AI Study Companion

# Introduction

Students often struggle to condense lecture materials into digestible learning aids. Traditional flashcards and summaries are effective but creating them manually is time-consuming. An AI powered study companion could automate this process by turning lecture notes into summaries, flashcards, and quizzes.

This project aims to design a tool that supports self-directed learning using open-source NLP models — avoiding external API costs. The main challenge is engineering the pipeline, not inventing new AI.

# Positioning

## Problem Statement

|  |  |
| --- | --- |
| The problem of | Manually creating study materials from lecture content |
| affects | Students who wish to learn from large numbers of course materials efficiently. |
| the impact of which is | Students spend extra time manually preparing study resources instead of learning; they usually lack consistency and motivation, resulting in poor exam performance. |
| a successful solution would be | a tool that automatically generates summaries, flashcards and quiz questions from the uploaded lecture material, helping students track their progress and save time. |

## Product Position Statement

|  |  |
| --- | --- |
| For | University students engaged in self-learning |
| Who | Need a faster and efficient way to convert lecture materials into learning aids |
| The AI Study Companion | is an AI-powered web application |
| That | Automatically summarizes lecture contents and generates flashcards and quizzes for efficient studying |
| Unlike | Traditional study methods or commercial AI tools |
| Our product | Uses open-source NLP models to provide free and customizable study support while tracking user progress. |

# Stakeholder Descriptions

## Stakeholder Summary

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Students | Primary users of the system | Upload lecture materials, use generated learning aids, provide feedback on usability. |

## User Environment

Number of users: Initially small (student testers), can be scaled to university-wide deployment.

Usage context: Students use it via a web interface, often on laptops, in study environments like libraries or at home.

Task cycle:

1. Upload lecture material (PDF, DOCX, plain text)
2. Wait for the automatic generation of contents (flashcards, quiz, summary)
3. Study using the generated contents
4. Track progress over multiple sessions

Platforms: Web (React or similar frontend), backend with Python, NLP models using open-source frameworks, database (postgresql, ChromaDB for AI)

Integrations: Possible integration with student login systems or cloud storage (Google Drive or OneDrive) in future releases.

Constraints:

* Must run on open source or free infrastructure
* Must preserve data privacy (no external API calls)
* Should support moderate documents (around 30 pages)

# Product Overview

## Needs and Features

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Upload lecture materials | High | File Upload | V1.0 |
| Text preprocessing | High | Text cleaning, segmentation | V1.0 |
| Generate summary | High | NLP-based summarization using open-source models | V1.0 |
| Generate flashcards | High | Automated key-term extraction | V1.0 |
| Generate quiz | High | Multiple-choice question generation | V1.0 |
| Interactive mode | Medium | Flashcard/Quiz interface with scoring | V1.1 |
| Tracking progress | Medium | Store history, completion rates, performance stats | V1.1 |
| User accounts | Medium | Simple authentication, session saving | V1.2 |
| Usability and accessibility | High | Clear UI | All |
| Scalability and maintainability | Medium | Modular architecture, open-source models | Continuous |

# Other Product Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
| Platform: Web-based | High | V1.0 |
| Performance: Should process a 10-page lecture note within 30 seconds | Medium | V1.1 |
| Robustness: Must handle corrupted uploads gracefully | Medium | V1.0 |
| Usability: Simple, minimal UI suitable for students with no training. | High | V1.0 |
| Security: Basic user authentication, local data storage | High | V1.2 |
| Documentation: User manual and developer documentation | High | Final release |
| Assumptions: Access to pre-trained open-source NLP models | High | Ongoing |
| Dependencies: Python 3.x, Web framework | High | V1.0 |