-10	Performance optimisation (Lighthouse scores), responsive QA across all breakpoints, final WCAG audit
Sprint 11-12	Pilot user feedback collection, quick-fix UX iterations, help/documentation page, post-launch UX metrics baseline

UX Metrics & Success Criteria

Metric	Target	Measurement Method	When Measured
Task completion rate (Quick Analysis)	>= 90%	Usability testing	Sprints 4, 8, 12
Time to first analysis	< 60 seconds	Usability testing	Sprints 4, 8, 12
System Usability Scale (SUS) score	>= 72 (above average)	Post-task questionnaire	Sprints 8, 12
Error rate (user mistakes per task)	< 10%	Usability testing	Sprints 4, 8, 12
Accessibility score (Lighthouse)	>= 95	Lighthouse CI	Every sprint from Sprint 4
WCAG violations (critical + serious)	0	axe-core	Every sprint from Sprint 4
User satisfaction (post-pilot survey)	>= 4.2 / 5	Likert scale survey	Sprint 12
Mobile usability score	>= 90 (Lighthouse)	Lighthouse CI	Every sprint from Sprint 4

UX/UI Exit Criteria (Sprint 0)

Before Sprint 1, the following must be confirmed:

- [ ] User research completed (5+ educator interviews)
- [ ] Personas validated and signed off by stakeholders

- [ ] Competitor analysis documented
- [ ] Design system defined (colours, typography, spacing, grid, icons)
- [ ] Low-fidelity wireframes approved for all pages
- [ ] Interactive prototype usability-tested with 3+ educators
- [ ] High-fidelity mockups approved for Sprint 3-4 pages (Homepage, Results, Error)
- [ ] Responsive breakpoints and layout rules documented
- [ ] Accessibility standards documented and communicated to all developers
- [ ] Component library specs ready for frontend build

Production Requirements

Production Prerequisites (Sprint 0)

All infrastructure decisions, security policies, and operational procedures must be defined **before Sprint 1 begins**. Production readiness is not a Sprint 9 activity — it starts on day one.

Cloud Provider & Account Setup

Action	Owner	Completion Criteria
Select cloud provider (AWS / Azure / GCP) and document rationale in Architecture Decision Record	DevOps Lead / Tech Lead	ADR approved by stakeholders
Create separate AWS/Azure/GCP accounts or resource groups for dev, staging, and production	DevOps Lead	Accounts provisioned with billing alerts
Configure IAM policies — least-privilege access for all team members and service accounts	DevOps Lead / Security Lead	IAM audit log enabled, no root account usage
Establish Infrastructure as Code (IaC) repository using Terraform or Pulumi	DevOps Lead	IaC repo created, base modules committed
Register production domain and configure DNS	DevOps Lead	Domain registered, DNS records pointing to staging placeholder
Procure TLS certificates (wildcard or per-subdomain via Let's Encrypt / ACM)	DevOps Lead	Certificates issued and auto-renewal configured

Network Architecture

Component	Design Decision	Specification
VPC / Virtual Network	Isolated network per environment	Separate VPCs for dev, staging, production with no peering between dev and prod

Component	Design Decision	Specification
Subnets	Public / private separation	Public: load balancer, CDN origin. Private: app servers, database, cache, queue
Security Groups / NSGs	Least-privilege network rules	Ingress: 443 (HTTPS) only on public subnet. App servers: only from LB. DB: only from app servers
NAT Gateway	Outbound internet for private subnets	NAT gateway in each AZ for app server outbound (package updates, API calls)
Bastion / Jump Host	Secure SSH access	Single bastion host with MFA, session logging, auto-shutdown after 30 minutes
DNS	Internal service discovery	Private hosted zone for internal service names (e.g., `db.internal`, `cache.internal`)

Disaster Recovery & Business Continuity

Scenario	Strategy	RPO (Recovery Point Objective)	RTO (Recovery Time Objective)
Single server failure	Auto-scaling group replaces instance automatically	0 (no data on app servers)	< 2 minutes
Database failure (primary)	Automated failover to read replica promoted to primary	< 1 minute (synchronous replication)	< 5 minutes
Availability Zone outage	Multi-AZ deployment — LB routes to healthy AZ	0	< 2 minutes
Region-level outage	Cross-region DB backup restore, DNS failover to secondary region	< 1 hour (async replication)	< 4 hours
Data corruption / accidental deletion	Point-in-time recovery from automated DB backups + S3 versioning	< 5 minutes (PITR)	< 30 minutes
Ransomware / security breach	Immutable backups in separate account, incident response plan activation	< 1 hour	< 4 hours (clean rebuild)

Action	Owner	Completion Criteria
Document DR plan covering all scenarios above	DevOps Lead / Security Lead	DR plan approved by stakeholders
Configure automated database backups (daily full, continuous WAL archiving)	DevOps Lead	Backups verified with test restore
Configure S3 versioning and lifecycle policies	DevOps Lead	Versioning enabled, lifecycle rules set (30-day transition to IA, 90-day to Glacier)

Action	Owner	Completion Criteria
Setup cross-region backup replication	DevOps Lead	Replication verified, restore tested
Schedule quarterly DR drill (failover test)	DevOps Lead / Project Manager	First drill date in calendar

---

Infrastructure Specifications

Component	Specification
Application Servers	3x instances: 8 vCPU, 16GB RAM, 100GB SSD (auto-scaling enabled)
Database Server	Primary + Replica: 16 vCPU, 32GB RAM, 500GB SSD, automated backups
Cache Server	Redis cluster: 3 nodes, 8GB RAM each
Load Balancer	HA proxy with SSL termination, health checks, session persistence
Storage	S3-compatible object storage: 1TB, encrypted, versioned
CDN	CloudFlare or equivalent for static assets and DDoS protection

Auto-Scaling Policy

Metric	Scale-Out Trigger	Scale-In Trigger	Min Instances	Max Instances	Cooldown
CPU Utilisation	> 70% for 3 minutes	< 30% for 10 minutes	3	10	5 minutes
Memory Utilisation	> 75% for 3 minutes	< 35% for 10 minutes	3	10	5 minutes
Request Queue Depth	> 100 pending requests	< 10 pending requests	3	10	3 minutes

Database Configuration

Parameter	Primary	Replica	Justification
Instance	16 vCPU, 32GB RAM, 500GB SSD	16 vCPU, 32GB RAM, 500GB SSD	Matched specs for seamless failover
PostgreSQL Version	15.x (latest stable)	15.x (latest stable)	LTS, strong JSON support, parallel queries
Replication	Synchronous streaming	—	Zero data loss on failover

Parameter	Primary	Replica	Justification
Connection Pooling	PgBouncer (transaction mode, max 200 connections)	PgBouncer (max 100 connections)	Prevent connection exhaustion from app servers
Backups	Daily full + continuous WAL archiving	—	PITR to any second within retention window
Backup Retention	30 days	—	Meets GDPR audit trail and DR requirements
Maintenance Window	Sunday 02:00-04:00 UTC	Sunday 04:00-06:00 UTC	Lowest traffic period
Encryption	AES-256 at rest, TLS 1.3 in transit	AES-256 at rest, TLS 1.3 in transit	Compliance requirement

Cache Configuration (Redis)

Parameter	Specification	Justification
Topology	3-node cluster (1 primary + 2 replicas per shard)	HA with automatic failover
Memory	8GB per node	Sufficient for session data, analysis result caching, rate limiting
Eviction Policy	`allkeys-lru`	Least recently used eviction prevents OOM
Persistence	AOF (appendonly) with 1-second fsync	Durability for session data without significant performance cost
Max TTL (analysis cache)	24 hours	Balance freshness vs performance
Max TTL (session data)	30 minutes	Matches session timeout policy
TLS	Enabled (in-transit encryption)	Compliance requirement

Container & Orchestration

Component	Specification	Justification
Container Runtime	Docker 24.x	Industry standard, well-supported
Orchestration	Kubernetes 1.28+ (managed: EKS / AKS / GKE)	Auto-healing, rolling deploys, auto-scaling
Namespace Isolation	Separate namespaces: `ai-detection-dev`, `ai-detection-staging`, `ai-detection-prod`	Environment isolation within cluster

Component	Specification	Justification
Resource Limits	CPU: 2 cores request / 4 cores limit per pod. Memory: 4GB request / 8GB limit per pod	Prevent noisy-neighbour and OOM kills
Health Checks	Liveness: `/health` (10s interval). Readiness: `/ready` (5s interval)	Automatic restart of unhealthy pods, traffic routing to ready pods only
Rolling Update Strategy	`maxSurge: 1`, `maxUnavailable: 0`	Zero-downtime deployments
Pod Disruption Budget	`minAvailable: 2`	Maintain availability during node maintenance
Secrets Management	Kubernetes Secrets backed by AWS Secrets Manager / Azure Key Vault / GCP Secret Manager	No secrets in environment variables or code
Image Registry	Private container registry (ECR / ACR / GCR) with vulnerability scanning	Prevent supply chain attacks

---

Security Requirements

Authentication & Authorization

- Multi-factor authentication (MFA) for admin accounts
- Role-based access control (RBAC): Admin, Educator, Reviewer
- Session timeout after 30 minutes of inactivity

Data Protection

- End-to-end encryption for data in transit (TLS 1.3)
- Encryption at rest for sensitive data (AES-256)
- GDPR compliance for personal data handling
- Automated data retention and deletion policies

RBAC Permission Matrix

Permission	Admin	Educator	Reviewer	Unauthenticated
Analyse single document	Yes	Yes	Yes	No
Analyse batch	Yes	Yes	No	No
Export PDF report	Yes	Yes	Yes	No
View own analysis history	Yes	Yes	Yes	No
View department-wide results	Yes	No	Yes	No
Manage users (create/edit/deactivate)	Yes	No	No	No

Permission	Admin	Educator	Reviewer	Unauthenticated
Manage roles	Yes	No	No	No
View system health dashboard	Yes	No	No	No
View audit logs	Yes	No	Yes	No
Configure system settings	Yes	No	No	No
Access API (programmatic)	Yes	Yes (own scope)	Yes (own scope)	No

GDPR Compliance Checklist

Requirement	Implementation	Owner	Sprint
Lawful basis for processing	Legitimate interest assessment documented (academic integrity)	DPO	Sprint 0
Privacy notice	Displayed at first login and accessible from footer on every page	UX Lead / DPO	Sprint 6
Data Subject Access Requests (DSAR)	Admin tool to export all data for a given user within 30 days	Lead Dev	Sprint 6
Right to erasure	Admin tool to permanently delete user data and analysis history	Lead Dev	Sprint 6
Data minimisation	Only store data necessary for analysis — no raw submission text retained after 90 days	Lead Dev	Sprint 3
Data Processing Agreement	DPA signed with cloud provider and any third-party processors	DPO / Project Manager	Sprint 0
Breach notification	Automated alerting to DPO within 24 hours; DPO notifies ICO within 72 hours	Security Lead / DPO	Sprint 0 (IRP)
Data Protection Impact Assessment	DPIA completed and reviewed annually	DPO	Sprint 0
International transfers	Confirm all data remains within UK/EEA or adequate safeguards in place	DPO	Sprint 0
Record of processing activities	Maintained in compliance register	DPO	Sprint 0

Security Hardening Checklist

Category	Requirement	Validation	Sprint
HTTP Headers	`Strict-Transport-Security`, `Content-Security-Policy`, `X-Content-Type-Options`, `X-Frame-Options`, `Referrer-Policy`	Automated header scan (SecurityHeaders.com)	Sprint 6

Category	Requirement	Validation	Sprint
Input Validation	All user inputs validated server-side: file type (MIME via `python-magic`), size (25 MB/file, 500 MB/batch), word count (50-50,000), malware scan (ClamAV). Filename sanitised (UUID). Parameterised queries only (no string concatenation in SQL)	Bandit + manual code review	Sprint 3 onwards
CSRF Protection	CSRF tokens on all state-changing forms. `SameSite=Strict` on session cookies	OWASP ZAP scan	Sprint 6
Rate Limiting	API: 100 requests/minute per user. Single file upload: 10 files/minute per user. Batch upload: 5 batches/hour per user (max 50 files/batch). Login: 5 attempts/15 minutes then lockout	Load test + manual verification	Sprint 3
Dependency Scanning	Automated vulnerability scanning on all Python and JavaScript dependencies (Dependabot / Snyk)	CI pipeline check on every PR	Sprint 1 onwards
Container Security	Base images from trusted sources only. No `latest` tags. Run as non-root user. Read-only filesystem where possible	Trivy scan in CI	Sprint 1 onwards
Logging & Audit	All authentication events, role changes, data exports, and admin actions logged with timestamp, user ID, IP address	Log review + alerting rules	Sprint 6
Secrets Management	No secrets in code, environment variables, or Docker images. All secrets via Secrets Manager with rotation	Manual audit + automated scan (gitleaks)	Sprint 1 onwards
Penetration Testing	External pen test by independent third party before production launch	Pen test report with zero critical/high findings	Sprint 10

CI/CD Pipeline Architecture

Pipeline Stages

Stage	Trigger	Actions	Gate (must pass)
1. Build	PR opened / commit pushed	Lint (flake8, eslint), type check, Docker image build	Zero lint errors

Stage	Trigger	Actions	Gate (must pass)
2. Unit Test	Build passes	pytest with coverage, frontend unit tests	Coverage >= 80%, all tests pass
3. Security Scan	Unit tests pass	Bandit (Python), gitleaks (secrets), Trivy (container), Dependabot (dependencies)	Zero critical/high findings
4. Contract Test	Security scan passes	Pact contract verification	All contracts satisfied
5. Integration Test	Contract tests pass	API integration tests against ephemeral DB/Redis containers	All tests pass
6. Deploy to Staging	PR merged to `main`	Automated deploy to staging via Helm/Kustomize	Health checks pass
7. E2E Test	Staging deploy completes	Selenium E2E suite against staging	All critical paths pass
8. Performance Test	E2E passes (sprint cadence)	JMeter load test against staging	Response time < 2s (p95)
9. Approval Gate	All automated gates pass	Manual approval by Tech Lead for production deploy	Approval granted
10. Deploy to Production	Approval granted	Blue-green or canary deployment via Kubernetes	Health checks pass, error rate < 0.5%
11. Post-Deploy Validation	Production deploy completes	Smoke tests, monitoring dashboard check, rollback readiness confirmed	No anomalies in 15-minute window

Deployment Strategy

Aspect	Specification
Strategy	Blue-green deployment (zero-downtime) with canary option for high-risk releases
Rollback	Automated rollback if error rate > 2% or p95 response time > 5s within 10 minutes of deploy
Deployment Window	Any time (zero-downtime), but prefer Tuesday-Thursday 10:00-16:00 UTC for team availability
Feature Flags	LaunchDarkly or equivalent for gradual feature rollout and kill switches
Database Migrations	Forward-compatible only (no destructive migrations). Run as pre-deploy hook. Tested in staging first

Aspect	Specification
Deployment Frequency Target	At least once per sprint (fortnightly), daily capability once stabilised

---

Infrastructure as Code (IaC) Standards

Standard	Requirement
Tool	Terraform (or Pulumi) — all infrastructure defined in code, no manual console changes
State Management	Remote state backend (S3 + DynamoDB lock / Terraform Cloud) with encryption
Module Structure	Reusable modules for VPC, compute, database, cache, monitoring — shared across environments
Environment Parity	Dev, staging, and production defined as separate Terraform workspaces with variable overrides
Code Review	All IaC changes require PR review by DevOps Lead before merge
Drift Detection	Weekly automated plan to detect manual drift — alert if differences found
Tagging	All resources tagged: `Environment`, `Project`, `Owner`, `CostCentre`, `ManagedBy=terraform`

Monitoring & Maintenance

Monitoring & Maintenance Prerequisites (Sprint 0)

Monitoring is **not a Sprint 9 activity**. Observability infrastructure must be provisioned in the dev environment from day one and progressively enhanced. All tooling, standards, and operational processes must be defined **before Sprint 1 begins**.

Observability Tooling Setup

Tool	Purpose	Layer	Setup Owner	Completion Criteria
Prometheus	Metrics collection and time-series storage	Infrastructure + Application	DevOps Lead	Scraping app and node metrics in dev, retention configured
Grafana	Dashboards and visualisation	All layers	DevOps Lead	Connected to Prometheus, base dashboards templated

Tool	Purpose	Layer	Setup Owner	Completion Criteria
Elasticsearch	Log indexing and search	Application + Security	DevOps Lead	Cluster running in dev, index templates defined
Logstash / Fluentd	Log shipping and transformation	Application + Infrastructure	DevOps Lead	Shipping structured JSON logs from app containers to Elasticsearch
Kibana	Log exploration and visualisation	Application + Security	DevOps Lead	Connected to Elasticsearch, saved searches for common queries
PagerDuty	On-call scheduling and incident alerting	Operations	Project Manager	Account provisioned, schedules created, test page sent
Jaeger / OpenTelemetry	Distributed tracing	Application	Lead Dev	Tracing SDK integrated, traces visible in dev
Uptime Robot / Pingdom	Synthetic uptime monitoring (external)	Availability	DevOps Lead	Checks configured for production domain (activated at launch)
Sentry	Application error tracking and grouping	Application	Lead Dev	SDK integrated, source maps uploaded, alert rules set
AWS Cost Explorer / CloudHealth	Cloud spend monitoring	Financial	DevOps Lead / Project Manager	Budget alerts configured, weekly cost report automated

Application Instrumentation Requirements

Every application component must expose standard metrics **from Sprint 1**. No code is considered complete without instrumentation.

Metric Category	Specific Metrics	Format	Endpoint
HTTP Request	`http_requests_total` (counter, labels: method, path, status), `http_request_duration_seconds` (histogram, labels: method, path)	Prometheus	`/metrics`

Metric Category	Specific Metrics	Format	Endpoint
Analysis Engine	`analysis_requests_total` (counter, labels: he_level, source_type), `analysis_duration_seconds` (histogram, labels: he_level), `analysis_document_words` (histogram)	Prometheus	`/metrics`
Queue / Batch	`batch_jobs_queued` (gauge), `batch_jobs_processing` (gauge), `batch_jobs_completed_total` (counter), `batch_job_duration_seconds` (histogram)	Prometheus	`/metrics`
Database	`db_query_duration_seconds` (histogram, labels: query_type), `db_connection_pool_active` (gauge), `db_connection_pool_idle` (gauge)	Prometheus	`/metrics`
Cache	`cache_hits_total` (counter), `cache_misses_total` (counter), `cache_operation_duration_seconds` (histogram)	Prometheus	`/metrics`
Authentication	`auth_login_attempts_total` (counter, labels: result), `auth_active_sessions` (gauge), `auth_mfa_challenges_total` (counter, labels: result)	Prometheus	`/metrics`
File Upload	`upload_requests_total` (counter, labels: file_type, result), `upload_file_size_bytes` (histogram), `upload_duration_seconds` (histogram)	Prometheus	`/metrics`

Distributed Tracing Standards

Standard	Specification
Protocol	OpenTelemetry (OTLP) — vendor-neutral, future-proof
Trace Propagation	W3C Trace Context headers (`traceparent` , `tracestate`) across all HTTP calls
Span Naming	`{service}.{operation}` (e.g., `api.analyse_document` , `detector.calculate_perplexity` , `db.insert_result`)
Mandatory Span Attributes	`user_id` , `he_level` , `document_word_count` , `http.method` , `http.status_code`
Sampling Strategy	100% in dev/staging, 10% in production (with head-based sampling), 100% for error traces
Retention	7 days in Jaeger (hot), 30 days archived
PII in Traces	Never attach submission text or user PII to span attributes. Use anonymised IDs only

Synthetic Monitoring & Health Checks

Check	Type	Frequency	Target	Alert If
Homepage availability	HTTP GET	60 seconds	`https://{domain}/` — expect 200	Non-200 for 2 consecutive checks
API health endpoint	HTTP GET	30 seconds	`https://{domain}/api/health` — expect `{"status": "healthy"}`	Non-healthy for 2 consecutive checks
Analysis endpoint (canary)	HTTP POST	5 minutes	Submit a 500-word test document, expect result within 10s	Timeout or error response
Login page availability	HTTP GET	60 seconds	`https://{domain}/login` — expect 200	Non-200 for 2 consecutive checks
Database connectivity	Internal	15 seconds	Application health check queries DB	Connection failure for 3 consecutive checks
Redis connectivity	Internal	15 seconds	Application health check pings Redis	Connection failure for 3 consecutive checks
SSL certificate validity	External (Uptime Robot)	Daily	Check certificate expiry date	< 14 days to expiry
DNS resolution	External	5 minutes	Resolve production domain	Resolution failure

On-Call & Operational Readiness

Action	Owner	Completion Criteria
Create on-call rotation schedule (primary + secondary) for Sprints 11-12 and post-launch	Project Manager	Rotation published in PagerDuty, team notified
Define on-call responsibilities and expectations document	Tech Lead	Document signed by all rotation members
Establish on-call compensation / time-off-in-lieu policy	Project Manager / HR	Policy approved
Create `#incidents` and `#alerts` Slack channels with appropriate membership	DevOps Lead	Channels created, integrations configured
Configure PagerDuty escalation policies matching the alerting matrix	DevOps Lead	Policies tested with dry-run alerts
Procure on-call equipment (laptops with VPN access for out-of-hours response)	Project Manager	Equipment distributed to all on-call staff

Service Level Objectives (SLOs)

SLO	SLI (how measured)	Target	Error Budget (per 30 days)	Budget Burn Rate Alert
Availability	Percentage of successful HTTP responses (non-5xx)	99.5%	3.6 hours of downtime	Alert if >50% of monthly budget consumed in 1 hour
Latency (page load)	p95 response time for page requests	< 2 seconds	5% of requests may exceed	Alert if p95 > 2s for 15 consecutive minutes
Latency (analysis)	p95 time to return analysis results for documents < 5000 words	< 5 seconds	5% of requests may exceed	Alert if p95 > 5s for 15 consecutive minutes
Correctness	Percentage of analyses returning valid, non-error results	99.9%	43 seconds of incorrect results	Alert if error rate > 0.5% for 10 minutes
Data Durability	Zero permanent data loss events	100%	0 events	Any data loss event is P1

Error Budget Policy

Budget Status	Action
> 50% remaining	Normal development velocity, feature work prioritised
25-50% remaining	Reliability work prioritised over new features. Review recent changes for contributing factors
< 25% remaining	Feature freeze. All engineering effort directed at reliability improvements
Exhausted (0%)	Full incident response. No deploys except reliability fixes. Post-mortem required for each SLO breach

Key Performance Indicators

Metric	Target	Alert Threshold	Escalation
System Uptime	99.5%	< 99%	Tier 3 — immediate page
Response Time	< 2s	> 5s	Tier 2 — investigate within 2 hours
Analysis Time	< 5s	> 10s	Tier 2 — investigate within 2 hours

Metric	Target	Alert Threshold	Escalation
Error Rate	< 0.5%	> 2%	Tier 3 — immediate page
CPU Usage	< 70%	> 85%	Auto-scale triggered + Tier 1 notification
Memory Usage	< 75%	> 90%	Auto-scale triggered + Tier 1 notification
Database Queries	< 100ms avg	> 500ms	Tier 2 — investigate within 2 hours
DB Replication Lag	< 100ms	> 1 second	Tier 2 — investigate within 2 hours
Disk Usage	< 70%	> 85%	Tier 1 — plan capacity expansion
Redis Hit Rate	> 80%	< 60%	Tier 1 — review cache strategy
Queue Depth (batch)	< 50 jobs	> 200 jobs	Tier 2 — scale workers or investigate backlog
SSL Certificate Expiry	> 30 days	< 14 days	Tier 1 — renew immediately
Failed Login Attempts	< 50/hour	> 200/hour	Tier 3 — potential brute force, investigate immediately
Deployment Frequency	>= 1/sprint	< 1/month	Review in retrospective — delivery pipeline may be blocked
Mean Time to Recovery (MTTR)	< 1 hour (P1)	> 4 hours	Post-mortem required, process improvement action
Cloud Spend (monthly)	Within budget (+/- 10%)	> 120% of forecast	Tier 1 — review resource usage, right-size

---

Monitoring Dashboard Structure

Dashboard	Audience	Key Panels	Refresh Rate
System Overview	DevOps, Tech Lead	Uptime, request rate, error rate, response time (p50/p95/p99), active users, SLO burn rate	30 seconds
Application Performance	Developers	Endpoint-level latency, slowest queries, analysis engine throughput, queue depth, Sentry error groups	30 seconds
Infrastructure	DevOps	CPU/memory/disk per instance, auto-scaling events, pod restarts, node health, network I/O	15 seconds
Database	DevOps, DBA	Connection pool usage, replication lag, query duration, table sizes, backup status, vacuum progress	1 minute

Dashboard	Audience	Key Panels	Refresh Rate
Cache & Queue	DevOps, Developers	Redis hit/miss rate, memory usage, evictions, batch queue depth, job processing rate, dead-letter queue	30 seconds
Security	Security Lead	Failed logins, rate limit breaches, RBAC violations, certificate status, WAF events, suspicious IP activity	1 minute
Business Metrics	Product Owner	Analyses per day, unique users, batch vs single ratio, most-used HE levels, export count, user growth trend	5 minutes
SLO Status	Tech Lead, Product Owner	SLO compliance per objective, error budget remaining, budget burn rate, 7/30-day trends	5 minutes
Cost & Capacity	DevOps, Project Manager	Monthly cloud spend vs budget, cost per analysis, resource utilisation trends, capacity forecasts	Daily
Distributed Traces	Developers	Slowest traces, error traces, service dependency map, trace duration distribution	On-demand

---

Alerting & Escalation Matrix

Severity	Examples	Notification Channel	Response Time	Escalation If Unacknowledged
P1 — Critical	System down, data breach, >5% error rate, SLO budget exhausted	PagerDuty (phone call) + Slack `#incidents`	15 minutes	Auto-escalate to Tech Lead after 15 min, CTO after 30 min
P2 — High	Degraded performance (>5s response), single AZ outage, auth service down, >50% SLO budget burned in 1 hour	PagerDuty (push) + Slack `#incidents`	30 minutes	Auto-escalate to Tech Lead after 1 hour
P3 — Medium	Elevated error rate (>1%), high CPU/ memory, slow queries, replication lag >1s	Slack `#alerts` + email	2 hours	Review in next standup if unresolved

Severity	Examples	Notification Channel	Response Time	Escalation If Unacknowledged
P4 — Low	Disk usage warning, certificate renewal due, non-critical dependency degraded, cost overrun warning	Slack `#alerts`	Next business day	Add to sprint backlog

Alert Hygiene Standards

Standard	Requirement
Signal-to-noise ratio	Every alert must be actionable. If an alert fires and requires no action, it must be tuned or removed
Alert fatigue review	Monthly review of all alerts: silence rate, false positive rate, mean time to acknowledge. Remove or tune alerts with >20% false positive rate
Alert documentation	Every alert rule must link to a runbook with investigation steps
Deduplication	Alerts for the same root cause must be grouped (PagerDuty alert grouping enabled)
Maintenance suppression	Alerts automatically suppressed during scheduled maintenance windows
Test alerts	Quarterly dry-run of all P1 and P2 alert paths to verify delivery and escalation

---

Logging Standards

Standard	Specification
Format	Structured JSON: `{"timestamp", "level", "service", "trace_id", "span_id", "user_id", "message", "metadata"}`
Levels	`DEBUG` (dev only), `INFO`, `WARN`, `ERROR`, `FATAL`
Retention	30 days hot (searchable in Kibana), 90 days warm (compressed archive, queryable), 1 year cold (compliance, S3 Glacier)
PII Handling	Never log raw submission text, passwords, or tokens. Mask user emails in logs (`j***@example.com`). Hash user IDs in non-audit logs
Correlation	Unique `trace_id` per request (from OpenTelemetry), propagated across all services and log entries
Audit Trail	Separate audit log index for: login/logout, role changes, data exports, data deletion, admin actions — retained for 2 years (GDPR compliance)
Log Volume Budget	Target < 5GB/day in production. Alert if >10GB/day sustained — investigate verbose logging or unexpected traffic

Standard	Specification
Sensitive Data Scanner	Automated regex scanner in Logstash/Fluentd pipeline to detect and redact accidental PII leaks before indexing

Mandatory Log Events

The following events **must** be logged at the specified level. This is enforced via code review and CI linting.

Event	Log Level	Required Fields	Purpose
HTTP request received	`INFO`	trace_id, method, path, user_id, ip_address	Request tracking
HTTP response sent	`INFO`	trace_id, method, path, status_code, duration_ms	Latency monitoring
Analysis started	`INFO`	trace_id, user_id, he_level, document_word_count, source_type	Throughput tracking
Analysis completed	`INFO`	trace_id, user_id, he_level, duration_ms, confidence_score	Performance + business metrics
Analysis failed	`ERROR`	trace_id, user_id, error_type, error_message, stack_trace	Error tracking
Login attempt	`INFO`	user_id (or email_hash), result (success/failure), ip_address, mfa_used	Security monitoring
Login failure (threshold exceeded)	`WARN`	email_hash, ip_address, failure_count	Brute force detection
Role change	`INFO` (audit)	admin_user_id, target_user_id, old_role, new_role	Audit trail
Data export	`INFO` (audit)	user_id, export_type, record_count	Audit trail
Data deletion	`INFO` (audit)	admin_user_id, target_user_id, records_deleted	GDPR compliance
Batch job queued	`INFO`	trace_id, user_id, file_count, batch_id	Queue monitoring
Batch job completed	`INFO`	trace_id, batch_id, duration_ms, files_processed, files_failed	Throughput monitoring
Health check failure	`ERROR`	component (db/redis/queue), error_message	Dependency monitoring
Auto-scale event	`WARN`	direction (up/down), trigger_metric, old_count, new_count	Capacity monitoring
Deployment event	`INFO`	version, deployer, environment, strategy	Change tracking

Incident Response Procedures

Phase	Actions	Owner	SLA
1. Detection	Automated alert fires or user report received	Monitoring system / Support	Immediate
2. Triage	On-call engineer acknowledges alert, assesses severity using the matrix above	On-call engineer	15 min (P1), 30 min (P2)
3. Containment	Isolate affected component (e.g., rollback deploy, failover DB, block IP)	On-call engineer	30 min (P1), 2 hours (P2)
4. Communication	Update `#incidents` Slack channel, notify stakeholders if user-facing impact	On-call engineer / Project Manager	Concurrent with containment
5. Resolution	Root cause identified and fix deployed, or permanent workaround applied	Engineering team	4 hours (P1), 24 hours (P2)
6. Post-Mortem	Blameless post-mortem document: timeline, root cause, impact, lessons, action items	Tech Lead	Within 48 hours of resolution
7. Follow-Up	Action items from post-mortem tracked as sprint tasks, process improvements implemented	Project Manager	Within 1 sprint

Runbook Template

Every alert must have an associated runbook. Runbooks follow this standard template and are stored in the ops repository.

Section	Content
Title	Alert name and severity
Description	What this alert means and why it matters
Impact	What users/systems are affected if this is not resolved
Likely Causes	Numbered list of most common root causes (ordered by likelihood)
Investigation Steps	Step-by-step commands and dashboard links to diagnose the issue
Resolution Steps	For each likely cause: specific commands or actions to resolve
Rollback Procedure	How to revert to last known good state if resolution fails
Escalation	When and to whom to escalate if resolution is not achieved within SLA
Post-Resolution	Verification steps to confirm the issue is fully resolved

Section	Content
History	Log of previous occurrences, root causes, and resolutions

Required Runbooks (Sprint 0)

Runbook	Trigger Alert	Priority
Application server unresponsive	P1: System down	Must-have
Database failover	P1: DB primary unreachable	Must-have
High error rate (>2%)	P1: Error rate threshold	Must-have
Deployment rollback	P1/P2: Post-deploy anomaly	Must-have
Redis cluster failure	P2: Cache unavailable	Must-have
SSL certificate expiry	P1: Certificate expired / P4: Renewal warning	Must-have
High CPU / memory	P3: Resource threshold	Must-have
Disk space critical	P3: Disk usage >85%	Must-have
Brute force login detected	P3: Failed login threshold	Must-have
Batch queue backlog	P2: Queue depth >200	Must-have
Database replication lag	P3: Lag >1 second	Should-have
Cloud spend overrun	P4: Budget >120%	Should-have

Post-Mortem Template

Section	Content
Incident ID	Unique identifier (e.g., INC-2026-001)
Date & Duration	Start time, detection time, resolution time, total duration
Severity	P1 / P2 / P3 / P4
Impact	Users affected, requests failed, data impacted, SLO budget consumed
Timeline	Minute-by-minute chronology of events, actions taken, and communications
Root Cause	Technical explanation of what caused the incident
Contributing Factors	Process, tooling, or human factors that allowed the root cause to occur
What Went Well	Detection speed, response effectiveness, communication quality

Section	Content
What Could Be Improved	Gaps identified in monitoring, runbooks, processes, or tooling
Action Items	Numbered list with owner, due date, and priority. Each item must be tracked as a sprint task
Lessons Learned	Key takeaways for the team

Support & Maintenance Plan

Tier 1 Support (Response Time: 4 hours)

- User account issues
- Basic troubleshooting
- Documentation requests
- Password resets and access queries

Tier 2 Support (Response Time: 2 hours)

- Performance issues
- Integration problems
- Data export/import issues
- Incorrect analysis results (escalate to ML team if pattern detected)

Tier 3 Support (Response Time: 1 hour)

- System outages
- Security incidents
- Critical bugs affecting core functionality
- Data loss or corruption

Support Channel Matrix

Channel	Availability	Audience	Response SLA	Escalation Path
In-app help / FAQ	24/7 (self-service)	All users	Immediate (automated)	Link to email support
Email (`support@{domain}`)	Business hours (Mon-Fri 09:00-17:00 UTC)	All users	4 hours (Tier 1)	Auto-escalate if no response in 8 hours
Slack `#ai-detection-support`	Business hours	Educators with Slack access	2 hours (Tier 1/2)	Tag on-call engineer if unresolved in 4 hours

Channel	Availability	Audience	Response SLA	Escalation Path
PagerDuty (direct page)	24/7	Internal team / automated alerts only	15 minutes (P1)	Per escalation matrix
Quarterly review meeting	Quarterly	Department heads, academic integrity officers	N/A	Strategic issues raised to steering committee

Support Metrics & Targets

Metric	Target	Review Cadence
First response time (Tier 1)	< 4 hours	Weekly
Resolution time (Tier 1)	< 24 hours	Weekly
Resolution time (Tier 2)	< 8 hours	Weekly
Resolution time (Tier 3)	< 4 hours	Per incident
Ticket backlog (open > 48 hours)	< 5 tickets	Daily standup
User satisfaction (post-resolution survey)	>= 4.0 / 5	Monthly
Repeat contact rate (same issue)	< 10%	Monthly
Self-service resolution rate (FAQ/docs)	>= 40%	Monthly

Maintenance Windows & Patching

Maintenance Type	Frequency	Window	Approval Required	Notification	Rollback Plan
OS security patches	Weekly	Sunday 02:00-04:00 UTC	Auto-approved if non-breaking	Slack `#ops`	Revert to pre-patch AMI/ snapshot
Database minor upgrades	Monthly	Sunday 02:00-06:00 UTC	DevOps Lead	48-hour advance notice	Restore from pre-upgrade snapshot
Application dependency updates	Fortnightly (sprint cadence)	During sprint deploy	PR review	Part of release notes	Standard deployment rollback
Kubernetes version upgrades	Quarterly	Scheduled maintenance window	Tech Lead + DevOps Lead	2-week advance notice	Roll back node pool version

Maintenance Type	Frequency	Window	Approval Required	Notification	Rollback Plan
TLS certificate renewal	Auto-renewed 30 days before expiry	Automated	None (automated)	Alert if auto-renewal fails	Manual renewal from CA
Database vacuum / reindex	Weekly	Sunday 04:00-05:00 UTC	Auto-approved	None (silent)	N/A (non-destructive)
Log rotation / archive	Daily	03:00 UTC	Auto-approved	None (silent)	Restore from archive if needed
Backup verification (test restore)	Monthly	Saturday 02:00-06:00 UTC	DevOps Lead	Slack `#ops`	N/A (read-only test)

Capacity Planning

Resource	Current Sizing	6-Month Projection	Growth Trigger	Scale Action	Lead Time
App servers	3x instances	5x instances (70%+ dept adoption)	Sustained >70% CPU across fleet	Add instances via auto-scaling (immediate) or increase max pool	Immediate (auto)
Database storage	500GB SSD	800GB (based on 500 analyses/ day)	>70% utilisation	Expand volume (online resize)	1 hour
Database compute	16 vCPU / 32GB	May need 32 vCPU / 64GB	Sustained >70% CPU or connection pool exhaustion	Vertical scale (requires brief restart)	2 hours (scheduled maintenance)
Redis memory	8GB per node	12GB per node	>70% utilisation, hit rate <60%	Add shards or increase node memory	1 hour
Object storage	1TB	2TB	>70% utilisation	Effectively unlimited — review lifecycle policies	N/A

Resource	Current Sizing	6-Month Projection	Growth Trigger	Scale Action	Lead Time
CDN bandwidth	Base tier	Mid tier	Sustained >80% of plan limit	Upgrade tier	24 hours
Log storage	50GB/ month	150GB/ month	>80% of Elasticsearch disk	Expand cluster or increase retention tiering	4 hours

Capacity Review Cadence

Review Type	Frequency	Attendees	Output
Weekly resource check	Weekly (automated)	DevOps Lead (dashboard review)	Slack notification if any resource >60%
Monthly capacity review	Monthly	DevOps Lead, Tech Lead	Capacity report: trends, forecasts, recommendations
Quarterly planning review	Quarterly	DevOps Lead, Tech Lead, Project Manager, Product Owner	Updated 6-month projections, budget adjustment requests

---

Cost Monitoring & Optimisation

Action	Owner	Frequency	Target
Configure cloud provider budget alerts (50%, 80%, 100%, 120% of monthly forecast)	DevOps Lead	Sprint 0 (one-time)	Alerts active before any production spend
Tag all resources for cost allocation (`Project` , `Environment` , `Service` , `CostCentre`)	DevOps Lead	Enforced from Sprint 0	100% resource tagging compliance
Generate weekly cost breakdown by service and environment	DevOps Lead	Weekly (automated)	Report delivered to `#ops` Slack and Project Manager
Review cost-per-analysis metric	DevOps Lead / Product Owner	Monthly	Establish baseline Sprint 11, target <=£0.02/analysis

Action	Owner	Frequency	Target
Right-sizing review (identify over-provisioned resources)	DevOps Lead	Monthly	Reduce waste — target <20% idle capacity
Reserved instance / savings plan evaluation	DevOps Lead / Project Manager	After 3 months of production data	Target 30-40% savings vs on-demand pricing
Dev/staging environment shutdown outside business hours	DevOps Lead	Sprint 3 (automated)	Staging: weekdays 08:00-20:00 UTC only. Dev: on-demand

Compliance & Audit Monitoring

Requirement	Monitoring Approach	Alert Condition	Owner
GDPR data retention	Automated job checks for records past retention period (90 days for submissions, 2 years for audit logs)	Any records found past retention deadline	Lead Dev / DPO
DSAR (Data Subject Access Request)	Dashboard tracking open DSARs with SLA countdown (30-day statutory deadline)	DSAR approaching 25 days without completion	DPO
Right to erasure	Audit log confirming deletion of all user data when requested	Deletion request open >7 days	Lead Dev / DPO
Access control compliance	Quarterly RBAC review: verify all user roles match current job functions	Users with roles not matching HR records	Security Lead
Encryption compliance	Automated check that all data stores have encryption enabled (at rest and in transit)	Any unencrypted data store detected	DevOps Lead
Backup compliance	Automated verification that backups completed successfully and test-restore passed	Backup failure or test-restore failure	DevOps Lead
Penetration test follow-up	Track remediation of all pen test findings	Any high/critical finding open >14 days	Security Lead
Dependency vulnerabilities	Automated scan (Dependabot/ Snyk) results tracked on dashboard	Any critical CVE unpatched >7 days	Lead Dev

Monitoring & Maintenance Cadence Per Sprint

Monitoring and operational maturity is built progressively. Each sprint has defined activities.

Sprint	Monitoring & Maintenance Activities
Sprint 0	Provision Prometheus + Grafana + ELK in dev. Configure PagerDuty. Define SLIs/SLOs. Write Sprint 0 runbooks. Set cloud budget alerts. Define logging standards. Establish on-call schedule
Sprint 1-2	Instrument application metrics (`/metrics` endpoint). Integrate Sentry for error tracking. Integrate OpenTelemetry tracing. Enforce structured JSON logging. Create System Overview dashboard
Sprint 3-4	Provision staging monitoring stack. Create Application Performance + Infrastructure dashboards. Configure synthetic health checks (staging). Implement cost tagging. Set up dev/staging auto-shutdown
Sprint 5-6	Create Security dashboard + audit log index. Configure GDPR compliance monitoring jobs. Create Cache & Queue dashboard. Write security runbooks. First alert hygiene review
Sprint 7-8	Create Business Metrics + SLO Status dashboards. Full alert rule audit against runbooks. Load-test staging and tune alert thresholds based on real data. Dry-run P1 alert path
Sprint 9-10	Provision production monitoring stack (replicate from staging IaC). Activate synthetic monitoring on production domain. Create Cost & Capacity dashboard. Execute DR drill and validate runbooks. Write remaining runbooks
Sprint 11-12	Activate all production alerting. On-call rotation goes live. Monitor SLO burn rates during pilot. Conduct first weekly capacity check. First post-incident review (if applicable). Baseline all operational metrics

Operational Maturity Targets

Maturity Area	Sprint 0-2 (Foundation)	Sprint 3-6 (Build-out)	Sprint 7-10 (Hardening)	Sprint 11-12+ (Production)
Metrics	App metrics instrumented	All dashboards live in staging	Alert thresholds tuned from load tests	All dashboards live in production, SLO tracking active
Logging	Structured JSON enforced	Audit logging, PII scanner active	Log volume budgets enforced	Full compliance monitoring, 2-year audit retention
Tracing	SDK integrated, traces in dev	Traces in staging, service map visible	Sampling tuned for production	Production tracing active, used for incident investigation
Alerting	Basic health check alerts	Full alert rules in staging	All alerts linked to runbooks, dry-runs complete	Production alerting live, monthly hygiene reviews

Maturity Area	Sprint 0-2 (Foundation)	Sprint 3-6 (Build-out)	Sprint 7-10 (Hardening)	Sprint 11-12+ (Production)
Incident Response	IRP documented, runbook templates	Core runbooks written	Tabletop exercise completed, escalation tested	On-call active, post-mortems after every P1/P2
Cost Management	Budget alerts set	Tagging enforced, weekly reports	Right-sizing review, dev/staging shutdown	Baseline cost-per-analysis, savings plan evaluation

---

Monitoring & Maintenance Exit Criteria (Sprint 0)

Before Sprint 1, the following must be confirmed:

- [ ] Prometheus + Grafana provisioned and accessible in dev environment
- [ ] ELK stack (or managed equivalent) provisioned and indexing logs from dev
- [ ] PagerDuty account provisioned, on-call schedules created, test page verified
- [ ] SLIs and SLOs documented, error budget policy approved
- [ ] Structured JSON logging standard documented and shared with all developers
- [ ] OpenTelemetry tracing SDK selected and sample integration tested
- [ ] Sentry account provisioned (or equivalent error tracking tool)
- [ ] Synthetic monitoring tool selected and account provisioned (activated at launch)
- [ ] All Sprint 0 runbooks written (12 must-have runbooks per the required runbooks table)
- [ ] Post-mortem template committed to ops repository
- [ ] Cloud budget alerts configured (50%, 80%, 100%, 120% thresholds)
- [ ] Cost allocation tagging standard defined and documented
- [ ] `#incidents` and `#alerts` Slack channels created with integrations
- [ ] On-call responsibilities document signed by all rotation members
- [ ] GDPR compliance monitoring requirements documented and assigned to sprints

---

Risk Management

Risk Management Prerequisites (Sprint 0)

Risk management is a **continuous discipline**, not a one-time planning exercise. All risk governance structures, assessment frameworks, and monitoring processes must be established **before Sprint 1 begins**.

Risk Governance Structure

Role	Responsibility	Person / Forum
Risk Owner	Accountable for a specific risk — ensures mitigation actions are executed and status is reported	Assigned per risk (see risk register below)
Risk Manager	Maintains the risk register, facilitates risk reviews, escalates to steering committee	Project Manager
Change Control Board (CCB)	Approves or rejects change requests that could introduce new risks or affect scope	Project Manager (chair), Tech Lead, Product Owner, Security Lead
Steering Committee	Strategic oversight, approves risk appetite changes, makes go/no-go decisions at major gates	Head of Department, CTO, DPO, Product Owner
All Team Members	Identify and raise new risks at any time via the risk register or standup	Full team

Risk Assessment Framework

All risks are scored using a **Likelihood x Impact** matrix. This provides a consistent, objective basis for prioritisation.

Likelihood Scale

Score	Likelihood	Definition
1	Rare	< 10% probability. Has never occurred in similar projects
2	Unlikely	10-25% probability. Could occur but not expected
3	Possible	25-50% probability. Has occurred in similar projects
4	Likely	50-75% probability. Expected to occur at least once
5	Almost Certain	> 75% probability. Will almost certainly occur

Impact Scale

Score	Impact	Schedule	Budget	Quality	Reputation
1	Negligible	< 1 day delay	< 1% overrun	Cosmetic defect	No external awareness
2	Minor	1-3 day delay	1-5% overrun	Minor feature degraded	Internal awareness only
3	Moderate	1-2 week delay	5-15% overrun	Key feature impaired	Limited external awareness

Score	Impact	Schedule	Budget	Quality	Reputation
4	Major	2-4 week delay	15-30% overrun	Core feature broken, workaround needed	Reported externally, stakeholder concern
5	Critical	> 4 week delay / project at risk	> 30% overrun	System unusable, data loss, legal exposure	Reputational damage, regulatory action

Risk Rating Matrix

	Impact 1	Impact 2	Impact 3	Impact 4	Impact 5
Likelihood 5	Medium (5)	High (10)	High (15)	Critical (20)	Critical (25)
Likelihood 4	Low (4)	Medium (8)	High (12)	High (16)	Critical (20)
Likelihood 3	Low (3)	Medium (6)	Medium (9)	High (12)	High (15)
Likelihood 2	Low (2)	Low (4)	Medium (6)	Medium (8)	High (10)
Likelihood 1	Low (1)	Low (2)	Low (3)	Low (4)	Medium (5)

Risk Response by Rating

Rating	Response	Review Frequency	Escalation
Critical (15-25)	Immediate mitigation required. Sprint work paused if necessary. Steering committee notified	Weekly (or more frequently if active)	Steering committee
High (10-14)	Active mitigation plan with named owner and sprint-level actions	Fortnightly (at sprint review)	CCB
Medium (5-9)	Mitigation plan in place, monitored regularly	Monthly	Project Manager
Low (1-4)	Accept and monitor. Revisit if conditions change	Quarterly	N/A

Risk Register

> **All risks have pre-sprint mitigation actions defined in Sprint 0.** Sprint 0 must be completed before development begins. The register below provides the full risk profile including scoring, indicators, and contingency plans.

RISK-001: Scope Creep

Attribute	Detail
Risk ID	RISK-001
Category	Project Management
Description	Uncontrolled expansion of project scope through ad-hoc feature requests, stakeholder misalignment, or unclear requirements, leading to schedule overrun and diluted quality
Likelihood	4 (Likely)
Impact	4 (Major)
Rating	**16 — High**
Risk Owner	Project Manager

Phase	Mitigation Actions
Sprint 0	PRD signed off by all stakeholders. CCB established with change request process. Sprint 1-2 backlog frozen. Definition of Done agreed. Fortnightly stakeholder sessions scheduled
Ongoing	CCB reviews all change requests — must demonstrate business value and impact assessment before approval. DoD enforced per sprint. Backlog grooming every sprint. Velocity tracked to detect overcommitment early

Early Warning Indicators	Detection Method	Trigger Action
Change requests exceeding 2 per sprint	CCB log review	CCB escalates to steering committee. Stakeholder realignment session
Sprint velocity declining >20% over 2 sprints	Sprint burndown chart	Scrum Master raises in retrospective. Backlog re-prioritised
Sprint goals not met for 2 consecutive sprints	Sprint review outcomes	Project Manager escalates to steering committee. Scope reduction discussion
Backlog growing faster than items are completed	Backlog size trend	CCB freezes all non-critical additions until backlog stabilises

Contingency Plan (if risk materialises)
1. Invoke CCB emergency session to halt all new change requests
2. Re-baseline scope against original PRD — identify and defer non-essential features to Phase 2
3. Extend timeline by 1-2 sprints (with steering committee approval) rather than cutting quality
4. Communicate revised timeline to all stakeholders within 48 hours

RISK-002: Data Privacy Breach

Attribute	Detail
Risk ID	RISK-002
Category	Security & Compliance
Description	Unauthorised access to, loss of, or exposure of personal data (student submissions, educator details), resulting in GDPR violation, ICO enforcement action, and reputational damage
Likelihood	2 (Unlikely — with mitigations)
Impact	5 (Critical)
Rating	**10 — High**
Risk Owner	Security Lead / DPO

Phase	Mitigation Actions
Sprint 0	DPIA completed and submitted. Incident Response Plan documented and tabletop-tested. Threat model (STRIDE) completed. Encryption standards set (TLS 1.3 / AES-256). Data classification defined. DPA signed with cloud provider
Ongoing	Security audits (Sprint 6, 10). Automated data retention enforcement (90-day submission deletion). Dependency scanning on every PR. Pen test before launch (Sprint 10). Annual DPIA review. GDPR compliance monitoring dashboards active

Early Warning Indicators	Detection Method	Trigger Action
Unencrypted data store detected	Automated compliance monitoring	P1 alert — encrypt immediately, investigate how it was missed
PII found in application logs	Log PII scanner (Logstash/Fluentd)	P2 alert — redact, fix logging code, code review
Failed pen test findings (high/critical)	Pen test report (Sprint 10)	Sprint work paused until all high/ critical findings resolved
DSAR response approaching 25 days	DSAR tracking dashboard	DPO escalates to Lead Dev for immediate completion
Unusual data access patterns	Security dashboard (audit logs)	Security Lead investigates within 2 hours

Contingency Plan (if breach occurs)
1. Activate Incident Response Plan — containment within 30 minutes
2. DPO notified within 1 hour. Forensic investigation begins
3. If personal data confirmed compromised: notify ICO within 72 hours (GDPR Article 33)
4. Affected data subjects notified without undue delay (GDPR Article 34)

Contingency Plan (if breach occurs)
5. System isolated until root cause identified and patched
6. Post-mortem within 48 hours. Corrective actions tracked as P1 sprint tasks
7. Legal counsel engaged if regulatory action anticipated

RISK-003: False Positives / Detection Accuracy

Attribute	Detail
Risk ID	RISK-003
Category	Product Quality
Description	Detection engine incorrectly flags human-written work as AI-generated (false positive) or fails to detect AI-generated content (false negative), undermining educator trust and potentially harming students
Likelihood	4 (Likely)
Impact	4 (Major)
Rating	**16 — High**
Risk Owner	ML Lead

Phase	Mitigation Actions
Sprint 0	Validation dataset curated (650+ samples across HE levels 4-8, AI-generated, human, mixed). Accuracy/precision/recall/F1 targets defined per HE level. Known limitations documented. Confidence interval display strategy approved
Ongoing	Educator feedback loop captures flagged incorrect results. Quarterly dataset refresh with new AI model outputs. Continuous algorithm refinement based on feedback. Confidence intervals displayed in UI — tool never presents results as absolute. Clear disclaimer: "This tool supports educator judgement; it does not replace it"

Early Warning Indicators	Detection Method	Trigger Action
Accuracy below 85% on validation set	Automated regression test on model changes	Block deployment until accuracy restored
False positive rate >15%	Educator feedback dashboard	ML Lead prioritises algorithm refinement in next sprint
Spike in "incorrect result" feedback submissions	Business metrics dashboard	ML Lead investigates within 1 sprint. Pattern analysis on affected documents

Early Warning Indicators	Detection Method	Trigger Action
New AI model release (e.g., GPT-5, Claude 4)	AI industry monitoring	Immediate validation dataset expansion and re-benchmark within 2 weeks
Performance divergence across HE levels	Per-level accuracy dashboard	Targeted refinement for underperforming levels

Contingency Plan (if accuracy drops significantly)
1. Increase confidence threshold — tool reports "uncertain" rather than "AI-generated" for borderline cases
2. Add prominent UI warning: "Detection accuracy is currently under review. Please exercise additional judgement"
3. ML Lead allocates full sprint to algorithm investigation and retraining
4. Stakeholder communication: transparent update to all departments on accuracy status and remediation plan
5. If accuracy cannot be restored to 85% within 2 sprints: steering committee review on whether to pause production use

---

RISK-004: Performance Degradation

Attribute	Detail
Risk ID	RISK-004
Category	Technical
Description	System fails to meet performance targets (page load <2s, analysis <5s, 500+ concurrent users) under real-world load, resulting in poor user experience and low adoption
Likelihood	3 (Possible)
Impact	3 (Moderate)
Rating	**9 — Medium**
Risk Owner	Tech Lead

Phase	Mitigation Actions
Sprint 0	Performance budgets defined. Monitoring provisioned in dev from day one. Scalable architecture chosen (K8s auto-scaling). Load testing cadence agreed (every sprint from Sprint 4)
Ongoing	Load tests every sprint against staging. Auto-scaling configured with defined triggers. CDN for static assets. Database query optimisation tracked. Redis caching for repeated analyses. Performance gate in CI/CD pipeline

Early Warning Indicators	Detection Method	Trigger Action
p95 response time >2s in staging	Performance dashboard	Investigate and optimise before sprint close
Analysis time >5s for documents <5000 words	Load test results	Profile analysis engine, identify bottleneck
Database query avg >100ms	Database dashboard	DBA reviews query plans, adds indexes
Auto-scaling events >3 per day during normal load	Infrastructure dashboard	Review resource sizing — may need vertical scale
Batch processing queue growing continuously	Cache & Queue dashboard	Scale batch workers or investigate stuck jobs

Contingency Plan (if performance targets not met at launch)
1. Enable aggressive caching (increase Redis TTL, add page-level caching)
2. Implement request queuing with user-visible progress indicator for analysis
3. Reduce concurrent user target for soft launch (250 instead of 500) and scale progressively
4. Allocate dedicated Sprint 13 (extension) for performance-only optimisation
5. Consider horizontal scaling: add analysis worker nodes as a separate microservice

---

RISK-005: Integration Delays

Attribute	Detail
Risk ID	RISK-005
Category	Technical
Description	Delays integrating with third-party systems (institutional SSO, LMS platforms, email services) or internal module integration failures, blocking dependent features
Likelihood	3 (Possible)
Impact	3 (Moderate)
Rating	**9 — Medium**
Risk Owner	Lead Dev

Phase	Mitigation Actions
Sprint 0	All API contracts defined (OpenAPI specs) and committed before coding. Contract testing (Pact) in CI pipeline. Third-party integration inventory complete with confirmed access credentials. Contact details for all external integration teams documented

Phase	Mitigation Actions
Ongoing	Contract tests on every PR. Dedicated integration sprint (Week 7). Early smoke tests against LMS/SSO in staging from Sprint 4. Mock services for all external dependencies to unblock development

Early Warning Indicators	Detection Method	Trigger Action
Contract test failures on PR	CI pipeline	Block merge until contracts aligned
Third-party API credentials not received by Sprint 2	Integration inventory tracker	Project Manager escalates to institutional IT
SSO integration not testable by Sprint 4	Sprint review	Build standalone auth as fallback; SSO becomes Phase 2
LMS integration spec changes mid-project	External team communication	CCB evaluates impact; re-scope if necessary

Contingency Plan (if integration is blocked)
1. Deploy with standalone authentication (email/password + MFA) — SSO added post-launch as Phase 2
2. Provide CSV/JSON export as interim alternative to LMS integration
3. Use mock/stub services to unblock all dependent features in development
4. Negotiate extended timeline for integration-specific features with steering committee

---

RISK-006: Resource Constraints

Attribute	Detail
Risk ID	RISK-006
Category	People & Process
Description	Key team members become unavailable (illness, departure, competing priorities), or specialist skills (ML, security, DevOps) are insufficient, delaying delivery
Likelihood	3 (Possible)
Impact	3 (Moderate)
Rating	**9 — Medium**
Risk Owner	Project Manager

Phase	Mitigation Actions
Sprint 0	Team allocation confirmed for full 12 weeks. Bus-factor risks identified and cross-training pairs established. External consulting budget secured (security audit, accessibility audit). All dev environments provisioned for all team members
Ongoing	Weekly capacity checks in standup. Pair programming on critical-path work. Knowledge sharing sessions every sprint. External consultants on standby for specialist gaps. Buffer time built into Sprints 7-8

Early Warning Indicators	Detection Method	Trigger Action
Team member unavailable for >3 days unplanned	Standup / absence tracker	Cross-training pair takes over. Backlog re-prioritised
Sprint velocity drops >20%	Burndown chart	Scrum Master investigates in retrospective. Resource rebalancing
Specialist work (ML, security) blocked for >1 week	Sprint board	Engage external consultant within 48 hours
Team morale declining	Anonymous sprint retrospective feedback	Project Manager addresses in 1:1s. Workload review

Contingency Plan (if key person leaves)
1. Cross-training pair assumes ownership immediately — no single point of failure
2. External contractor engaged within 1 week for specialist roles (ML, DevOps, Security)
3. Scope reduced to critical path only if resource gap cannot be filled within 2 weeks
4. Knowledge base and documentation (maintained throughout) enables faster onboarding

---

RISK-007: AI Landscape Rapid Evolution

Attribute	Detail
Risk ID	RISK-007
Category	External / Strategic
Description	Rapid evolution of AI models (new capabilities, new evasion techniques, new detection methods) renders the detection engine less effective or obsolete mid-project or shortly after launch
Likelihood	4 (Likely)
Impact	3 (Moderate)
Rating	**12 — High**
Risk Owner	ML Lead / Product Owner

Phase	Mitigation Actions
Sprint 0	Modular detection architecture designed — engine can be updated independently of web application. Quarterly dataset refresh planned. Known limitations documented up front
Ongoing	ML Lead monitors major AI model releases (GPT, Claude, Gemini, Llama, Mistral). Validation dataset expanded within 2 weeks of any major release. Detection engine designed as pluggable module for algorithm swaps. Roadmap includes post-launch iteration budget

Early Warning Indicators	Detection Method	Trigger Action
New major AI model released	AI industry news monitoring (ML Lead)	Re-benchmark validation dataset within 2 weeks
New AI paraphrasing/evasion tool gains popularity	Academic integrity community forums, educator feedback	Test against tool outputs, update detection rules
Competitor tools announce significantly improved detection	Competitor monitoring	Review competitor approach, assess if technique is applicable
Accuracy drop on newly submitted documents vs validation set	Business metrics dashboard (live accuracy vs baseline)	ML Lead investigates, dataset refresh triggered

Contingency Plan (if detection approach becomes ineffective)
1. Increase reliance on structural/critical analysis scoring (less susceptible to model changes) over phrase detection
2. Engage academic research partners for latest detection techniques
3. Consider integrating third-party detection APIs as supplementary signals
4. Communicate honestly to stakeholders: "detection is an evolving challenge" — set expectations for continuous improvement
5. Pivot messaging from "AI detection" to "writing quality and critical analysis assessment" if detection accuracy proves fundamentally unreliable

---

RISK-008: Regulatory & Policy Changes

Attribute	Detail
Risk ID	RISK-008
Category	External / Compliance
Description	Changes to UK GDPR, Data Protection Act 2018, EU AI Act, or institutional AI policies mid-project that impose new requirements on the tool (e.g., mandatory transparency, right to explanation, prohibition of automated decision-making)

Attribute	Detail
Likelihood	2 (Unlikely within 12 weeks, but possible post-launch)
Impact	4 (Major)
Rating	**8 — Medium**
Risk Owner	DPO / Product Owner

Phase	Mitigation Actions
Sprint 0	Current regulatory landscape reviewed and documented. Tool positioned as decision-support (not automated decision-making) to reduce AI Act exposure. DPO confirms current compliance posture
Ongoing	DPO monitors UK GDPR / AI Act developments quarterly. Architecture designed for explainability (confidence scores, factor breakdowns). No fully automated decisions — educator always makes final judgement. Privacy notice updateable without code changes

Early Warning Indicators	Detection Method	Trigger Action
EU AI Act enforcement timeline update	DPO regulatory monitoring	DPO assesses impact, reports to steering committee
Institutional AI policy revision	Academic governance communications	Product Owner reviews alignment, raises CCB change request if needed
ICO guidance update on AI in education	ICO newsletter / DPO monitoring	DPO circulates to team with impact assessment

Contingency Plan (if new regulation requires changes)
1. DPO produces impact assessment within 2 weeks of regulation publication
2. CCB evaluates scope of required changes and re-prioritises backlog
3. If changes are minor: absorb into existing sprint cadence
4. If changes are major: dedicated compliance sprint with steering committee approval
5. Legal counsel engaged if compliance uncertainty exists

RISK-009: Low User Adoption

Attribute	Detail
Risk ID	RISK-009
Category	Business / Stakeholder

Attribute	Detail
Description	Educators do not adopt the tool due to distrust of AI detection, perceived complexity, lack of training, or philosophical objections to detection tools, resulting in failure to meet the 70% department adoption target
Likelihood	3 (Possible)
Impact	4 (Major)
Rating	**12 — High**
Risk Owner	Product Owner

Phase	Mitigation Actions
Sprint 0	User research with 5+ educators across HE levels. Personas validated with stakeholders. Competitor analysis completed. Educator champions identified in each target department
Ongoing	Usability testing every other sprint (Sprints 4, 8). Training sessions at launch (Sprint 11). In-app onboarding flow. FAQ and video tutorials. Quarterly review meetings with department heads. Educator feedback loop for continuous improvement

Early Warning Indicators	Detection Method	Trigger Action
SUS score <72 in usability testing	Usability test results (Sprint 4, 8)	UX Lead prioritises redesign of problem areas
Negative sentiment in UAT feedback	UAT session notes	Product Owner addresses concerns directly with educators
Pilot user active usage <50% after 2 weeks	Business metrics dashboard	Targeted outreach and 1:1 training for inactive users
Educator champions expressing reservations	Stakeholder meetings	Product Owner hosts concerns workshop, adjusts messaging

Contingency Plan (if adoption is below target)
1. Conduct additional training workshops (drop-in format, recorded for async viewing)
2. Appoint departmental "super-users" with dedicated support channel
3. Add "quick win" features based on educator feedback to demonstrate value fast
4. Reposition tool messaging: emphasise writing quality assessment alongside AI detection
5. Offer pilot departments extended hands-on support (dedicated Slack channel, weekly office hours)

Risk Register Summary

Risk ID	Risk	Likelihood	Impact	Rating	Owner	Status
RISK-001	Scope Creep	4	4	**16 — High**	Project Manager	Mitigating (Sprint 0)
RISK-002	Data Privacy Breach	2	5	**10 — High**	Security Lead / DPO	Mitigating (Sprint 0)
RISK-003	False Positives / Detection Accuracy	4	4	**16 — High**	ML Lead	Mitigating (Sprint 0)
RISK-004	Performance Degradation	3	3	**9 — Medium**	Tech Lead	Mitigating (Sprint 0)
RISK-005	Integration Delays	3	3	**9 — Medium**	Lead Dev	Mitigating (Sprint 0)
RISK-006	Resource Constraints	3	3	**9 — Medium**	Project Manager	Mitigating (Sprint 0)
RISK-007	AI Landscape Rapid Evolution	4	3	**12 — High**	ML Lead / Product Owner	Mitigating (Sprint 0)
RISK-008	Regulatory & Policy Changes	2	4	**8 — Medium**	DPO / Product Owner	Monitoring
RISK-009	Low User Adoption	3	4	**12 — High**	Product Owner	Mitigating (Sprint 0)

Risk Review Cadence

Review Type	Frequency	Forum	Attendees	Output
Risk standup	Weekly (5 minutes in daily standup)	Daily standup	Full team	Any new risks flagged, blockers raised
Sprint risk review	Fortnightly (end of each sprint)	Sprint review / retrospective	Full team + Product Owner	Risk register updated, ratings re-assessed, new risks added
Steering committee risk report	Monthly	Steering committee meeting	Project Manager, Tech Lead, Product Owner, DPO, Head of Dept	Executive risk summary, critical/high risk status, escalations

Review Type	Frequency	Forum	Attendees	Output
Comprehensive risk audit	Quarterly (or at major gate: Sprint 0 exit, UAT, launch)	Dedicated risk workshop	All risk owners + steering committee	Full re-assessment of all risks, new risk identification, lessons learned

Risk Metrics & Reporting

Metric	Target	Review Cadence
Open critical/high risks	0 critical unmitigated, <3 high	Weekly
Risk actions overdue	0	Fortnightly (sprint review)
New risks identified per sprint	Tracked (no target — more is better than missed risks)	Fortnightly
Risks closed / downgraded per sprint	Tracked (trend should be positive)	Fortnightly
Risk register last updated	< 14 days ago	Continuous
Mean time from risk identification to mitigation start	< 1 sprint	Monthly

Risk Escalation Path

```
Team Member identifies risk
        |
        v
Raised in standup / added to risk register
        |
        v
Risk Manager (Project Manager) assesses and scores
        |
    ┌───┴───┐
    |       |
  Low/    High/
  Medium  Critical
    |       |
    v       v
  Monitor  CCB reviews within
  in       48 hours
  register     |
               v
          Mitigation plan
          assigned to Risk Owner
               |
               v
          If unresolvable or
          risk materialises
               |
               v
          Steering Committee
          (emergency session
           for Critical risks)
```

Risk Management Cadence Per Sprint

Sprint	Risk Management Activities
Sprint 0	Risk governance established. Risk assessment framework defined. Initial risk register created (RISK-001 to RISK-009). All Sprint 0 mitigation actions executed. First steering committee risk briefing
Sprint 1-2	First sprint risk review. Validate Sprint 0 mitigations are effective. Monitor early warning indicators for scope creep and resource constraints
Sprint 3-4	Risk register updated with any new technical risks from API/DB work. Integration risk indicators monitored. Performance baseline established — validate RISK-004
Sprint 5-6	Security risk review after hardening sprint. Pen test scope agreed — validates RISK-002 mitigation. Accuracy metrics from initial testing — validates RISK-003
Sprint 7-8	UAT feedback: adoption risk (RISK-009) indicators assessed. Integration testing: RISK-005 validated. Full risk register re-assessment at UAT gate

Sprint	Risk Management Activities
Sprint 9-10	Pre-launch risk audit (comprehensive workshop). Pen test results: RISK-002 final validation. Load test results: RISK-004 final validation. DR drill: validates contingency plans. Go/no-go risk assessment for steering committee
Sprint 11-12	Production risk monitoring active. Live SLO tracking validates RISK-004. Pilot user adoption tracked validates RISK-009. Post-launch risk register handover to BAU operations team

Risk Management Exit Criteria (Sprint 0)

Before Sprint 1, the following must be confirmed:

- [ ] Risk governance structure established (Risk Manager, CCB, steering committee roles assigned)
- [ ] Risk assessment framework documented (likelihood/impact scales, rating matrix, response policy)
- [ ] Risk register created with all 9 identified risks scored, owned, and mitigation actions assigned
- [ ] Early warning indicators defined for all critical and high risks with detection methods
- [ ] Contingency plans documented for all critical and high risks
- [ ] All Sprint 0 mitigation actions completed (per individual risk sections and Sprint 0 prerequisites)
- [ ] First steering committee risk briefing delivered
- [ ] Risk review cadence added to sprint ceremonies calendar
- [ ] Risk escalation path documented and communicated to full team
- [ ] Risk register accessible to all team members (e.g., shared document, project management tool)

Success Criteria

Success Criteria Prerequisites (Sprint 0)

Success cannot be measured without agreed definitions, baselines, and measurement infrastructure. All of the following must be established **before Sprint 1 begins**.

Baseline & Measurement Setup

Action	Owner	Completion Criteria
Agree all success criteria KPIs with steering committee — no ambiguity on targets or measurement methods	Product Owner / Project Manager	KPI document signed off

Action	Owner	Completion Criteria
Identify and document current baseline for "manual AI detection review time" (pre-tool) to measure 50% reduction	Product Owner / Academic Liaison	Baseline survey completed with 10+ educators
Confirm department count and list for "70%+ adoption" target — define what constitutes a "target department"	Product Owner	Department list approved by steering committee
Define "adoption" precisely — e.g., at least 1 educator per department submitting >=5 analyses per month	Product Owner	Definition documented and approved
Provision analytics infrastructure to track business metrics (analyses per user, department, HE level)	Lead Dev / DevOps Lead	Analytics pipeline running, dashboards templated
Create user satisfaction survey template (Likert scale, SUS questionnaire, open-ended feedback)	UX Lead / Product Owner	Survey template approved, distribution plan agreed
Establish academic integrity committee feedback mechanism (quarterly presentation + feedback form)	Product Owner / Academic Liaison	First committee presentation scheduled
Define "positive feedback" threshold for academic integrity committees (e.g., formal endorsement, continued funding approval)	Product Owner	Definition documented

Technical Metrics

KPI	Target	Measurement Method	Data Source	Measurement Frequency	First Measure
Detection accuracy (overall)	>= 85%	F1 score against validation dataset (650+ documents)	Automated regression test suite	Every model change + quarterly full benchmark	Sprint 2
Detection precision	>= 85%	True positives / (True positives + False positives) on validation set	Automated regression test suite	Every model change	Sprint 2
Detection recall	>= 80%	True positives / (True positives + False negatives) on validation set	Automated regression test suite	Every model change	Sprint 2

KPI	Target	Measurement Method	Data Source	Measurement Frequency	First Measure
Accuracy per HE level (4-8)	>= 80% per level	F1 score per HE level subset of validation dataset	Automated regression test suite	Every model change	Sprint 2
System uptime	99.5%	Percentage of time all health checks pass (excluding scheduled maintenance)	Synthetic monitoring (Uptime Robot) + Prometheus	Continuous (30-day rolling)	Sprint 11 (productio
Page load time (p95)	< 2 seconds	95th percentile response time for page requests	Prometheus `http_request_duration_seconds`	Continuous	Sprint 4 (staging), Sprint 11 (productio
Analysis completion time (p95)	< 5 seconds	95th percentile for documents <= 5000 words	Prometheus `analysis_duration_seconds`	Continuous	Sprint 2 (dev), Sprint 11 (productio
Concurrent user support	>= 500	Sustained load test with 500+ virtual users, zero errors, p95 response < 2s	JMeter load test results	Sprint 7, 10 (staging), Sprint 11 (production)	Sprint 7
Code coverage	>= 80%	Line coverage across all Python modules	pytest-cov in CI pipeline	Every PR	Sprint 1
Security vulnerabilities (critical/high)	0	OWASP ZAP scan + Bandit + Trivy + pen test	CI pipeline + external pen test report	Every PR (automated) + Sprint 10 (pen test)	Sprint 1
WCAG 2.1 AA violations (critical/ serious)	0	axe-core automated scan + manual audit	CI pipeline + manual accessibility audit	Every PR (automated) + Sprint 8 (manual)	Sprint 4
API availability (health endpoint)	99.9%	Percentage of successful `/api/health` responses	Synthetic monitoring (30-second checks)	Continuous	Sprint 11

Technical Metrics — Stretch Goals

These are aspirational targets beyond the core success criteria. Meeting them is desirable but not required for launch.

KPI	Stretch Target	Core Target
Detection accuracy (overall)	>= 90%	>= 85%
System uptime	99.9%	99.5%
Analysis completion time (p95)	< 3 seconds	< 5 seconds
Concurrent user support	>= 1000	>= 500
Code coverage	>= 90%	>= 80%

User Satisfaction Metrics

KPI	Target	Measurement Method	Data Source	Measurement Frequency	First Measured
User satisfaction score	>= 4.2 / 5	Post-analysis Likert scale survey (optional, non-intrusive)	In-app survey widget	Monthly aggregate	Sprint 11 (pilot)
System Usability Scale (SUS) score	>= 72 (above average)	10-question SUS questionnaire after usability testing	Facilitated usability sessions + online survey	Sprint 4 (prototype), Sprint 8 (UAT), Sprint 12 (post-launch)	Sprint 4
Task completion rate (Quick Analysis)	>= 90%	Percentage of users who start an analysis and receive results	Analytics pipeline (funnel tracking)	Weekly	Sprint 11 (pilot)
Task completion rate (Batch Analysis)	>= 80%	Percentage of batch uploads that complete successfully	Analytics pipeline	Weekly	Sprint 11 (pilot)
Time to first analysis (new users)	< 60 seconds	Time from first login to receiving first analysis result	Analytics pipeline (event timestamps)	Monthly	Sprint 11 (pilot)
Error rate (user-facing errors)	< 2%	Percentage of user actions resulting in an error message	Sentry + analytics pipeline	Continuous	Sprint 4 (staging)
Support ticket resolution time (Tier 1)	< 24 hours	Mean time from ticket creation to resolution	Support ticketing system	Weekly	Sprint 11 (pilot)

KPI	Target	Measurement Method	Data Source	Measurement Frequency	First Measured
Support ticket resolution time (Tier 2)	< 8 hours	Mean time from ticket creation to resolution	Support ticketing system	Weekly	Sprint 11 (pilot)
Net Promoter Score (NPS)	>= 30 (good)	"How likely are you to recommend this tool?" (0-10 scale)	Quarterly educator survey	Quarterly	3 months post-launch
Feature request rate	Tracked (no target)	Number of feature requests submitted per month	Support system + feedback form	Monthly	Sprint 12

User Satisfaction — Measurement Instruments

Instrument	When Administered	Sample Size Target	Owner
In-app micro-survey (1 question: "Was this result helpful?" + optional comment)	After every 5th analysis (non-blocking)	All active users (opt-in, ~30% expected response)	Product Owner
SUS Questionnaire (10 standardised questions)	After usability testing sessions + quarterly to all users	20+ responses per measurement point	UX Lead
Post-pilot survey (satisfaction, usefulness, accuracy perception, improvement suggestions)	End of pilot period (Sprint 12)	Target 50+ responses from pilot users	Product Owner
NPS survey (single question + open comment)	Quarterly (starting 3 months post-launch)	All active users (target 25%+ response rate)	Product Owner
Academic integrity committee presentation	Quarterly	All committee members	Product Owner / Academic Liaison
Educator interview (qualitative)	Sprint 8 (UAT) + 3 months post-launch	10+ educators across all personas	UX Lead

Business Impact Metrics

KPI	Target	Baseline	Measurement Method	Data Source	Measurement Frequency	First Measured
Department adoption	>= 70% of target departments within 6 months	0% (pre-launch)	Departments with >= 1 active educator submitting >= 5 analyses/month	Analytics pipeline (user department mapping)	Monthly	Sprint 11 (pilot)
Active user count (monthly)	>= 100 within 3 months, >= 250 within 6 months	0 (pre-launch)	Unique users submitting >= 1 analysis in a 30-day window	Analytics pipeline	Monthly	Sprint 11
Analyses per month	>= 1000 within 3 months, >= 5000 within 6 months	0 (pre-launch)	Total analyses submitted across all users	Analytics pipeline	Monthly	Sprint 11
Reduction in manual review time	>= 50% reduction within 6 months	Baseline measured in Sprint 0 (current manual time per case)	Time-tracking survey to educators: "How long does AI detection review take per case?"	Pre/post comparison survey	Quarterly (starting 3 months post-launch)	3 months post-launch
Academic integrity committee endorsement	Formal endorsement within 6 months	No endorsement (pre-launch)	Committee meeting minutes, formal recommendation letter	Committee feedback mechanism	Quarterly	3 months post-launch
Cost per analysis	<= £0.02 in steady state	Established Sprint 11	Total cloud spend / total analyses	Cost monitoring + analytics pipeline	Monthly	Sprint 11
Repeat usage rate	>= 60% of users return within 30 days	0% (pre-launch)	Percentage of users who submit a second analysis within 30 days of their first	Analytics pipeline	Monthly	1 month post-launch
Batch vs single analysis ratio	Tracked (no target — informs product decisions)	N/A	Percentage of analyses submitted via batch upload vs single analysis	Analytics pipeline	Monthly	Sprint 11

Business Impact — Leading Indicators (tracked from pilot)

These early signals predict whether 6-month targets will be met. If leading indicators are off-track, corrective action is taken immediately rather than waiting for lagging metrics.

Leading Indicator	Healthy Signal	Warning Signal	Corrective Action
Pilot user activation rate (first analysis within 7 days of account creation)	>= 80%	< 50%	Targeted onboarding emails, 1:1 training offers
Weekly active users during pilot	Growing week-over-week	Flat or declining	Product Owner interviews inactive users, UX improvements prioritised
Analyses per active user per week	>= 2	< 1	Investigate barriers (UX, performance, trust). Quick-fix sprint if needed
Educator champion engagement	Champions actively promoting in their departments	Champions disengaged or negative	Product Owner meets with champions, addresses concerns directly
Support ticket volume (pilot)	Low and decreasing	High or increasing	Review common issues, improve onboarding/ docs, fix recurring bugs
Educator qualitative feedback (pilot)	"Useful", "saves time", "reliable"	"Confusing", "inaccurate", "too slow"	UX/ML improvements prioritised for Sprint 12

Success Criteria — Gate Reviews

Success criteria are formally reviewed at 4 project gates. Each gate has specific pass/ fail criteria and decision outcomes.

Gate 1: Sprint 0 Exit (Pre-Development)

Criterion	Pass	Fail Action
All Sprint 0 exit criteria met (risk, testing, UX, production, monitoring, success criteria)	All checkboxes ticked	Do not proceed to Sprint 1. Address gaps. Re-review within 1 week
Steering committee sign-off	Formal approval received	Escalate blockers to steering committee chair

Gate 2: UAT Completion (Sprint 8)

Criterion	Pass	Fail Action
Detection accuracy >= 85% on validation set	F1 >= 0.85	Block production preparation. ML Lead prioritises refinement in Sprint 9
SUS score >= 72 from UAT participants	Score >= 72	UX redesign sprint before production. Delay launch by 1-2 sprints if necessary
UAT test scenarios >= 80% passed	>= 80% pass rate	Fix all P1/P2 defects. Re-run failed scenarios. Re-review at Sprint 9
Zero P1/P2 defects open	0 open	Sprint 9 starts with defect fix sprint before production preparation
Stakeholder sign-off	Product Owner + 2 educator representatives approve	Address feedback, re-run UAT for disputed areas

Gate 3: Go/No-Go for Production (Sprint 10)

Criterion	Pass	Fail Action
Load test: 500+ concurrent users sustained	Zero errors, p95 < 2s	Performance optimisation sprint. Delay launch. Consider reduced user target
Pen test: zero critical/high findings	All findings resolved	Security fix sprint before launch. No exceptions
DR drill completed successfully	Failover and recovery within RPO/RTO targets	Fix DR gaps. Re-run drill. Delay launch until DR is verified
All risk register critical/high risks have active mitigations	No unmitigated critical/high risks	Risk workshop. Steering committee reviews go/no-go
Production infrastructure validated	All IaC deployed, health checks passing, monitoring live	DevOps sprint to resolve infrastructure gaps
Steering committee approval	Formal go-ahead	Address all steering committee concerns before re-submission

Gate 4: Post-Launch Review (Sprint 12 + 30 days)

Criterion	Target	Actual (filled post-launch)	Status
System uptime (30-day)	>= 99.5%	TBD	
Detection accuracy (production)	>= 85%	TBD	
User satisfaction score	>= 4.2 / 5	TBD	

Criterion	Target	Actual (filled post-launch)	Status
Task completion rate	>= 90%	TBD	
Active users (first month)	>= 50	TBD	
P1/P2 incidents (first 30 days)	<= 2	TBD	
Support ticket resolution (Tier 1)	< 24 hours avg	TBD	
Pilot user retention (30-day return rate)	>= 60%	TBD	
Departments with active users	>= 3 (pilot target)	TBD	

Decision	Criteria
Full rollout approved	>= 7 of 9 criteria met, no critical failures
Conditional rollout	5-6 criteria met, improvement plan for gaps within 1 sprint
Rollout paused	< 5 criteria met or any critical failure (uptime, accuracy, security) — steering committee review required

---

Success Criteria Tracking Dashboard

Dashboard Panel	Metrics Displayed	Audience	Data Source
Technical Health	Accuracy (overall + per HE level), uptime, response times, error rate, code coverage	Tech Lead, Developers	Prometheus, pytest-cov, validation test suite
User Engagement	Active users, analyses/day, batch vs single, new user activation, retention	Product Owner, Steering Committee	Analytics pipeline
User Satisfaction	Satisfaction score, SUS, NPS, task completion rate, support ticket trends	Product Owner, UX Lead	In-app surveys, support system
Business Impact	Department adoption, manual review time reduction, cost per analysis, committee feedback	Steering Committee, Product Owner	Analytics pipeline, surveys, committee minutes
Gate Status	Current gate, criteria pass/fail status, next gate date, blockers	Project Manager, Steering Committee	Manual update (Project Manager)

Success Criteria Cadence Per Sprint

Sprint	Success Criteria Activities
Sprint 0	All KPIs agreed and signed off. Baselines measured (manual review time). Analytics infrastructure provisioned. Survey templates created. Gate criteria defined. Success dashboard templated
Sprint 1-2	Code coverage tracking active. First accuracy measurement against validation set. Technical metrics baseline established
Sprint 3-4	Staging performance metrics tracked. First axe-core accessibility baseline. Analytics pipeline capturing user events in staging
Sprint 5-6	Accuracy metrics refined with batch processing data. Security scan results tracked. Business metrics pipeline tested with synthetic data
Sprint 7-8	**Gate 2 (UAT):** SUS measured, UAT pass rate calculated, defect counts reviewed. Accuracy validated against UAT samples. First user satisfaction data from UAT participants
Sprint 9-10	**Gate 3 (Go/No-Go):** Load test results reviewed, pen test results reviewed, DR drill results reviewed. Steering committee go/no-go presentation. All dashboards populated with staging data
Sprint 11-12	**Production metrics live:** Uptime, response times, error rates tracked in real-time. Pilot user engagement and satisfaction tracked daily. Leading indicators monitored. Support ticket metrics active
Post-launch (30 days)	**Gate 4 (Post-Launch Review):** All metrics evaluated against targets. Steering committee presentation. Full rollout / conditional rollout / pause decision made

Success Criteria Exit Criteria (Sprint 0)

Before Sprint 1, the following must be confirmed:

- [ ] All success criteria KPIs agreed and signed off by steering committee
- [ ] "Adoption" and "target department" precisely defined and documented
- [ ] Baseline manual review time measured (survey of 10+ educators)
- [ ] Analytics infrastructure provisioned and capturing events in dev
- [ ] Survey templates created (in-app micro-survey, SUS, post-pilot, NPS)
- [ ] Gate review criteria documented (Gates 1-4)
- [ ] Success criteria tracking dashboard templated
- [ ] Academic integrity committee feedback mechanism established
- [ ] Leading indicators defined with healthy/warning thresholds and corrective actions
- [ ] First committee presentation date scheduled

Conclusion

Plan Summary

This comprehensive DevOps sprint plan provides a structured roadmap for developing and deploying a production-ready AI Detection Tool for UK Higher Education institutions. The plan has been expanded from the original 12-week delivery outline into a fully operationalised programme with seven integrated workstreams, each following a consistent structure:

#	Workstream	Section	Sprint 0 Prerequisites	Per-Sprint Cadence	Exit Criteria
1	Risk Management	Link	Yes	Yes	Yes
2	Testing Strategy	Link	Yes	Yes	Yes
3	UX/UI Requirements	Link	Yes	Yes	Yes
4	Production Infrastructure	Link	Yes	Yes	Yes
5	Monitoring & Maintenance	Link	Yes	Yes	Yes
6	Success Criteria	Link	Yes	Yes	Yes
7	Sprint Delivery (1-12)	Link	N/A (Sprint 0 is the prerequisite)	Built-in	Gate reviews

Every workstream follows the same pattern:

1. **Prerequisites (Sprint 0)** — what must be in place before development begins
2. **Standards & specifications** — detailed, measurable requirements with named owners
3. **Per-sprint cadence** — activities mapped to every sprint, ensuring nothing is deferred
4. **Metrics & targets** — quantified success measures with early warning indicators
5. **Exit criteria** — checklists that gate progression to the next phase

---

Project Timeline Overview

Phase	Duration	Key Activities	Gate
Sprint 0	1-2 weeks	Risk governance, test tooling, UX research & design system, cloud/IaC/ monitoring provisioning, KPI baselines, dataset curation	**Gate 1:** Sprint 0 Exit — all 7 workstream exit criteria met

Phase	Duration	Key Activities	Gate
Sprints 1-2	Weeks 1-2	Core detection engine, HE level integration, unit tests, CI/CD pipeline, application instrumentation	—
Sprints 3-4	Weeks 3-4	Flask API, PostgreSQL, frontend build, staging environment, API testing	—
Sprints 5-6	Weeks 5-6	PDF reports, batch processing, user management, RBAC, security hardening, GDPR tooling	—
Sprints 7-8	Weeks 7-8	System integration, documentation, UAT	**Gate 2:** UAT — accuracy >=85%, SUS >=72, >=80% scenarios passed, zero P1/P2
Sprints 9-10	Weeks 9-10	Production infrastructure, load testing (1000+), pen test, DR drill	**Gate 3:** Go/No-Go — steering committee production approval
Sprints 11-12	Weeks 11-12	Soft launch, pilot users, training, full launch, retrospective	—
Post-Launch	30 days	Production monitoring, pilot metrics, user feedback, support	**Gate 4:** Post-Launch Review — full rollout / conditional / pause decision

Total timeline: 14-16 weeks (Sprint 0 + 12 delivery weeks + 30-day post-launch review)

Consolidated Sprint 0 Master Checklist

This is the single authoritative checklist for Sprint 0 completion. All items must be confirmed before Sprint 1 begins. Each item links to its detailed definition in the relevant section.

Risk Mitigation (9 items)

#	Item	Owner	Section
1	PRD signed off by all stakeholders	Product Owner	Sprint 0 Risk Mitigation
2	DPIA completed and submitted	DPO	RISK-002
3	Incident Response Plan documented and tabletop-tested	Security Lead	RISK-002
4	Threat model (STRIDE) completed	Security Lead	RISK-002

#	Item	Owner	Section
5	API contracts (OpenAPI) defined and committed	Lead Dev	Sprint 0 Risk Mitigation
6	Performance budgets documented in NFRs	Lead Dev	Sprint 0 Risk Mitigation
7	All team members have working dev environments	DevOps Lead	Sprint 0 Risk Mitigation
8	Change Control Board established with charter	Project Manager	Sprint 0 Risk Mitigation
9	Validation dataset curated (650+ documents)	ML Lead	Sprint 0 Risk Mitigation

Testing (8 items)

#	Item	Owner	Section
10	All test tooling installed and configured in CI (pytest, Selenium, JMeter, OWASP ZAP, Pact, axe-core, Bandit)	QA Lead / Lead Dev	Testing Prerequisites
11	Test environments provisioned (local, CI, staging)	DevOps Lead	Test Environment Strategy
12	Quality gates defined, committed, and enforced as blocking CI checks	QA Lead	Quality Gate Definitions
13	Test data factories created for core models	QA Lead	Testing Prerequisites
14	Defect management process agreed and tooling configured	QA Lead	Defect Management
15	Test reporting dashboards live	QA Lead	Testing Prerequisites
16	Sprint testing cadence communicated to full team	QA Lead	Testing Cadence Per Sprint
17	Validation dataset versioned across all categories	ML Lead / QA Lead	Test Data Requirements

UX/UI (8 items)

#	Item	Owner	Section
18	User research completed (5+ educator interviews)	UX Lead	User Personas & Research
19	Personas validated and signed off by stakeholders	UX Lead / Product Owner	User Personas & Research
20	Competitor analysis documented	UX Lead	UX/UI Prerequisites

#	Item	Owner	Section
21	Design system defined (colours, typography, spacing, grid, icons)	UX Lead	Design System & Component Library
22	Low-fidelity wireframes approved for all pages	UX Lead	Wireframes & Prototyping
23	Interactive prototype usability-tested with 3+ educators	UX Lead	Wireframes & Prototyping
24	High-fidelity mockups approved for Sprint 3-4 pages	UX Lead	Wireframes & Prototyping
25	Accessibility standards documented and communicated to developers	UX Lead / Frontend Lead	Accessibility Requirements

Production Infrastructure (6 items)

#	Item	Owner	Section
26	Cloud provider selected, accounts provisioned (dev/staging/production)	DevOps Lead	Cloud Provider & Account Setup
27	IaC repository created with base modules committed	DevOps Lead	Infrastructure as Code Standards
28	Network architecture documented and approved	DevOps Lead	Network Architecture
29	DR plan documented with RPO/RTO targets, test-restore verified	DevOps Lead	Disaster Recovery & Business Continuity
30	CI/CD pipeline skeleton running (build + lint + test)	DevOps Lead	CI/CD Pipeline Architecture
31	TLS certificates and domain DNS configured	DevOps Lead	Cloud Provider & Account Setup

Monitoring & Maintenance (8 items)

#	Item	Owner	Section
32	Prometheus + Grafana provisioned in dev, scraping app metrics	DevOps Lead	Observability Tooling Setup
33	ELK stack provisioned in dev, indexing logs	DevOps Lead	Observability Tooling Setup
34	PagerDuty configured with on-call schedules and test page verified	Project Manager / DevOps Lead	On-Call & Operational Readiness
35	SLIs, SLOs, and error budget policy approved	DevOps Lead / Tech Lead	Service Level Objectives

#	Item	Owner	Section
36	Logging and tracing standards documented	DevOps Lead / Lead Dev	Logging Standards / Distributed Tracing Standards
37	All 12 must-have runbooks written and committed	DevOps Lead	Required Runbooks
38	Cloud budget alerts configured (50%, 80%, 100%, 120%)	DevOps Lead	Cost Monitoring & Optimisation
39	`#incidents` and `#alerts` Slack channels live with integrations	DevOps Lead	On-Call & Operational Readiness

Risk Management (6 items)

#	Item	Owner	Section
40	Risk governance structure established (Risk Manager, CCB, steering committee)	Project Manager	Risk Governance Structure
41	Risk assessment framework documented (likelihood/impact matrix, response policy)	Project Manager	Risk Assessment Framework
42	Risk register created with all 9 risks scored, owned, and mitigations assigned	Project Manager	Risk Register Summary
43	Early warning indicators and contingency plans for all critical/high risks	All Risk Owners	Risk Register
44	First steering committee risk briefing delivered	Project Manager	Risk Review Cadence
45	Risk review cadence added to sprint ceremonies calendar	Project Manager / Scrum Master	Risk Management Cadence Per Sprint

Success Criteria (8 items)

#	Item	Owner	Section
46	All success criteria KPIs agreed and signed off by steering committee	Product Owner	Success Criteria Prerequisites
47	"Adoption" and "target department" precisely defined	Product Owner	Business Impact Metrics
48	Baseline manual review time measured (10+ educator survey)	Product Owner / Academic Liaison	Success Criteria Prerequisites
49	Analytics infrastructure provisioned and capturing events	Lead Dev / DevOps Lead	Success Criteria Prerequisites
50	Survey templates created (micro-survey, SUS, post-pilot, NPS)	UX Lead / Product Owner	Measurement Instruments

#	Item	Owner	Section
51	Gate review criteria documented (Gates 1-4)	Product Owner / Project Manager	Gate Reviews
52	Success criteria tracking dashboard templated	DevOps Lead / Product Owner	Success Criteria Tracking Dashboard
53	First academic integrity committee presentation scheduled	Product Owner / Academic Liaison	Success Criteria Prerequisites

Total Sprint 0 items: 53

Immediate Next Steps

The following actions should be taken **immediately** upon approval of this plan, before Sprint 0 formally begins.

#	Action	Owner	Deadline	Dependency
1	Circulate this plan to all stakeholders for review and comment	Project Manager	Day 1	None
2	Schedule Sprint 0 kick-off meeting	Project Manager	Within 3 days	Action 1
3	Confirm steering committee membership and schedule first meeting	Project Manager	Within 5 days	Action 1
4	Confirm full team allocation and start dates	Project Manager / HR	Within 5 days	None
5	Initiate cloud provider account creation process	DevOps Lead	Within 3 days	Budget approval (Action 8)
6	Initiate DPO engagement for DPIA	Project Manager / DPO	Within 3 days	None
7	Begin educator recruitment for user research interviews	UX Lead	Within 3 days	None
8	Submit budget request for cloud infrastructure, external consulting, and tooling licences	Project Manager	Within 5 days	None
9	Request validation dataset source materials from academic partners	ML Lead / Academic Liaison	Within 5 days	Consent process (Action 6)
10	Obtain stakeholder sign-off on this plan (see below)	Project Manager	Within 10 days	Actions 1-3

Phase 2 Roadmap Considerations

The following features and improvements are explicitly **out of scope** for the initial 12-week delivery but are recommended for Phase 2 planning (post-launch + 3 months). These items should be captured in the product backlog but not addressed until Phase 1 success criteria are met at Gate 4.

Feature / Improvement	Rationale	Estimated Effort	Priority
Institutional SSO integration (if not completed in Phase 1)	Reduces login friction, aligns with institutional IT strategy	2-3 sprints	High
LMS integration (Moodle, Blackboard, Canvas)	Direct submission analysis within existing workflows	3-4 sprints	High
Multi-language support	International students, multilingual submissions	2-3 sprints	Medium
API access for third-party integrations	Programmatic access for institutional tools	1-2 sprints	Medium
Advanced analytics dashboard for department heads	Trend analysis, cohort comparisons, longitudinal data	2-3 sprints	Medium
Model retraining pipeline	Automated retraining from educator feedback loop data	3-4 sprints	High
Plagiarism detection integration	Combine AI detection with traditional plagiarism checking	2-3 sprints	Low
Student self-check mode	Allow students to check their own work before submission (requires policy decision)	1-2 sprints	Low (policy-dependent)
Mobile native app	Dedicated iOS/Android app for on-the-go analysis	4-6 sprints	Low

---

Stakeholder Sign-Off

This plan requires formal approval from the following stakeholders before Sprint 0 begins. Sign-off confirms agreement with the scope, timeline, resource requirements, risk appetite, and success criteria defined in this document.

Role	Name	Date	Signature
Head of Department / Sponsor	_____	_//_	_____
Chief Technology Officer	_____	_//_	_____
Data Protection Officer	_____	_//_	_____
Product Owner	_____	_//_	_____

Role	Name	Date	Signature
Tech Lead	_____	_/_	_____
Project Manager	_____	_/_	_____
Academic Integrity Committee Representative	_____	_/_	_____

Document Governance

Attribute	Value
Document Title	AI Detection Tool — DevOps Sprint Plan & Implementation Guide
Version	2.0
Status	Draft — Awaiting Stakeholder Approval
Author	Project Manager / Tech Lead
Last Updated	_____
Next Review Date	Sprint 0 kick-off meeting
Distribution	Steering committee, full delivery team, DPO, academic integrity committee chair
Classification	Internal — Not for public distribution

Version History

Version	Date	Author	Changes
1.0	_____	_____	Initial sprint plan (12-week delivery outline)
2.0	_____	_____	Comprehensive expansion: Sprint 0 prerequisites added across all workstreams. Risk register expanded to 9 risks with full assessment framework. Testing, UX/UI, production, monitoring, and success criteria sections expanded with prerequisites, standards, cadence, metrics, and exit criteria. Gate reviews defined. Phase 2 roadmap added. Stakeholder sign-off page added

---

Closing Statement

This plan represents a **risk-first, quality-driven** approach to delivering a production-ready AI Detection Tool for UK Higher Education. The addition of Sprint 0 — with 53 prerequisite items across 7 workstreams — ensures that the team enters development with a secure foundation, clear standards, and measurable targets.

The 4 formal gate reviews (Sprint 0 Exit, UAT, Go/No-Go, Post-Launch) provide structured decision points where the steering committee can assess readiness, approve progression, or pause delivery if quality or risk thresholds are not met. This protects the institution from premature deployment and ensures the tool meets the high standards required for academic integrity work.

Success depends on close collaboration between development, operations, and academic stakeholders, with continuous iteration based on user feedback and emerging requirements. The 9 identified risks are actively managed through a formal governance framework with early warning indicators, contingency plans, and regular review cadence.

The project emphasizes security, scalability, and user experience to deliver a tool that supports academic integrity while respecting the complexities of assessing student work in the age of AI. By investing in Sprint 0, the team maximises the probability of delivering a system that educators trust, students are treated fairly by, and institutions can rely on.