

AI Study Assistant - Complete Project Specification

1. Project Overview

A full-stack AI-powered learning platform that combines tutoring, document intelligence, analytics, and course management into a unified ecosystem.

2. Core Features

- Conversational AI Tutor (Socratic method)
- Document-based Q&A; using RAG
- Smart summarization
- Quiz and flashcard generation
- Study session tracking
- Performance analytics dashboard

3. User Roles

Student:

- Access AI tutor
- Upload resources
- Track study progress

Instructor:

- Manage courses
- Monitor student analytics
- Customize AI behavior

Admin:

- Manage system and users

4. Tech Stack

Frontend:

- React (Vite)
- Tailwind CSS

Backend:

- Node.js + Express

Database:

- Microsoft SQL Server
- ORM: Sequelize / TypeORM

AI:

- OpenAI API / Claude
- LangChain for RAG

5. System Architecture

- Client (React UI)
- REST API (Express)
- AI Service Layer
- SQL Server Database
- File Storage (cloud/local)

6. Database Design (Core Tables)

Users (id, name, email, role)

Courses (id, title, instructor_id)

Enrollments (user_id, course_id)

Resources (id, course_id, file_path)

Sessions (user_id, duration, timestamp)

7. API Endpoints (Examples)

POST /auth/register

POST /auth/login

GET /courses

POST /resources/upload

POST /ai/chat

GET /analytics

8. Development Roadmap

Phase 1: Setup environment

Phase 2: Backend & DB

Phase 3: Frontend UI

Phase 4: AI Integration

Phase 5: Advanced features

Phase 6: Testing & deployment

9. Security

- JWT Authentication
- Password hashing (bcrypt)
- CORS & Helmet
- Role-based access control

10. Deployment

Frontend: Vercel / Netlify

Backend: Render / Azure

Database: Azure SQL Server

11. Challenges & Solutions

- ORM complexity → start small
- Performance → pagination & caching
- AI hallucination → RAG + citations

12. Future Enhancements

- Voice-based tutor
- Mobile app
- Real-time collaboration
- AI personalization improvements