2. PROJECT UNDERSTANDING DOCUMENT

2.1 Problem Statement

In the contemporary educational landscape, students pursuing higher education, particularly in technical fields like Master of Computer Applications (MCA), encounter several critical challenges in managing their academic materials and optimizing their learning processes:

- **Information Overload:** Students receive voluminous course syllabi in PDF format containing complex hierarchical information about course units, topics, and learning outcomes, making manual extraction and organization time-consuming and error-prone.
- **Fragmented Resources:** Academic materials are scattered across multiple platforms - university portals, email attachments, physical documents, and personal storage, leading to inefficient access and retrieval.
- **Limited Study Assistance:** Traditional learning methods lack personalized, on-demand guidance for clarifying doubts and understanding complex topics outside of classroom hours.
- **Exam Preparation Challenges:** Students struggle to systematically organize and categorize previous year questions, making targeted exam preparation difficult.
- **Lack of Progress Tracking:** Absence of integrated tools to monitor study progress, set goals, and manage academic todos results in suboptimal time management.

2.2 Proposed Solution

Study in Woods addresses these challenges through an integrated, AI-powered platform that combines intelligent document processing, conversational AI, and comprehensive academic management capabilities.

2.2.1 Solution Components

Component 1: Intelligent Syllabus Processing

Automatically extracts structured information from syllabus PDFs using advanced NLP (Natural Language Processing) techniques powered by DigitalOcean's GradientAI platform (Llama 3.3 70B model), converting unstructured document content into organized, searchable data.

Component 2: AI-Powered Study Assistant

Provides students with an intelligent chat interface that understands context from uploaded course materials, answers questions, explains concepts, and offers personalized study guidance through conversational AI.

Component 3: Centralized Academic Hub

Organizes all academic content in a hierarchical structure (University → Course → Semester → Subject → Documents) with intuitive navigation and powerful search capabilities.

Component 4: Examination Support System

Extracts and categorizes questions from past year papers, automatically linking them to relevant syllabus topics for efficient exam preparation.

Component 5: Progress Monitoring Dashboard

Tracks user activity, study patterns, and engagement metrics, enabling students to monitor their academic progress and identify areas requiring attention.

2.3 Scope of the Project

2.3.1 In-Scope Features

The following functionalities are included in the current project scope:

1. **User Authentication and Authorization**
   - Registration with email verification
   - Secure login with JWT-based authentication
   - Password reset and recovery mechanisms
   - Role-based access control (Student, Administrator)
2. **Academic Hierarchy Management**
   - University creation and configuration
   - Course management with detailed information
   - Semester organization (up to 8 semesters)
   - Subject creation with AI-generated descriptions
3. **Document Management System**
   - PDF upload to DigitalOcean Spaces cloud storage
   - Automatic document indexing and categorization
   - Version control for updated syllabi
   - Secure document access and retrieval
4. **AI-Powered Syllabus Extraction**
   - Automated parsing of syllabus PDFs
   - Extraction of units, topics, subtopics, hours
   - Learning outcome identification
   - Real-time progress tracking during extraction
   - Server-Sent Events (SSE) for live status updates
5. **Conversational AI Interface**
   - Subject-specific chat agents with contextual awareness
   - Streaming responses for real-time interaction

- Integration with knowledge bases containing course documents
- Chat history persistence and retrieval
- Multi-turn conversations with context retention

6. **Past Year Questions (PYQ) Management**
    - PDF upload for examination papers
    - AI-based question extraction
    - Categorization by question type (MCQ, Short Answer, Essay)
    - Linking questions to syllabus topics
7. **Analytics and Reporting**
    - User activity tracking and visualization
    - Document access metrics
    - Chat interaction statistics
    - System performance monitoring
8. **Administrative Dashboard**
    - User management (create, update, delete, role assignment)
    - System configuration and settings
    - Audit logging for administrative actions
    - API key management for external access

2.3.2 Out-of-Scope Features

The following features are explicitly excluded from the current project scope but may be considered for future enhancements:

- Mobile native applications (iOS and Android)
- Offline mode functionality
- Video lecture integration and streaming
- Live online classes or video conferencing
- Plagiarism detection for assignments
- Automated assignment grading
- Direct integration with Learning Management Systems (LMS) like Moodle or Canvas
- Payment gateway integration for premium features
- Multi-language support (currently English only)
- Collaborative document editing (Google Docs-style)

2.4 Project Objectives and Goals

2.4.1 Primary Objectives

Objective	**Success Criteria**
Automate Syllabus Processing	Extract structured data from 95% of uploaded syllabi with ≥90% accuracy
Enhance Learning Efficiency	Reduce time spent searching for course information by 60%
Provide Intelligent Assistance	AI chat achieves 85%+ user satisfaction rating for answer relevance
Improve Exam Preparation	Enable access to categorized PYQs covering 80% of syllabus topics

Ensure System Performance	Maintain 99% uptime with <2 second average response time
Support Scalability	Handle 1000+ concurrent users without performance degradation

2.4.2 Secondary Objectives

- Establish a foundation for future AI-enhanced educational tools
- Demonstrate practical application of modern web technologies in education
- Create a reusable platform architecture adaptable to other academic contexts
- Provide insights into student engagement patterns through analytics
- Foster collaborative learning through shared resources

2.5 Stakeholder Analysis

2.5.1 Primary Stakeholders

Students (End Users)

- **Needs:** Easy access to course materials, AI-assisted learning, exam preparation tools
- **Expectations:** Intuitive interface, fast performance, accurate information, reliable AI responses
- **Benefits:** Time savings, improved understanding, better exam preparation, organized study materials

Faculty Members

- **Needs:** Platform for sharing course materials, tracking student engagement
- **Expectations:** Easy content upload, accurate syllabus representation, usage analytics
- **Benefits:** Reduced administrative overhead, insights into student learning patterns

Academic Administrators

- **Needs:** System oversight, user management, performance monitoring
- **Expectations:** Comprehensive admin controls, audit trails, system stability
- **Benefits:** Centralized management, data-driven decision making, automated processes

2.5.2 Secondary Stakeholders

- **University Management:** Interested in improving academic infrastructure and student outcomes
- **IT Department:** Responsible for system deployment, maintenance, and security
- **Cloud Service Providers (DigitalOcean):** Infrastructure and AI service providers
- **Project Development Team:** Developers, designers, and testers building the platform

2.6 Project Deliverables

2.6.1 Software Deliverables

1. **Frontend Application**
    - Next.js 15 + React 19 web application
    - Responsive design for desktop, tablet, mobile
    - Progressive Web App (PWA) capabilities
2. **Backend API Server**
    - Go + Fiber RESTful API (96 endpoints)
    - Docker containerized deployment
    - Comprehensive API documentation
3. **Database Schema**
    - PostgreSQL database with 14 tables
    - Migration scripts and seed data
    - Backup and restore procedures
4. **Deployment Configuration**
    - Docker Compose setup for local development
    - Production deployment scripts
    - Environment configuration templates

2.6.2 Documentation Deliverables

- Technical architecture document
- API reference documentation
- User manual and guides
- Administrator handbook
- Deployment and installation guide
- Source code with inline documentation
- This project report

2.6.3 Testing Deliverables

- Test plan and test cases
- Unit test suite with coverage reports
- Integration test scenarios
- Performance test results
- Security audit report