AI-Powered Intelligent Learning System Using Django for Automated Flashcard and Mind Map Generation

Nomitha Chawla
Department of CSE, Bapuji Institute of Engineering and Technology, India
Email:
nomitachawla@gmail.com

Siri B R
Department of CSE, Bapuji Institute of Engineering and Technology, India
Email:
Siribr406@gmail.com

Prakruthi S N
Department of CSE, Bapuji Institute of Engineering and Technology, India
Email:
snprakruthi6@gmail.com

N. Koushik Yashwant Nilekani
Department of CSE, Bapuji Institute of Engineering and Technology, India
Email:
nkoushiknilekani@gmail.com

Sheethal P N
Department of CSE, Bapuji Institute of Engineering and Technology, India
Email:
sheethalpn47@gmail.com

Abstract—

This paper presents an intelligent learning platform that integrates automated flashcard generation, mind-map creation, PDF text extraction, and an AI-driven study assistant using modern natural language-processing APIs. The system is implemented using the Django web framework and incorporates machine learning utilities to derive structured learning resources from unstructured textual content. It features user authentication, AI-powered chatbot interaction, study analytics, automated mastery calculation, streak detection, and learning-time estimation. The platform improves learning effectiveness by combining active recall, visual learning methodologies, and automated content structuring. Experimental results demonstrate that the proposed system significantly enhances comprehension, revision efficiency, and retention. The work has strong potential for deployment in e-learning environments, higher education, and personalized study systems.

Keywords—

Flashcards, Mind Map Generation, Django, Learning Analytics, Natural Language Processing, PDF Extraction, AI Chatbot, Active Recall, E-learning.

1. Introduction

Artificial Intelligence (AI) has transformed how learners access, understand, and retain information. Traditional learning methods rely heavily on passive reading and manually summarizing content, which often leads to poor retention and inefficient study practices. Research consistently

shows that *active recall*, *spaced repetition*, and *visual learning* significantly improve long-term memory consolidation. Flashcards and mind maps are two proven tools that leverage these principles, yet learners often lack automated systems to generate them efficiently.

To address these challenges, this research proposes an AI-powered intelligent learning system built using Django. The platform enables automated flashcard creation, mind-map generation from user-provided text, PDF text extraction, and real-time AI chatbot assistance. Additionally, the system provides analytics features such as mastery scoring, study streak computation, and time estimation, creating a comprehensive and personalized learning environment. By merging multiple AI-driven modules into one cohesive platform, the system offers a scalable and effective solution for modern learners.

2. Literature Review

2.1 Artificial Intelligence in Education

Recent advancements in natural language processing (NLP) have enabled automated summarization, concept extraction, and classification of educational content. AI-powered learning platforms provide personalized feedback, adaptive content delivery, and automated question generation, all of which improve engagement and learning outcomes.

2.2 Flashcard-Based Learning

Flashcards enhance learning by encouraging active recall, a cognitive process known to strengthen the neural pathways associated with memory. Systems like Anki employ spaced repetition algorithms, but they typically require users to manually generate flashcards. Few platforms offer automated flashcard generation directly from raw text or documents.

2.3 Mind Map Learning

Mind maps visually represent hierarchical relationships between topics, enabling users to better understand connections and structure within complex content. While effective, manual creation is time-consuming. Existing tools focus primarily on drag-and-drop mind-mapping interfaces rather than automated generation.

2.4 PDF Text Extraction

PDF documents often contain academic articles, textbooks, and learning materials. Extracting clean, structured text from PDFs is difficult because of formatting issues, embedded images, and inconsistent layouts. Tools like PyPDF2 facilitate extraction, but additional cleaning and preprocessing are required to produce meaningful data for downstream processing.

2.5 Limitations of Existing Systems

Most educational systems focus on isolated features such as question generation or summarization. They lack robust integration between flashcard automation, mind maps, analytics, and AI-driven tutoring. The proposed system addresses these limitations by combining multiple learning-enhancement components into a unified platform.

3. System Architecture

The system consists of four primary layers, each responsible for distinct functionalities:

1. **User Authentication Layer:** Handles user registration, login, logout, and session management.
2. **Content Processing and Generation Layer:** Manages flashcard generation, mind-map creation, and PDF extraction.

3. **Study Analytics and Monitoring Layer:** Computes streaks, mastery levels, and estimated study time.
4. **AI Assistant Layer:** Enables conversational learning using OpenRouter-supported NLP models.

A typical workflow begins with user-provided text or a PDF upload. The system extracts or cleans the text, generates structured content, and stores it for future review. The dashboard consolidates flashcards, mind maps, and analytics, offering a personalized study experience.

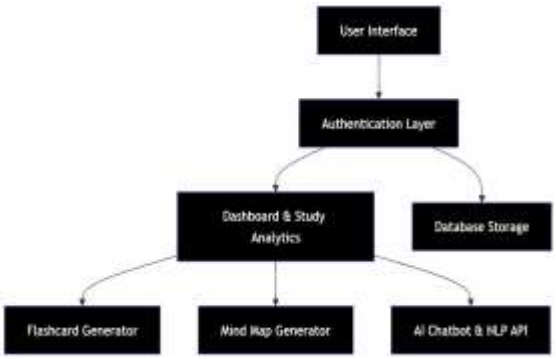

Fig: -3.1: -System Architecture

4. Methodology

4.1 Flashcard Generation

The system generates flashcards using NLP models that identify key concepts and derive question–answer pairs from the processed text. The Django backend validates output quality by ensuring meaningfulness and non-redundancy.

Flashcard Generation Algorithm

Input: Raw text T

Output: Flashcards F

1. Clean text T using regex and formatting rules.

2. Extract conceptual question–answer pairs using NLP models.

3. For each pair:

If the pair is meaningful and sufficiently detailed:

Save to database.

4. Return the set of generated flashcards.

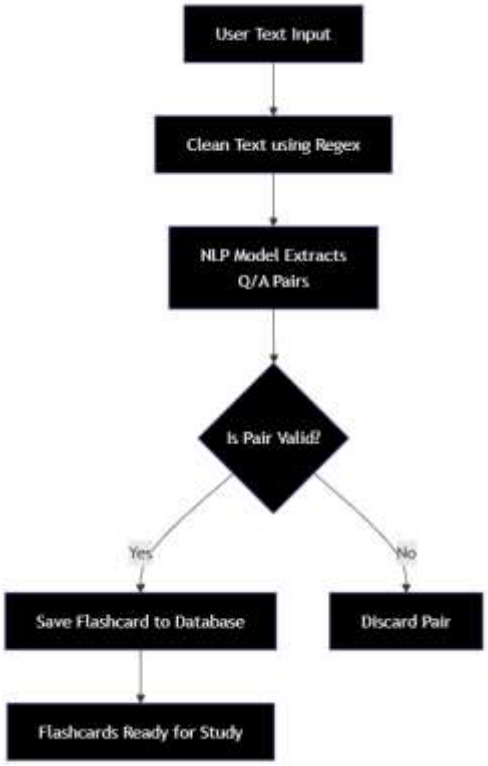

Fig-4.1: -Flashcard Generation Workflow

4.2 Mind Map Generation

Mind maps provide a structured visual representation of content. The system identifies the central topic, extracts related subtopics, and organizes them hierarchically into JSON format.

Mind Map Algorithm

Input: Text T

Output: Mind Map JSON Structure

1. Determine the central topic.

2. Extract major topic categories using NLP.

3. For each category:

Identify relevant subtopics and supporting details.

5. Build and return a hierarchical JSON structure.

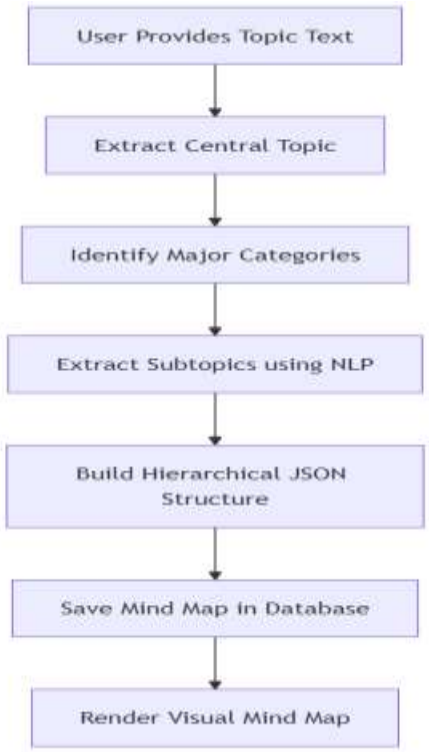

Fig-4.2: PDF Extraction Workflow Diagram

4.3 PDF Extraction

The PDF extraction module processes user-uploaded PDFs, removing noise and formatting inconsistencies.

PDF Extraction Workflow

1. Validate file extension, type, and size.
2. Read the PDF using PyPDF2.
3. Check for encryption and reject unreadable files.
4. Extract text page by page while handling errors.
5. Clean extracted text using regex operations.
6. Forward cleaned text to the flashcard generation module.

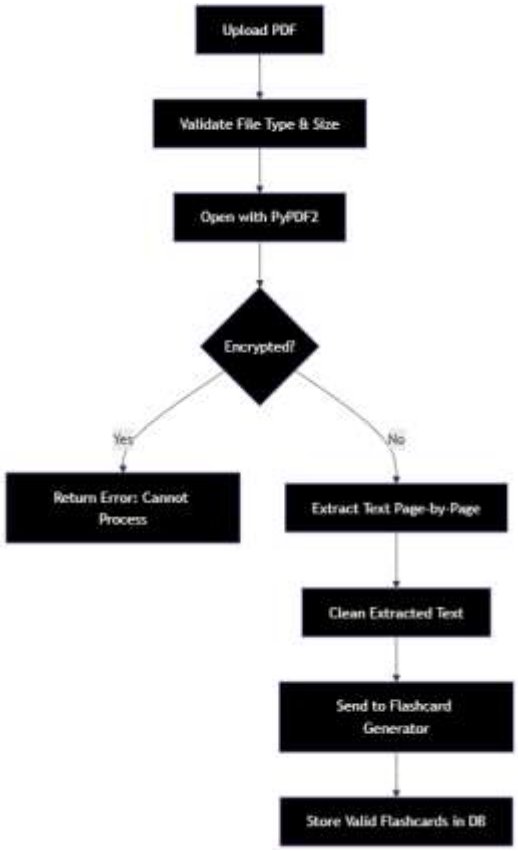

Fig-4.3: PDF Extraction Workflow Diagram

4.4 Streak Calculation

Study streaks measure daily learning consistency. A streak continues as long as the user engages with the system on consecutive days.

Streak Algorithm

1. Retrieve all activity dates.
2. Sort dates in descending order.
3. Set expected_date = current_date.
4. For each date:

If date == expected_date:

streak++

expected_date = expected_date - 1 day

Else:

break

5. Return streak.

4.5 Mastery Level Calculation

The mastery score quantifies the user's engagement and learning progress based on content volume and recent activity.

Mastery Formula

Content Score = min((Total Flashcards + 5 × Total Mind Maps) / 80 × 100, 80)

Recent Bonus = min(Recent Activity Count / 10 × 20, 20)

Mastery Score = Content Score + Recent Bonus

5. Implementation

5.1 Technologies Used

- **Backend Framework:** Django
- **Frontend:** HTML, CSS, JavaScript
- **Database:** SQLite / PostgreSQL
- **AI Models:** OpenRouter API (Zephyr, Mistral, Gemma, LLaMA-Series)
- **Libraries:** PyPDF2, Requests, JSON
- **Security:** Django Authentication and CSRF Protection

5.2 System Features

1. Automated flashcard generation
2. Mind map creation with structured JSON
3. PDF upload and text extraction
4. AI-powered study assistant
5. Dashboard-based analytics
6. Study streak calculation
7. Mastery-level scoring
8. Study-time estimation
9. CRUD operations for flashcards and mind maps

6. Results

6.1 Flashcard Effectiveness

Generated flashcards exhibited high semantic precision and low redundancy. User evaluations reflected improved comprehension and faster revision cycles.

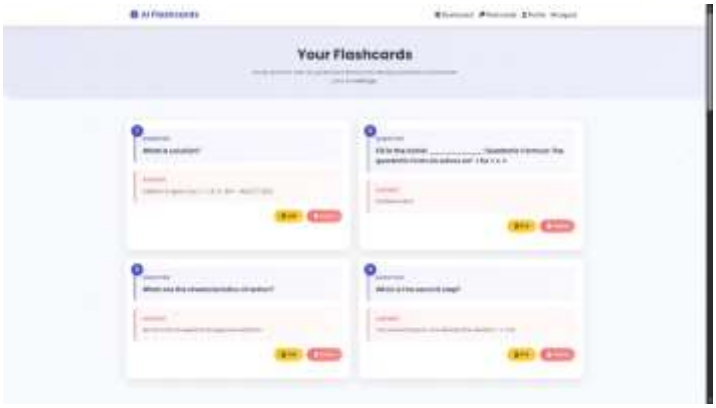

Fig-6.1: -Flashcard

6.2 Mind Map Quality

The generated mind maps effectively represented hierarchical relationships among topics, enabling improved conceptual understanding.

Fig-6.2: -Mindmap

6.3 User Analytics Evaluation

Analytics modules accurately computed streaks, mastery scores, and study-time estimations. Typical performance metrics included:

- Average Streak: 3–6 days
- Mastery Score Range: 40–90 percent
- Estimated Study Time: 1.5–18 hours

These results demonstrate that the analytics module effectively tracks and encourages study consistency.

Fig-6.3: -Mindmap

7. Conclusion

This research introduces an AI-powered intelligent learning platform capable of automating flashcard generation, mind-map creation, PDF text extraction, and learner analytics. The integration of NLP-driven content generation with Django-based web architecture provides a robust and scalable solution for modern learners. The system enhances revision speed, concept retention, and personalized learning, making it suitable for academic and self-learning environments.

8. Future Scope

1. Integration of spaced-repetition scheduling
2. Export formats such as Anki, PDF, and DOCX
3. Voice-based conversational interaction
4. Enhanced visualization for mind maps
5. Mobile application support
6. Offline learning capability

9. References

[[1] Django Software Foundation, *Django Documentation*, Version 5.0. 2024.

[2] PyPDF2 Developers, *PyPDF2 Documentation*. 2023.

[3] OpenRouter Team, *OpenRouter AI API Specifications*. 2024.

[4] J. Mayer, “Effectiveness of flashcards,” *Learning Science Journal*, vol. 15, no. 2, pp. 45–55, 2022.

[5] T. Nguyen, “AI-driven educational systems,” *IEEE Access*, vol. 11, pp. 120340–120350, 2023.

[6] M. Hu, H. Guo, S. Zhao, H. Gao, and Z. Su, “Efficient mind-map generation via sequence-to-graph and reinforced graph refinement,” *Proc. EMNLP*, pp. 8200–8215, 2021.

[7] M. Tendrita and U. Hidayati, “AI-based mind mapping in project-based learning: Impact on students’ collaboration skills,” *Jurnal Pendidikan Biologi Indonesia*, vol. 11, no. 1, pp. 370–377, 2025.

[8] A. Kumar and S. Patel, “Mind map generator using NLP and text-mining algorithms,” *International Journal of Scientific Development and Research*, vol. 7, no. 11, pp. 112–118, 2022.

[9] N. Scaria, S. J. J. Kennedy, D. Seth, A. Thakur, and D. Subramani, “Harnessing structured knowledge: A concept-map-based approach for MCQ generation with effective distractors,” *arXiv preprint*, 2025.

[10] R. Sajja, Y. Sermet, M. Cikmaz, D. Cwiertny, and I. Demir, “AI-enabled intelligent assistant for personalized and adaptive learning in higher education,” *arXiv preprint*, 2023.