Athenaeum AI Study Companion MVP

Functional Requirements Specification - Streamlined Version

Drafted by: Austen Jones (austenj@me-too.net) for Paul Rampedi (paulrampedi@gmail.com)

Version 1.0 – 25 January 2025

1. Project Overview

1.1 Purpose

To create a minimal but effective AI-powered web-based study companion for UNISA's Applied Mathematics and Computer Science modules, focusing solely on core features needed to validate the concept.

1.2 Scope

In Scope:

- Web interface for syllabus access and AI interaction
- Focus on UNISA Applied Mathematics and Computer Science modules
- Basic user management
- Core AI tutoring functionality
- Simple feedback collection

Out of Scope (for this phase):

- Mobile applications
- Progress tracking
- Study scheduling
- Analytics dashboard
- Administrative interface
- Split-screen functionality

2. Core Functionality

2.1 User Management

- Basic email/password authentication
- Single user role (student)
- Password reset via email

2.2 Content Structure

- Display module syllabus content
- Organize by:
  - Module code and name
  - Main topics
  - Learning outcomes
  - Basic navigation between topics

2.3 AI Study Assistant

- Accept natural language questions about course content
- Provide concept explanations
- Support mathematical notation rendering
- Support code syntax display for CS topics
- Maintain context within a single session

2.4 Feedback Collection

- Simple “Was this helpful?” button after AI responses
- Basic feedback form for general comments
- Error reporting button

3. Technical Requirements

3.1 Platform

- Web browser based (Chrome, Firefox, Safari, Edge)
- Desktop/laptop focused
- Standard responsive web design

3.2 Performance

- AI response times under 5 seconds
- Support for 30 concurrent users
- 99% uptime during SA business hours

3.3 Security

- Basic user authentication
- Password encryption

- HTTPS implementation

4. User Interface

4.1 Main Interface

- Clean, simple design
- Topic/module selection
- AI chat interface
- Basic navigation menu

4.2 Chat Interface

- Text input for questions
- Display of AI responses
- Math equation rendering
- Code block formatting
- Simple feedback buttons

5. Development Phases

Phase 1 (Weeks 1-2)

- Basic system setup
- User authentication
- Content structure implementation

Phase 2 (Weeks 3-4)

- AI integration
- Chat interface development
- Math/code rendering implementation

Phase 3 (Week 5)

- Testing and refinement
- Initial content loading
- Bug fixes

6. Pilot Program

6.1 Parameters

- Duration: 4 weeks

- Users: Up to 30 students
- Modules: 2-3 core first-year modules
- Focus on high-difficulty topics

6.2 Success Metrics

- Usage frequency
- AI response helpfulness ratings
- User feedback responses
- System reliability

7. Deliverables

1. Web-based chat interface
2. Syllabus content structure
3. Basic user documentation
4. Pilot results summary

8. Technical Stack (tentative)

- Frontend: React.js (or similar)
- AI: Integration with existing models via API
- Database: MySQL
- Hosting: Dedicated Server

9. Assumptions and Dependencies

- Access to UNISA syllabus data
- Available test users
- Stable AI API access