# 1. INTRODUCTION

## 1.1 About the Project

**Study in Woods** is an innovative AI-powered educational platform designed to transform the way students interact with their academic materials. In today's digital age, students face significant challenges in organizing course content, understanding complex syllabi, and accessing relevant study materials efficiently. This project addresses these challenges by providing an intelligent study companion that leverages artificial intelligence to enhance the learning experience.

The platform serves as a comprehensive academic management system that integrates modern web technologies with advanced AI capabilities to create a seamless educational ecosystem. It enables students to upload course syllabi in PDF format, which are then automatically processed using state-of-the-art natural language processing to extract structured information including course units, topics, subtopics, and learning outcomes.

Furthermore, the system incorporates an intelligent chat interface powered by large language models, allowing students to engage in natural conversations about their course materials. This conversational AI has contextual awareness of uploaded documents and can provide detailed explanations, answer questions, and offer study guidance tailored to specific subjects and topics.

## 1.2 Purpose of the Project

The primary objectives of Study in Woods are:

- **Academic Organization:** To provide students with a centralized platform for organizing their university courses, semesters, and subjects in a hierarchical structure that mirrors real academic institutions.
- **Intelligent Content Extraction:** To automatically process and extract structured information from syllabus PDFs using AI, eliminating the need for manual data entry and ensuring accuracy in course content representation.
- **AI-Assisted Learning:** To offer students an intelligent study companion that can answer questions, provide explanations, and offer study guidance based on their specific course materials and uploaded documents.

- **Exam Preparation Support:** To help students prepare for examinations by extracting and categorizing questions from past year question papers, linking them to relevant syllabus topics for targeted revision.
- **Progress Tracking:** To enable students to track their study progress, create personalized study plans, and manage academic todos effectively.
- **Multi-University Scalability:** To support multiple educational institutions, courses, and programs, making the platform adaptable to diverse academic environments.
- **Real-Time Collaboration:** To facilitate knowledge sharing and collaborative learning among students through shared resources and study materials.

## 1.3 Project and Product Overview

**Study in Woods** is architected as a full-stack web application following modern software engineering practices. The system employs a microservices-oriented architecture with clearly separated frontend and backend components, ensuring maintainability, scalability, and optimal performance.

### *1.3.1 System Architecture*

The application follows a three-tier architecture comprising:

- **Presentation Layer (Frontend):** Built using Next.js 15 with React 19, providing a responsive and intuitive user interface accessible across all devices.
- **Application Layer (Backend):** Developed in Go (Golang) using the Fiber framework, handling business logic, API endpoints, and third-party integrations.
- **Data Layer:** PostgreSQL serves as the primary relational database, with Redis providing high-performance caching and session management.

### *1.3.2 Core Functionalities*

The platform delivers the following key functionalities:

1. **User Management:** Secure authentication and authorization system with role-based access control (Admin and Student roles)
2. **Academic Hierarchy Management:** Create and manage Universities, Courses, Semesters, and Subjects
3. **Document Processing:** Upload syllabi and study materials with automatic AI-powered content extraction
4. **AI Chat Interface:** Interactive conversational AI providing subject-specific assistance and study guidance
5. **Past Year Questions (PYQ) Management:** Extract, categorize, and organize previous examination questions
6. **Analytics Dashboard:** Track usage statistics, student engagement, and system performance

7. **API Access:** RESTful API with 96 endpoints for programmatic access and external integrations

## 1.4 Overall Description of the Project

### *1.4.1 Product Perspective*

Study in Woods operates as a standalone web-based application while maintaining integration capabilities with external AI services and cloud infrastructure. The system is designed to be:

- **Self-Contained:** All core functionalities operate independently without mandatory external dependencies
- **Cloud-Native:** Deployed on DigitalOcean cloud infrastructure with containerized services for scalability
- **API-First:** RESTful architecture enables integration with mobile applications and third-party services
- **AI-Enhanced:** Leverages DigitalOcean's GradientAI platform (Llama 3.3 70B model) for intelligent features

### *1.4.2 User Characteristics*

The system caters to two primary user categories:

**Students (Primary Users):**

- Enrolled in Master of Computer Applications (MCA) or similar programs
- Possess basic computer literacy and web browsing skills
- Require tools for organizing academic content and exam preparation
- Seek AI-assisted learning and study guidance

**Administrators (System Managers):**

- Academic staff or system administrators
- Responsible for managing universities, courses, and system settings
- Monitor system usage, user activity, and performance metrics
- Require advanced access controls and audit capabilities

### *1.4.3 Operating Environment*

The application operates in the following environment:

- **Client-Side:** Modern web browsers (Chrome, Firefox, Safari, Edge) on desktop, tablet, and mobile devices
- **Server-Side:** Containerized deployment using Docker on Linux-based systems
- **Database:** PostgreSQL 15 for relational data storage
- **Caching:** Redis 7 for high-speed data caching and session management
- **Cloud Services:** DigitalOcean Spaces for file storage and GradientAI for AI capabilities
- **Network:** HTTPS-enabled secure communication for all client-server interactions

### *1.4.4 Design and Implementation Constraints*

- **Technology Stack:** Backend must be implemented in Go (Golang) for performance; frontend must use Next.js with React
- **Database:** PostgreSQL required for ACID compliance and relational data integrity
- **Security:** Must comply with industry-standard security practices including JWT authentication, password hashing, and encrypted API keys
- **AI Integration:** Limited to DigitalOcean GradientAI platform for cost-effectiveness and regional availability
- **File Processing:** PDF format only for document uploads, with maximum file size constraints
- **Scalability:** System must support minimum 1000 concurrent users with horizontal scaling capabilities

### *1.4.5 Assumptions and Dependencies*

**Assumptions:**

- Users have stable internet connectivity for accessing cloud-based services
- Uploaded syllabi are in readable PDF format (not scanned images without OCR)
- Students have valid institutional email addresses for registration
- Administrative personnel have technical competency for system management

**Dependencies:**

- **DigitalOcean Services:** For AI processing and cloud file storage
- **PostgreSQL Database:** For persistent data storage
- **Redis Cache:** For session management and performance optimization
- **Third-Party Libraries:** Go modules and npm packages as specified in dependency manifests