## 1.1. Subject Description

The project consists of designing and developing an **AI-powered online learning platform**. The platform will allow instructors to create and manage courses, students to enroll and track progress, and administrators to supervise the system.

The theoretical and technical notions to be studied include:

- **Theoretical Aspects**:
    - Principles of online learning platforms (LMS – Learning Management Systems).
    - Role-based access control (RBAC) and authentication models.
    - Recommendation system basics (content-based and collaborative filtering).
    - Chatbot–AI interaction in education (AI tutoring systems).
- **Technical Aspects (Documentation)**:
    - Next.js framework for full-stack web development.
    - MongoDB database management with Mongoose ODM.
    - NextAuth.js for authentication and session management.
    - Gemini API integration for natural language AI chatbot.
    - TailwindCSS/Bootstrap for responsive UI.

---

## 1.2. Needs Analysis and Specification

The work includes analyzing requirements and specifying the solution to be developed:

- **Study of the existing systems**: Reviewing popular LMS platforms (e.g., Coursera, Udemy, Moodle) to identify features, strengths, and limitations.
- **Formalism Used**:
    - UML diagrams (use case diagrams, class diagrams).

### 1.2.1. Functional Requirements

- **Authentication & Roles**:
    - Secure login/registration (email/password or Google login).
    - Roles: Student, Instructor, Admin.
- **Student Features**:
    - Browse and search for courses.
    - Enroll in courses and track progress.
    - Ask questions to the AI chatbot.
    - Receive course recommendations.
- **Instructor Features**:
    - Create and manage courses (title, description, lessons).
    - Upload PDF documents and video links.
    - Track student engagement.
- **Admin Features**:
    - Manage users (activate/deactivate).
    - Approve or reject instructor courses.
    - Monitor global platform activity.
- **AI Features**:
    - Chatbot tutor to answer student questions using Gemini API.
    - Recommendation system to suggest relevant courses.

### 1.2.2. Non-Functional Requirements

- **Performance**: Fast loading and responsive interface.
- **Scalability**: Ability to support many users and courses simultaneously.
- **Security**: Data protection, secure authentication, and role-based access.

- **Usability**: User-friendly UI, easy navigation for students and instructors.
- **Maintainability**: Modular and well-documented codebase.
- **Compatibility**: Support across browsers and mobile devices.

# 2. Working Environment

## 2.1. Software

- **Programming Languages**:
    - JavaScript, TypeScript (optional).
- **Frameworks & Libraries**:
    - Next.js (frontend + backend with API routes).
    - Mongoose (MongoDB integration).
    - NextAuth.js (authentication).
    - TailwindCSS / Bootstrap (styling).
    - Gemini API (AI chatbot).
- **Development Tools**:
    - Visual Studio Code (IDE).
    - Git/GitHub (version control).
    - Postman (API testing).
    - MongoDB Compass (database management).
- **Other Tools**:
    - UML diagram tools (Draw.io).