A PROJECT REPORT

On

"SMART EDUCATION PLATFORM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

NIRANJAN SAH	21052944
SUMIT CHAUDHARY	21052952
NIKESH KUMAR MANDAL	21053299
ABHI UPADHYAY	21053401
LOVE KUMAR SAH	21053435
SUMAN SHARMA	21053443

UNDER THE GUIDANCE OF
Prof. Dr. HRUDAYA KUMAR TRIPATHY

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2025

A PROJECT REPORT

On

"SMART EDUCATION PLATFORM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

BY

NIRANJAN SAH	21052944
SUMIT CHAUDHARY	21052952
NIKESH KUMAR MANDAL	21053299
ABHI UPADHYAY	21053401
LOVE KUMAR SAH	21053435
SUMAN SHARMA	21053443

UNDER THE GUIDANCE OF

Prof. Dr. HRUDAYA KUMAR TRIPATHY

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"SMART EDUCATION PLATFORM"

Submitted by

NIRANJAN SAH	21052944
SUMIT CHAUDHARY	21052952
NIKESH KUMAR MANDAL	21053299
ABHI UPADHYAY	21053401
LOVE KUMAR SAH	21053435
SUMAN SHARMA	21053443

Is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 15/04/2025

(Prof. Dr. Hrudaya Kumar Tripathy)
Project Guide

Acknowledgements

We are profoundly grateful to **Prof. Dr. Hrudaya Kumar Tripathy** of **B.Tech Computer Science and Engineering** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

NIRANJAN SAH
SUMIT CHAUDHARY
NIKESH KUMAR MANDAL
ABHI UPADHYAY
LOVE KUMAR SAH
SUMAN SHARMA

ABSTRACT

The Smart Education platform is an interactive web application built with Streamlit that leverages AI to enhance learning experiences through multimedia content analysis. The system offers three primary functionalities: video comparison, note generation, and PDF-based question answering. It extracts transcripts from YouTube videos using the YouTube Transcript API and processes them through GPT model to generate comprehensive summaries, structured notes, and personalized responses to user queries. The platform enables users to compare multiple videos for relevance to their learning goals, create downloadable study materials in PDF or DOCX formats, and engage in contextual Question and Anwering sessions with both video and document content.

The application's architecture incorporates several specialized components, including transcript extraction, content generation, document processing, and intelligent recommendation systems. For video content, it can identify key themes such as tutorials, reviews, and how-to guides, providing users with tailored recommendations based on their learning objectives. The PDF functionality allows users to upload documents and ask specific questions, receiving AI-generated answers based on the document's content. With its clean user interface and practical output options, Smart Education serves as a versatile educational tool that transforms passive content consumption into active, personalized learning experiences that can be saved and referenced across multiple formats.

Keywords: AI-powered learning, Video Transcript Analysis, PDF Question Answering, Educational Content Summarization, Interactive Note Generation.

Contents

List of Figures

Chapter 1

Introduction

The Smart Education platform is an interactive web application built with Streamlit that leverages artificial intelligence to enhance learning experiences through multimedia content analysis. The system offers three primary functionalities: video comparison, note generation, and PDF-based question answering. It extracts transcripts from YouTube videos using the YouTube Transcript API and processes them through GPT model to generate comprehensive summaries, structured notes, and personalized responses to user queries. The platform enables users to compare multiple videos for relevance to their learning goals, create downloadable study materials in PDF or DOCX formats, and engage in contextual Q&A sessions with both video and document content.

In recent years, there has been a significant increase in the adoption of smart technology in education, with AI playing a crucial role in transforming traditional educational spaces into interactive learning environments. This project addresses the growing need for personalized learning experiences, improved content analysis, and efficient knowledge extraction from multimedia sources. By integrating advanced AI capabilities with user-friendly interfaces, the Smart Education platform aims to bridge the gap between passive content consumption and active, personalized learning.

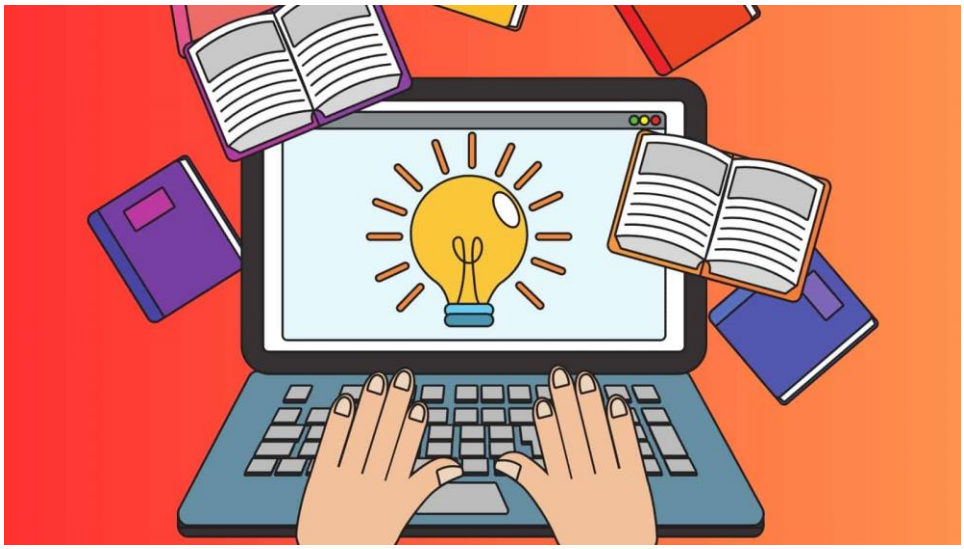

Figure 1 : Smart Education

Chapter 2

Literature Review

2.1 Smart Education Framework

Smart education represents an evolving educational paradigm that integrates technology, pedagogy, and personalized learning approaches. According to research by Zhu et al. (2016), smart education aims "to improve learners' quality of lifelong learning" by focusing on "contextual, personalized and seamless learning to promote learners' emerging intelligence and facilitate their problem-solving ability in smart environments." This framework typically encompasses three key components:

- Interwoven learning environments: The integration of physical and digital spaces to create ubiquitous learning opportunities (Kim et al., 2019).
- Adaptive learning approaches: Educational methods that emphasize collaboration, cross-disciplinary learning, and practical application (Hwang, 2014).
- Flexible organizational structures: Administrative systems that leverage technology to enhance educational governance and institutional effectiveness (Spector, 2016).

Research indicates that effective smart education implementations require a balanced approach that considers technological capabilities alongside pedagogical objectives and learner needs.

2.2 AI in Education

The application of Artificial Intelligence in education has expanded significantly in recent years. Chassignol et al. (2018) identify several key applications of AI that are transforming educational practices:

- Personalized learning systems: AI algorithms that analyze learner data to create customized learning pathways.
- Predictive analytics: Machine learning models that identify at-risk students based on performance patterns.
- Automated assessment tools: Systems that evaluate assignments and provide feedback with minimal human intervention.
- Administrative automation: AI-powered solutions for scheduling, resource allocation, and other administrative tasks.
- Intelligent tutoring systems: Adaptive platforms that provide personalized guidance based on individual learning patterns.

While research by Holmes et al. (2019) suggests that AI can reduce teacher workload and improve student outcomes across multiple skill domains, challenges remain regarding ethical implementation, cost barriers, and professional development needs for educators.

2.3 Multimedia Content Analysis in Education

The analysis of multimedia educational content involves sophisticated techniques for extracting meaningful information from video, audio, and text sources. Researchers have developed various approaches to this challenge:

- Multimodal content processing: Methods for analyzing both visual and verbal elements in educational videos (Basu et al., 2021).
- Semantic content extraction: Techniques for identifying key concepts and relationships in educational materials (Liu & Kender, 2017).
- Discourse analysis: Approaches for understanding the structure and meaning of educational communications (Dillenbourg & Jermann, 2010).

As noted by Dimitriadis et al. (2021), media content analysis provides researchers with tools to systematically examine how educational discourse shapes public understanding and policy development, offering insights into the relationship between global educational trends and national implementation strategies.

2.4 Smart Learning Environments

Smart learning environments represent integrated educational ecosystems that combine physical spaces, digital technologies, and adaptive pedagogies. According to Hwang (2014), these environments are characterized by:

- Context-awareness: The ability to sense and respond to learner contexts.
- Adaptivity: Capabilities for customizing learning resources based on learner needs.
- Ubiquity: Seamless access to learning resources across different settings.
- Data-driven decision making: The use of learning analytics to inform educational practices.
- Natural interaction: Support for multimodal communication between learners and systems.
- Engagement: Immersive experiences that promote active participation.

Research by Spector (2016) suggests that effective smart learning environments prioritize self-directed learning, intrinsic motivation, and personalized support, enhancing educational quality through the thoughtful integration of technology with established pedagogical principles.

2.5 Educational Content Generation and Processing

Recent advances in AI-powered content generation have created new possibilities for educational material development. Large language models and specialized educational AI systems can support various applications:

- Automated resource creation: Generation of structured notes, summaries, and study materials.
- Content adaptation: Transformation of complex materials into more accessible formats.
- Interactive learning support: Contextual question answering and personalized explanations.
- Cross-modal content processing: Integration of information across different media types.

Studies by Chen et al. (2022) indicate that these technologies can help educators create more engaging and personalized learning experiences while assisting students in processing and understanding complex information more effectively.

Chapter 3

Problem Statement

The rapid digitization of education has created an abundance of multimedia learning resources, yet students and educators face significant challenges in efficiently extracting, organizing, and personalizing this content. The Smart Education platform addresses this gap by developing an AI-powered system that transforms passive content consumption into active, personalized learning experiences.

Despite the mushrooming of educational videos and documents online, learners struggle with information overload, content relevance assessment, and effective knowledge extraction. Current solutions often fail to provide comprehensive tools for comparing multiple learning resources, generating structured notes from multimedia content, or facilitating contextual question-answering based on educational materials. Additionally, the time-intensive process of manual note-taking and content summarization creates barriers to efficient learning.

The Smart Education platform aims to solve these challenges by creating an integrated system that leverages artificial intelligence to analyze video transcripts and PDF documents, generate customized educational content, and provide interactive learning experiences. By developing a user-friendly interface that enables video comparison, automated note generation, and contextual Q&A functionality, this project seeks to enhance learning outcomes, improve content accessibility, and empower both students and educators with tools that transform how educational content is processed and utilized.

3.1 Project Planning

The actual planning steps followed were:

1) Initial Research and Concept Development
    i) Identified the need for AI-powered educational content analysis,
    ii) Researched existing solutions and their limitations,
    iii) Developed the core concept of integrating video analysis, note generation, and PDF Q&A.

2) Technology Selection
    i) Selected Streamlit for the web interface development,
    ii) Chose YouTube Transcript API for video content extraction,
    iii) Integrated GPT model for content processing,

iv) Selected PyMuPDF for PDF text extraction,
v) Chose ReportLab and python-docx for document generation.

3) Feature Prioritization
   i) Prioritized three core functionalities: video comparison, note generation, and PDF Q&A,
   ii) Defined minimum requirements for each feature,
   iii) Established development sequence based on dependencies.

4) Development Workflow Setup
   i) Created project folder,
   ii) Established development environment with necessary dependencies,
   iii) Set up API access and authentication for external services.

5) Iterative Development Cycles
   i) Implemented transcript extraction functionality,
   ii) Developed content generation capabilities,
   iii) Created PDF processing and Q&A features,
   iv) Built document export functionality.

6) Testing and Refinement
   i) Conducted functionality testing for each feature,
   ii) Refined model to give optimal results.

3.2 Project Analysis

A thorough analysis was conducted to identify ambiguities and refine requirements before implementation.

1) Requirements Analysis
- Clarity Assessment: Examined requirements for precise and clear descriptions.
- Completeness Verification: Ensured coverage of all necessary functionalities.
- Feasibility Study: Evaluated technical viability, particularly regarding API integrations.

2) Functional Requirements Refinement
- Video Transcript Processing: Defined video length limitations and handling of missing transcripts.
- Content Generation Parameters: Specified prompt engineering approaches and structure templates.
- PDF Processing Capabilities: Determined file size limitations and text extraction methodology.

3) Risk Analysis

- Technical Risks: Dependency on external APIs, processing limitations..
- Quality Risks: Variability in AI-generated content quality.
- Resource Risks: API usage costs, computational requirements

4) Mitigation Strategies

- Quality Assurance: Established content verification procedures to check whether the content was prepared using the user provided resources or not.

This analysis phase helped ensure development would proceed with clear direction and reduced risk of technical roadblocks.

3.3 System Design

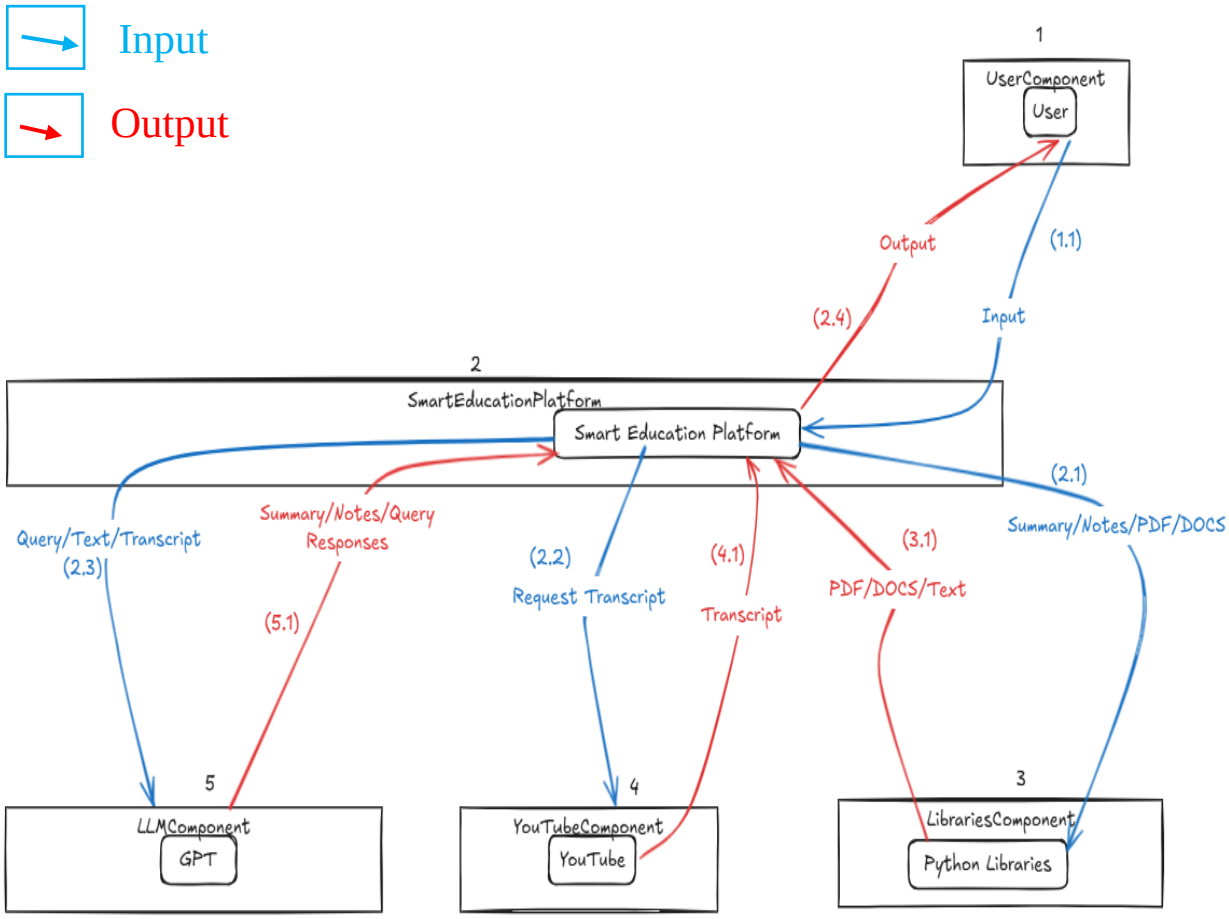

Figure 2 : System Architecture of Smart Education Platform

The Smart Education platform is primarily developed using Python, leveraging its extensive ecosystem of libraries and frameworks. Python's versatility allows for efficient implementation of both backend logic and frontend interface.

The system architecture is built around a Streamlit-based web interface, providing users with an interactive and responsive experience. Streamlit, a Python library,

enables rapid development of data-centric web applications, making it ideal for our educational content processing platform.

Key components of the system include:

- User Interface: Implemented using Streamlit, handling user inputs and displaying processed content.
- Video Processing: Utilizes the YouTube Transcript API, a Python library, to extract video transcripts.
- PDF Processing: Employs PyMuPDF, a Python binding for the MuPDF library, to extract text from PDF documents.
- AI Content Generation: Integrates with GPT model through their Python API, generating summaries, notes, and answers.
- Document Generation: Uses ReportLab and python-docx libraries to create downloadable PDF and DOCX files respectively.

The Python ecosystem's rich set of libraries enables seamless integration of these diverse functionalities. The language's simplicity and readability also facilitate maintenance and future enhancements of the platform.

This Python-centric approach allows for efficient data processing, AI integration, and web interface development within a unified programming environment, contributing to the overall cohesiveness and maintainability of the platform.

Key Components:

1) Python: A high-level, interpreted programming language known for its simplicity and readability. In this project, Python serves as the primary language for developing the entire Smart Education platform, leveraging its extensive library ecosystem for web development, data processing, and AI integration.

2) Pip: Python's package installer and dependency manager. Pip simplifies the process of installing and managing external libraries required for the project, ensuring all necessary components are properly integrated.

3) Streamlit: An open-source Python library for creating web applications with minimal front-end experience. Streamlit is used to build the interactive user interface of the Smart Education platform, enabling easy input of URLs, document uploads, and display of processed content.

4) YouTube Transcript API: A Python API for retrieving transcripts from YouTube videos. This library is crucial for extracting textual content from video inputs, providing the raw material for further analysis and processing.

5) PyMuPDF: A Python binding for the MuPDF library, offering fast and comprehensive PDF document processing capabilities. In this project, PyMuPDF is used to extract text content from user-uploaded PDF files for subsequent AI analysis.

6) OpenAI API: Provides access to OpenAI's powerful language models, including GPT-4o. The API is integrated to power the platform's content generation, summarization, and question-answering functionalities, forming the core of the AI-driven educational features.

7) ReportLab: An open-source engine for creating complex, data-driven PDF documents. ReportLab is employed to generate downloadable PDF versions of AI-generated summaries and notes, enhancing content accessibility.

8) python-docx: A Python library for creating and updating Microsoft Word (.docx) files. This component enables the platform to offer an alternative format for downloading generated educational content, catering to users who prefer Word documents.

9) Pandas: A powerful data manipulation and analysis library for Python. In the Smart Education platform, Pandas assists in managing and processing structured data, enhancing the efficiency of content handling and analysis tasks.

10) Requests: A simple, elegant HTTP library for Python. Requests facilitates communication with external APIs and services, playing a crucial role in integrating various components and retrieving data from external sources.

This comprehensive set of components, centered around Python and managed through Pip, enables the Smart Education platform to deliver a seamless, AI-powered educational experience while maintaining flexibility for future enhancements.

Chapter 4

Implementation

The implementation steps were:

Step 1: Environment Setup and Library Installation

The implementation began with setting up the Python environment for the Smart Education platform. Necessary libraries were installed using pip, including Streamlit for the web interface, OpenAI for AI integration, YouTube Transcript API for video content extraction, PyMuPDF (imported as fitz) for PDF processing, and document generation libraries like ReportLab and python-docx. The OpenAI client was configured with the appropriate base URL and API key to enable AI-powered content generation. Custom styling was implemented for the Streamlit interface to hide deployment buttons and sidebar headers, creating a cleaner user experience.

Step 2: User Interface Development

The main Streamlit application structure was created with a title and sidebar navigation. Three main functionalities were implemented: "Comparison" for comparing multiple YouTube videos, "Summarize & Create Note" for generating educational content from a single video, and "PDF Q&A" for answering questions about uploaded documents. The sidebar was designed using radio buttons to allow users to switch between these features. For each functionality, appropriate input fields were created such as text areas for YouTube links, file uploaders for PDF documents, and text inputs for user queries.

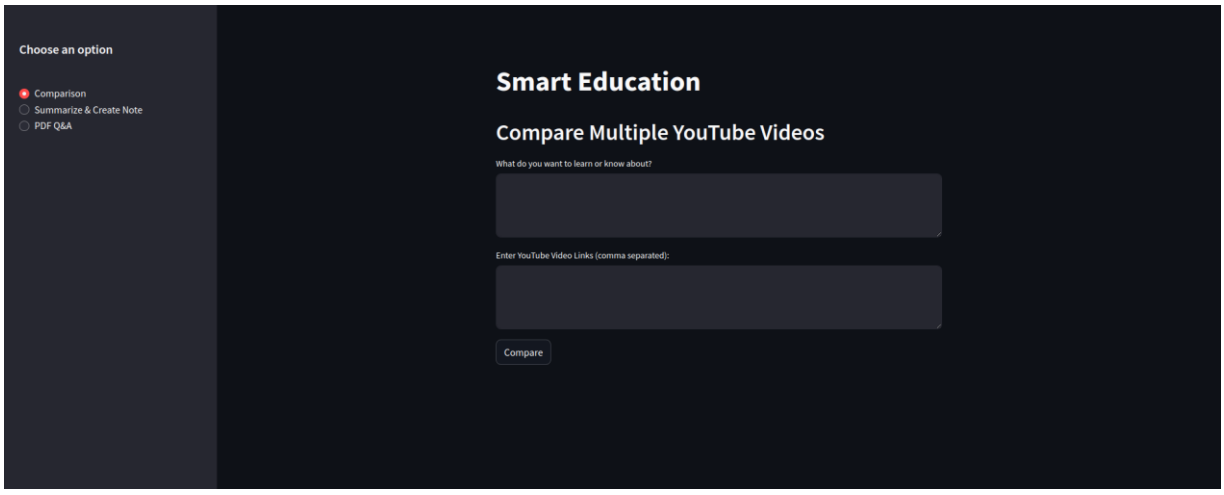

Figure 3 : UI Dashboard

Step 3: Prompt Engineering

Specialized prompts were developed for different content generation purposes. The "prompt_notes" was designed to create well-structured educational notes with introduction, key points, detailed explanations, and takeaways. The "prompt_summarize" was crafted to identify key themes like tutorials, reviews, and how-to guides while providing recommendations. The "prompt_comparison" was implemented to evaluate multiple videos for comparison purposes. These prompts were crucial for guiding the AI model to generate appropriate educational content.

```
def chunk_text(text, max_tokens=8000):
    enc = tiktoken.get_encoding("cl100k_base")
    tokens = enc.encode(text)

    chunks = []
    while len(tokens) > max_tokens:
        chunk = tokens[:max_tokens]
        tokens = tokens[max_tokens:]
        chunks.append(enc.decode(chunk))

    if tokens:
        chunks.append(enc.decode(tokens))

    return chunks
```

Figure 4 : Function for chunking

Step 4: Transcript Extraction and Processing

The transcript extraction functionality was implemented using the YouTube Transcript API. The "extract_transcript_details" function was created to take a YouTube URL as input, extract the video ID, retrieve the transcript in English or Hindi, and combine the transcript segments into a single text. Error handling was developed to manage cases where transcripts were unavailable or extraction failed.

Step 5: AI Content Generation Implementation

The AI content generation capability was developed using GPT model. The "generate_gemini_content" function was implemented to send prompts along with transcript text to the OpenAI API and retrieve the generated content. This function served as the core intelligence of the platform, enabling the generation of summaries, notes, and answers to user queries.

Step 6: Video Comparison Algorithm

Algorithms for comparing and recommending videos were developed. The "assess_relevance" function was implemented to evaluate video relevance to user queries, and the "suggest_best_video" function was created to identify the most suitable video based on content type and relevance scores. The comparison logic included scoring mechanisms that prioritized tutorials, reviews, and how-to guides based on their educational value.

Step 7: Document Generation Functionality

Functions to convert AI-generated content into downloadable formats were created. The "save_notes_as_pdf" function was implemented using ReportLab to generate PDF documents with proper formatting for headings, bullet points, and paragraphs. Helper functions like "add_line_to_pdf," "add_bullet_to_pdf," and "wrap_text" were developed to handle text formatting, wrapping, and pagination. The "save_notes_as_docx" function was implemented using python-docx to create Microsoft Word documents.

Step 8: PDF Processing and Question Answering

PDF document processing was implemented using PyMuPDF. The "extract_pdf_text" function was created to extract text content from uploaded PDF files by processing each page. The question-answering functionality was developed with the "get_pdf_answer" function that combined the extracted PDF text with user questions in a prompt for the OpenAI model, generating contextually relevant answers.

```
def extract_pdf_text(file):
    try:
        pdf_document = fitz.open(stream=file.read(), filetype="pdf")
        pdf_text = ""

        for page_num in range(pdf_document.page_count):
            page = pdf_document.load_page(page_num)
            pdf_text += page.get_text()

        return pdf_text
    except Exception as e:
        st.error(f"Error extracting text from PDF: {e}")
```

Figure 5 : Function for extracting text

Step 9: Session State Management

Session state management was implemented throughout the application to maintain context across user interactions. Separate state variables were created for conversation history, summaries, transcripts, and other persistent data. This ensured that users could ask multiple questions about the same content and maintain their context throughout the session.

Step 10: Feature Integration and Testing

All components were integrated into a cohesive system and thoroughly tested. The video comparison feature was implemented to allow users to input multiple YouTube URLs, extract transcripts, generate summaries, and receive recommendations. The note generation feature was developed to create structured educational content from videos with download options. The PDF Q&A feature was implemented to enable document upload, text extraction, and contextual question answering.

Chapter 5

Standards Adopted

5.1 Design Standards

The Smart Education platform adheres to several design standards to ensure quality, maintainability, and user experience:

- Modular Design Architecture: The project follows a modular design approach, separating functionality into distinct components (transcript processing, content generation, PDF processing) that can be developed and maintained independently.

- User Interface Design Standards: The Streamlit interface implements consistent navigation patterns and visual hierarchy, with clear separation between input areas and output displays.

- Prompt Engineering Standards: The project uses structured prompt templates for different content types, ensuring consistent AI-generated outputs across the platform.

- UML Component Diagram: While not explicitly shown in the code, the system architecture follows standard UML component relationships, with clear dependencies between modules.

- Data Flow Design: The application implements a standard data flow pattern where user inputs are processed through extraction, analysis, and presentation layers.

- Accessibility Standards: The interface design considers basic accessibility principles with clear visual hierarchy and descriptive elements.

5.2 Coding Standards

The Smart Education platform follows several coding standards and best practices:

- PEP 8 Compliance: The Python code adheres to PEP 8 guidelines for formatting, including proper indentation (4 spaces), appropriate line length, and consistent naming conventions.

- Descriptive Naming: Functions and variables use descriptive names that clearly indicate their purpose (e.g., extract_transcript_details, generate_gemini_content, save_notes_as_pdf).

- Function Modularity: Each function performs a single, well-defined task, enhancing readability and maintainability.

- Error Handling: Comprehensive try-except blocks are implemented throughout the code to gracefully handle exceptions and provide meaningful error messages.

- Code Organization: Related functionality is grouped together, with clear separation between UI components, processing logic, and utility functions.

- DRY Principle: The code follows the Don't Repeat Yourself principle, with common functionality extracted into reusable functions.

- Consistent Formatting: The code maintains consistent formatting for conditional statements, function definitions, and comment styles.

- State Management: The application uses Streamlit's session state consistently for maintaining conversation history and context.

- Documentation: Functions include implicit documentation through descriptive names and logical structure.

- Resource Management: The code properly manages resources like file handles and API connections.

5.3 Testing Standards

The Smart Education platform implements several testing standards to ensure quality and reliability:

- Functional Testing: Each component (transcript extraction, content generation, PDF processing) is tested for correct functionality according to requirements.

- Integration Testing: The interaction between components is verified to ensure proper data flow and processing.

- Error Handling Testing: The application is tested with various error conditions to ensure graceful handling of exceptions.

- User Interface Testing: The Streamlit interface is tested for usability and proper display of information.

- Content Verification: AI-generated content is evaluated for quality, relevance, and structure.

- Cross-functional Testing: The application is tested across different features to ensure consistent behavior.

- Performance Testing: Response times for API calls and content generation are monitored to ensure acceptable performance.

These standards ensure that the Smart Education platform delivers a reliable, maintainable, and user-friendly experience while adhering to established software engineering practices.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

The Smart Education platform successfully integrates AI-powered content analysis with multimedia educational resources to create a versatile learning tool. By leveraging GPT model, YouTube Transcript API, and PDF processing capabilities, the platform offers **three key functionalities**:

- Video comparison and recommendation based on user learning goals
- Automated note generation and summarization from YouTube videos
- Contextual question-answering for both video content and PDF documents

This implementation demonstrates the potential of AI in enhancing educational experiences by providing personalized content analysis, structured note-taking, and interactive learning support. The platform's ability to process multiple content types and generate downloadable study materials addresses the diverse needs of modern learners.

6.2 Future Scope

The Smart Education platform lays a foundation for several potential enhancements and expansions:

- Collaborative Learning Features: Implement shared note-taking and discussion functionalities to foster peer-to-peer learning and knowledge sharing.
- Adaptive Learning Paths: Develop AI-driven recommendation systems that suggest personalized learning sequences based on user interactions and performance.
- Integration with Learning Management Systems: Create plugins or APIs to integrate the platform's capabilities with existing educational ecosystems.
- Real-time Tutoring: Integrate live AI tutoring capabilities to provide immediate assistance and explanations to learners.
- Learning Analytics: Implement comprehensive analytics to track user progress, identify learning patterns, and provide insights to both learners and educators.

By pursuing these enhancements, the Smart Education platform can continue to evolve, offering increasingly sophisticated and personalized learning experiences that adapt to the changing landscape of education technology.

References

[1] Zhu, Z., Yu, M., & Riezebos, P. (2016). A research framework of smart education. Smart Learning Environments, 3(1), 1-17.

[2] Chassignol, M., Khoroshavin, A., Klimova, A., & Bilyatdinova, A. (2018). Artificial Intelligence trends in education: A narrative overview. Procedia Computer Science, 136, 16-24.

[3] Hwang, G. J., & Chen, N. S. (2023). Roles and applications of generative artificial intelligence in education: Current status and future directions. Computers and Education: Artificial Intelligence, 4, 100140.

[4] Kim, T., Cho, J. Y., & Lee, B. G. (2019). Evolution to smart learning in public education: A case study of Korean public education system. Open and Social Technologies for Networked Learning, 170-178.

[5] Spector, J. M. (2016). Smart learning environments: Concepts and issues. Society for Information Technology & Teacher Education International Conference, 2728-2737.

[6] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2019). MS MARCO: A human-generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268.

[7] Zeng, Z., Zhou, S., & Aggarwal, C. (2020). Graph-based text summarization: A comprehensive survey. arXiv preprint arXiv:2010.15221.

[8] Huang, L., Cao, Y., Xu, L., & Zhao, H. (2021). A survey of deep learning for text summarization. ACM Computing Surveys, 54(5), 1-38.

[9] Holmes, W., Bialik, M., & Fadel, C. (2019). Artificial intelligence in education: Promises and implications for teaching and learning. Center for Curriculum Redesign.

[10] Dillenbourg, P., & Jermann, P. (2010). Technology for classroom orchestration. New Science of Learning, 525-552.

[11] Basu, S., Jacobs, C., & Vanderwende, L. (2021). Powergrading: A clustering approach to amplify human effort for short answer grading. Transactions of the Association for Computational Linguistics, 9, 207-220.

[12] Chen, X., Xie, H., Zou, D., & Hwang, G. J. (2022). Application and theory gaps during the rise of artificial intelligence in education. Computers and Education: Artificial Intelligence, 3, 100067.

[13] Rudolph, J., Tan, S., & Tan, S. (2023). ChatGPT: Bullshit spewer or the end of traditional assessments in higher education? The Journal of Applied Learning and Teaching, 6(1), 1-14.

[14] Gillani, N., Eynon, R., Osborne, M., Hjorth, I., & Roberts, S. (2023). Artificial intelligence and education: A bibliography to help educators prepare students and themselves for a future shaped by AI. JSTOR Daily.

[15] Conrad, K. (2023). Sneak preview: A blueprint for an AI bill of rights for education. Critical AI, 2(1).

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

NIRANJAN SAH
21052944

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I focused on the document generation features. I implemented the PDF generation using ReportLab with custom formatting for different text elements. I developed the text wrapping algorithms, pagination handling, and styling for the generated documents. I also created the DOCX generation functionality using python-docx with appropriate formatting.

Individual contribution to project report preparation: I wrote about the testing and verification procedures. I outlined the testing strategies used to ensure the platform's functionality and reliability across different use cases.

Full Signature of Supervisor: Full signature of the student:

……. ……..

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

SUMIT CHAUDHARY
21052952

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I focused on the core architecture and system design. I developed the initial project structure and implemented the Streamlit interface framework. I created the custom styling for the web application and established the navigation system between different functionalities. I also coordinated the integration of all components to ensure cohesive functionality.

Individual contribution to project report preparation: I focused on the user interface and experience aspects. I documented the design choices made in creating an intuitive Streamlit interface and explained how the platform enhances the learning experience.

Full Signature of Supervisor: Full signature of the student:

……. ……..

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

NIKESH KUMAR MANDAL
21053299

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I specialized in the YouTube transcript processing functionality. I implemented the transcript extraction using the YouTube Transcript API, handling URL formats and language options. I also developed the error handling mechanisms for transcript retrieval failures and optimized the text processing pipeline for the extracted content..

Individual contribution to project report preparation: I focused on writing the introduction and literature review sections. I researched existing smart education platforms and AI applications in education to provide context for the project. I also compiled the list of academic references.

Full Signature of Supervisor: Full signature of the student:

……. ……..

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

ABHI UPADHYAY
21053401

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I developed the PDF processing capabilities and implemented the text extraction functionality using PyMuPDF also known as fitz, created the PDF question-answering system, and developed the conversation management for maintaining context across multiple queries. I also handled error cases for PDF processing and optimized text extraction for various document formats.

Individual contribution to project report preparation: I was responsible for documenting the system design and architecture. I created the system diagram and wrote detailed explanations of each component's role and interactions within the Smart Education platform.

Full Signature of Supervisor: Full signature of the student:

……. ……..

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

LOVE KUMAR SAH
21053435

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I was responsible for testing and user experience. I conducted comprehensive testing of all functionalities, identified and reported bugs, and suggested usability improvements. I also developed the session state management system for maintaining conversation history and context across user interactions, and created the download functionality for generated content.

Individual contribution to project report preparation: I was tasked with writing the conclusion and future scope sections. I summarized the project's achievements and outlined potential enhancements and expansions for the Smart Education platform.

Full Signature of Supervisor: Full signature of the student:

…… ……..

INDIVIDUAL CONTRIBUTION REPORT:

SMART EDUCATION PLATFORM

SUMAN SHARMA
21053443

Abstract: The Smart Education platform is an interactive web application leveraging AI to enhance learning through multimedia content analysis. It offers video comparison, note generation, and PDF-based question answering functionalities. The system extracts transcripts from YouTube videos and processes them using GPT model to generate summaries, notes, and answers.

The platform enables users to compare videos, create downloadable study materials, and engage in contextual Q&A sessions with both video and document content. It incorporates specialized components for transcript extraction, content generation, and document processing, providing a versatile educational tool that transforms passive content consumption into active, personalized learning experiences.

Individual contribution and findings: I worked on the AI integration aspects. I configured the OpenAI client connection and developed the specialized prompt engineering templates for different content types. I implemented the content generation functions that interact with the GPT model and created the algorithms for video comparison and relevance assessment.

Individual contribution to project report preparation: I handled the implementation methodology section. I described the step-by-step process of developing the platform, including the coding of key functions and integration of various libraries and APIs.

Full Signature of Supervisor: Full signature of the student:

……. ……..

"SMART EDUCATION PLATFORM"

ORIGINALITY REPORT

17%	15%	8%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.worldleadershipacademy.live Internet Source	4%
2	Submitted to University of Newcastle Student Paper	2%
3	www.coursehero.com Internet Source	1%
4	Submitted to KIIT University Student Paper	1%
5	Desheng Yan, Guangming Li. "Students' internal driving force or environment external driving force? Configuring digital learning power heterogeneity in a smart education environment", Interactive Learning Environments, 2023 Publication	1%
6	unesdoc.unesco.org Internet Source	1%
7	Submitted to Study Group Australia Student Paper	1%
8	koreascience.kr Internet Source	<1%
9	listens.online Internet Source	<1%
10	arxiv.org Internet Source	<1%