

Smart Study Partner Project

- ▶ Subtitle: Interactive AI-powered Study Helper
- ▶ Name: SAIJASI SINHA
- ▶ Registration No: 25BSA10021
- ▶ Course: Introduction to AI & ML

Introduction

- ▶ * Smart Study Partner is a lightweight AI-powered tool that helps students learn topics efficiently.
- ▶ * Uses Python and ML concepts to suggest topics, provide summaries, and conduct mini quizzes.
- ▶ * Optional web interface for interactive study sessions.

Problem Statement

- ▶ * Students often struggle to revise topics effectively.
- ▶ * Need for an intelligent study companion that can suggest relevant content and assess understanding.

Functional Requirements

- ▶ * Accept topic input from user.
- ▶ * Suggest the closest topic using ML (TF-IDF + Cosine Similarity)
- ▶ .* Provide quick summary and key questions.
- ▶ * Conduct a mini quiz and check answers.
- ▶ * Optional web interface for interaction.

Non-functional Requirements

- ▶ * Lightweight and fast.
- ▶ * User-friendly interface.
- ▶ * Extendable for adding more topics.
- ▶ * Platform-independent (runs on Windows, Linux, Mac).

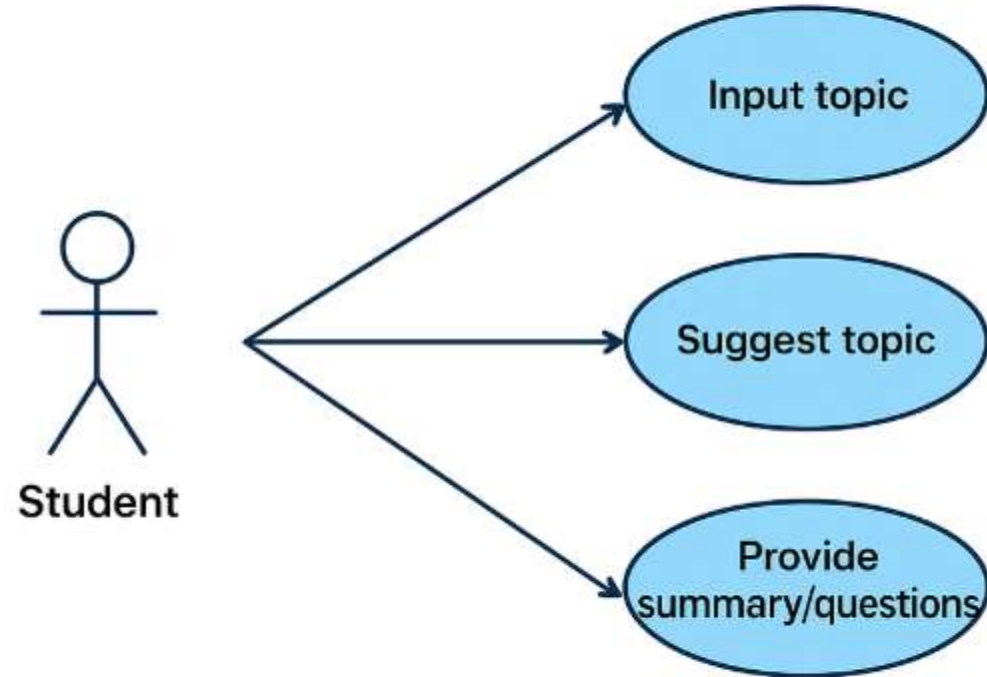
System Architecture*

- ▶ User Input
- ▶ TF-IDF Vectorization
- ▶ Cosine Similarity
- ▶ Topic Selection
- ▶ Display Summary, Questions, Quiz
- ▶ Feedback:- Optional: Web Front-End (Flask) connects to Python backend.

: Design Diagrams

- ▶ Use Case Diagram: Student inputs topic
- ▶ System suggests topic
- ▶ System provides summary/questions
- ▶ Student answers quiz.* Workflow Diagram: Input
- ▶ Processing
- ▶ Output
- ▶ Feedback:- Sequence Diagram:
 - ▶ User System -> Database/Knowledge Base -> Response.*
- ▶ Class/Component Diagram: Classes: Topic, Quiz, UserInterface, BackendProcessor.
- ▶ * ER Diagram: Not applicable (no persistent storage used).

Use Case Diagram



Design Decisions & Rationale

- ▶ * Chose TF-IDF for simplicity and effectiveness with small datasets.
- ▶ * Flask used for lightweight web front-end.
- ▶ * Emoji and humanized messages to increase engagement.
- ▶ * Modular code structure for easy updates and future extensions.

Implementation Details

- ▶ * Python-based backend using scikit-learn for TF-IDF and similarity.
- ▶ * Knowledge base stored as dictionaries.
- ▶ * Terminal-based humanized interaction.
- ▶ * Optional web interface with Flask, HTML, and CSS.

Screenshots / Results

- ▶ * Screenshot 1

```
Here's a quick summary for you:  
Plants make food from sunlight . They have this green stuff called chlo  
rophyll that helps. Think of it like a mini solar kitchen in the leaves  
! 🌱
```

- ▶ : Terminal interaction showing topic summary.

- ▶ * Screenshot 2:

```
Q: Where does photosynthesis mostly happen?  
Your answer: leaves  
Bingo! You got it!
```

- ▶ Mini quiz in terminal.

- ▶ * Screenshot 3:

```
All done! Your score: 1/1  
Keep learning, stay curious, and go drink some water 💧
```

- ▶ Web interface displaying topic summary and questions.

Testing Approach

- ▶ Manual testing with various topic inputs.
- ▶ Verified correct topic matching, summaries, and quiz answers.
- ▶ Checked web interface for usability and correct display of content.

Challenges Faced

- ▶ Limited dataset of topics.
- ▶ Matching user input accurately to correct topic.
- ▶ Integrating terminal-based interaction with web interface.
- ▶ Handling user input variations and typos.---

Learnings & Key Takeaways

- ▶ Understanding of TF-IDF and cosine similarity.
- ▶ Learned how to create a modular AI/ML-based Python application.
- ▶ Basics of Flask and web front-end integration.
- ▶ Importance of user-friendly interaction design.

Future Enhancements

- ▶ Add more topics dynamically.
- ▶ Store quizzes and scores for each user.
- ▶ Use advanced NLP models (like BERT) for better topic matching.
- ▶ Add text-to-speech and speech input for accessibility.
- ▶ Full web/mobile application with database support.

References

- ▶ scikit-learn documentation: [<https://scikit-learn.org/>] (<https://scikit-learn.org/>)
- ▶ Flask documentation: [<https://flask.palletsprojects.com/>] (<https://flask.palletsprojects.com/>)
- ▶ Python official docs: [<https://docs.python.org/>] (<https://docs.python.org/>)
- ▶ TF-IDF concept: [<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>] (<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>)