Product Requirements Document (PRD): Course Feedback Management System

1. Overview

1.1 Product Vision

A comprehensive, scalable platform for educational institutions to collect, manage, and analyze course feedback efficiently, enabling data-driven improvements in teaching quality and student satisfaction.

1.2 Problem Statement

Educational institutions struggle with:

- Manual, paper-based feedback collection processes

- Inefficient feedback analysis and reporting

- Lack of real-time insights into course performance

- Difficulty tracking feedback trends over time

- Limited ability to enforce feedback policies and limits

1.3 Solution

A full-stack web application that automates course feedback management with role-based access control, real-time analytics, and automated workflows.

2. Goals & Objectives

2.1 Primary Goals

- **Streamline Feedback Collection**: Digital, structured feedback submission process

- **Enable Data-Driven Decisions**: Real-time analytics and reporting dashboard

- **Ensure System Security**: Robust role-based access control and data protection

- **Support Scalability**: Multi-tenant architecture for multiple organizations

2.2 Success Metrics

- Reduce feedback processing time by 80%

- Increase feedback submission rate by 50%

- Achieve 99% system uptime

- Support 10,000+ concurrent users

- Process 100,000+ feedback entries annually

3. Target Users & Personas

3.1 Students

- **Needs**: Easy feedback submission, transparency in process
- **Pain Points**: Complicated forms, no confirmation of action taken
- **Usage**: Submit feedback, view submission history

3.2 Instructors/Faculty

- **Needs**: Access to feedback, analytics, improvement insights
- **Pain Points**: Delayed feedback, lack of actionable insights
- **Usage**: View course feedback, access analytics

3.3 Administrators

- **Needs**: System management, reporting, policy enforcement
- **Pain Points**: Manual reporting, compliance issues
- **Usage**: Manage courses, users, system configuration

3.4 Department Heads

- **Needs**: Department-wide insights, trend analysis
- **Pain Points**: Siloed data, difficult comparisons
- **Usage**: View departmental analytics, comparative reports

4. Core Features & Requirements

4.1 Authentication & Authorization

Must Have:

- Multi-tenant user authentication
- Role-based access control (Student, Instructor, Admin, Department Head)
- JWT token-based security
- Session management

Acceptance Criteria:

- Users can log in with credentials

- System enforces role-based permissions

- Session timeout after 30 minutes of inactivity

- Password reset functionality

4.2 Course Management

Must Have:

- CRUD operations for courses

- Course status (Active/Inactive)

- Course metadata (code, name, description, instructor, department)

- Bulk course import/export

Acceptance Criteria:

- Admin can create, edit, delete courses

- Courses can be marked active/inactive

- Search and filter courses by various criteria

- Support for course categories/departments

4.3 Feedback Management

Must Have:

- Structured feedback submission (1-5 ratings + comments)

- File attachment support (PDF, JPG, PNG up to 5MB)

- Feedback validation (active courses only)

- Per-course feedback limits enforcement

- Edit/delete own feedback (within time window)

Acceptance Criteria:

- Students can submit feedback only for active courses

- System prevents duplicate submissions per course

- File uploads are validated for type and size

- Feedback can be edited within 24 hours of submission

4.4 Dashboard & Analytics

Must Have:

- Real-time feedback statistics

- Top-rated courses display

- Department-wise performance metrics

- Trend analysis over time

- Exportable reports

Acceptance Criteria:

- Dashboard loads within 3 seconds

- Charts and graphs are interactive

- Data exports in CSV/PDF format

- Real-time updates without page refresh

4.5 File Management

Must Have:

- Secure file upload and storage

- File type validation (PDF, JPG, PNG)

- Size limitation (5MB per file)

- Virus scanning integration

- Secure file download

Acceptance Criteria:

- Files are scanned for malware

- Upload progress indication

- File preview capability

- Secure access control for downloads

4.6 Background Processing

Must Have:

- Automated inactive course checks

- Email notifications for important events

- Data cleanup and archiving

- Report generation scheduling

Acceptance Criteria:

- Daily job runs to check inactive courses

- Email notifications for system events

- Automated data archiving after 3 years

- Scheduled report generation

5. Technical Requirements

5.1 Performance

- Page load time: < 3 seconds

- API response time: < 500ms

- Support 10,000+ concurrent users

- Database queries optimized with indexing

5.2 Security

- OWASP compliance

- SQL injection prevention

- XSS protection

- CSRF protection

- Secure file upload validation

- Regular security audits

5.3 Scalability

- Horizontal scaling capability

- Database connection pooling

- Caching strategy implementation

- CDN for static assets

5.4 Reliability

- 99.5% uptime SLA

- Automated backup system

- Disaster recovery plan

- Monitoring and alerting

6. Non-Functional Requirements

6.1 Usability

- Intuitive user interface

- Mobile-responsive design

- Accessibility compliance (WCAG 2.1 AA)

- Multi-language support readiness

6.2 Compatibility

- Browser support: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+

- Mobile support: iOS Safari, Chrome Mobile

- Database: SQL Server 2019+

6.3 Maintainability

- Comprehensive logging

- API documentation

- Code coverage > 60%

- Modular architecture

7. Data Model

7.1 Core Entities

- **User**: Id, Username, Email, Role, TenantId

- **Course**: Id, Code, Name, Description, Instructor, Department, IsActive, TenantId

- **Feedback**: Id, CourseId, UserId, Rating, Comment, AttachmentUrl, CreatedAt

- **Tenant**: Id, Name, Settings

7.2 Key Relationships

- User ↔ Feedback (One-to-Many)

- Course ↔ Feedback (One-to-Many)

- Tenant ↔ Users/Courses (One-to-Many)

8. Integration Requirements

8.1 External Systems

- **Email Service**: SMTP/SendGrid for notifications

- **File Storage**: Local file system with cloud readiness

- **Monitoring**: Application performance monitoring

- **Authentication**: Future OAuth/SSO readiness

8.2 API Requirements

- RESTful API design

- OpenAPI/Swagger documentation

- Rate limiting implementation

- API versioning strategy

9. Deployment & Infrastructure

9.1 Development Environment

- Local development setup

- Docker containerization readiness

- CI/CD pipeline integration

9.2 Production Environment

- Cloud hosting capability

- Load balancer configuration

- Database clustering

- SSL certificate management

10. Compliance & Legal

10.1 Data Privacy

- GDPR compliance for EU users

- Data encryption at rest and in transit

- Privacy policy implementation

- Data retention policies

10.2 Accessibility

- WCAG 2.1 AA compliance

- Screen reader compatibility

- Keyboard navigation support

- Color contrast requirements

11. Success Criteria & KPIs

11.1 Business Metrics

- User adoption rate (> 80% of target users)

- Feedback submission completion rate (> 90%)

- System uptime (> 99.5%)

- User satisfaction score (> 4/5)

11.2 Technical Metrics

- API response time (< 500ms)

- Page load time (< 3s)

- Error rate (< 0.1%)

- Concurrent user support (> 10,000)

12. Risks & Mitigations

12.1 Technical Risks

- **Performance degradation**: Implement caching and optimization

- **Security vulnerabilities**: Regular security audits and updates

- **Data loss**: Automated backups and disaster recovery

12.2 Business Risks

- **Low user adoption**: User training and change management

- **Feature misuse**: Proper access controls and monitoring

- **Scalability issues**: Load testing and capacity planning

13. Future Enhancements (V2-V3)

13.1 Phase 2

- Advanced analytics with machine learning

- Mobile application

- Integration with LMS systems

- Real-time notifications

13.2 Phase 3

- AI-powered sentiment analysis

- Predictive analytics for course success

- Advanced reporting and benchmarking

- API for third-party integrations

14. Timeline & Milestones

14.1 Phase 1 (Months 1-3)

- Core platform development

- Basic feedback management

- User authentication

- Initial dashboard

14.2 Phase 2 (Months 4-6)

- Advanced analytics

- File upload capabilities

- Background jobs

- Multi-tenant enhancements

14.3 Phase 3 (Months 7-9)

- Performance optimization

- Security hardening

- User acceptance testing

- Production deployment

15. Approval

Product Owner: _____

Technical Lead: _____

Stakeholder: _____

Date: _____