International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI Smart Note Using API

Ms. Roshini S N[1], Deepika S[2], Harish P[3], Kousik KY[4] ,Kowshik S[5]

[1] *DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE, SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY, (AUTONOMOUS) COIMBATORE-641062.*

[2] *DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE, SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY, (AUTONOMOUS) COIMBATORE-641062.*

[3] *DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE, SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY, (AUTONOMOUS) COIMBATORE-641062.*

[4] *DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE, SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY, (AUTONOMOUS) COIMBATORE-641062.*

[5] *DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE, SRI SHAKTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY, (AUTONOMOUS) COIMBATORE-641062.*

Received: ; Revised: ; Accepted: ; Published:

ABSTRACT:

Effective information management and comprehension have grown in significance in the current digital era. AI Smart Notes is a Chrome browser plugin that combines artificial intelligence with user-friendly note-taking features to improve online reading and learning. With the help of this tool, users can easily organize their notes, create AI-powered simplified explanations for complicated language, and highlight and save important information from web pages. AI Smart Notes provides a cutting-edge solution for scholars, students, and information seekers by fusing conventional note-taking with intelligent content analysis. The study shows how to effectively integrate AI with browser extension development to increase understanding and productivity in digital contexts.

KEYWORDS: AI-Powered Explanations, Chrome-Extension ,Note-Taking, Web Page Text Selection, Natural Language Processing (NLP), JavaScript ,Manifest File ,Background Script.

INTRODUCTION:

Effectively storing and organizing information has become a critical task in the fast-paced digital age we live in today. Users frequently find it difficult to efficiently remember and recall significant ideas due to the constant inflow of material from meetings, lectures, and internet content. The idea of intelligent note-taking systems has drawn a lot of attention as a solution to this gap. By incorporating artificial intelligence, AI Smart Note aims to transform the conventional note-taking method and make note management smooth, simple, and effective.

An AI-powered note-taking online application called AI Smart Note makes use of the latest advances in machine learning (ML) and natural language processing (NLP) models. In addition to providing intelligent search, summarization, and chatbot-assisted Q&A features, the system lets users write, modify, and arrange notes. Artificial Intelligence Smart Note turns static notes into dynamic knowledge assets that users can meaningfully interact with by utilizing technologies such as Pinecone for vector search, OpenAI's GPT, and MongoDB for structured storage.

This project's main objective is to improve users' comprehension and memory of information by turning their digital or printed notes into interactive material. The program guarantees a responsive and accessible experience on all devices thanks to its user-friendly interface, which is constructed on Next.js and stylized with Tailwind CSS. Students, professionals, and researchers can all benefit from the secure, customized access that is further ensured by the integration of Clerk for authentication.

AI Smart Note tackles the cognitive overload that comes with conventional note-taking techniques by providing users with AI-driven tools for summarization, contextual search, and fast Q&A. It encourages deeper comprehension and interaction with personal knowledge stores in addition to increasing productivity. In addition to examining the AI Smart Note system's possible effects on nowledge management and digital learning, this journal attempts to document the system's development, deployment, and assessment.

LITERATURE REVIEW:

Developments in machine learning, natural language processing, and educational technologies have all had a big impact on the creation of AI-driven smart note systems. The transformer architecture, which introduced the idea of self-attention and allowed models to better comprehend context and relationships within text, is at the heart of these advancements. Building on this, deep bidirectional transformers have been developed that can understand intricate linguistic patterns, which are necessary to produce notes that are logical and contextually aware. This skill has been further improved by generative models, which allow for the production of human-like explanations and summaries from unprocessed information.

By enabling tailored learning experiences and modifying content delivery according to each student's needs, artificial intelligence (AI) has revolutionized the education sector. AI integration in smart classrooms has resulted in more dynamic and effective learning settings where teachers and students benefit from automatic note generation and summarization. Big data technologies and strong data manipulation tools have enabled the processing of vast amounts of educational data, which is essential for training AI models on a variety of extensive datasets. Furthermore, improvements in lightweight text classification methods have made it possible to organize instructional materials more quickly and effectively.

When taken as a whole, these technical developments have opened the door for the creation of intelligent systems that can generate brief, organized, and customized notes, greatly improving the teaching and learning process in contemporary education.

SYSTEM DESIGN & ARCHITECTURE:

The **CodeEval** platform has been designed with an emphasis on providing an interactive, scalable, and secure environment for coding practice. It ensures seamless interactions between users, real-time code evaluation, and prompt feedback. The system's architecture is built to be efficient and easy to maintain, structured into multiple layers that separate the platform's functionality into clear components. This section outlines the core elements of the architecture, highlighting the technologies used to bring CodeEval to life.

Frontend Development:

SYSTEM DESIGN & ARCHITECTURE

The AI Smart Notes platform's system architecture prioritizes efficiency, scalability, and modularity. The platform combines elements of machine learning (ML) and natural language processing (NLP) to produce succinct, organized notes from a variety of content sources, including scholarly papers, lectures, and code explanations.

1. System Overview

The platform is based on a client-server architecture, with the backend managing databases, processing data, and executing AI models, while the frontend acts as the user interface. There are five primary layers to the system: Layers for input

- Preprocessing
- Model inference
- Postprocessing and summarization
- User interface

2. Components

a. Input Layer

- Users can upload or enter raw data (text, PDFs, lecture transcripts, and code snippets) via the input layer.
- Supports file types like.txt,.docx,.pdf, and.py.

b. Preprocessing Module

• Tokenizes and cleans input data.
• Segments sentences, eliminates noise, and uses stop words.
• Makes use of libraries such as HuggingFace Transformers, spaCy, and NLTK.

c. Model Inference Layer

- Makes use of pre-trained language models, such as BERT, T5, or GPT, to provide summaries and extract relevant information.

- Tools such as OpenAI Codex or CodeBERT can be used to provide comments or explanations for coding inputs.
- Key points are extracted and mapped to structured note templates, which include definitions, bullet points, and headings.

d. Summarization & Enhancement Module

• Makes use of techniques like abstractive and extractive summarization to enhance model outcomes.
• Enhances readability and ensures grammatical accuracy.
• Adds relevant code snippets or diagrams as needed, connecting to libraries like Matplotlib or Mermaid. It is optional to use JavaScript for diagrams.

e. Database Layer

• Uses PostgreSQL or MongoDB to store user history, inputs, and generated notes.
• Guarantees session management and data persistence.

f. Frontend/UI Layer

• Designed with Flask/Jinja or ReactJS templates, providing an intuitive and responsive user experience.
• Features include a theme changer, version history, download choices, input area, and note preview.
• Incorporates edit possibilities for manual correction or customization along with real-time feedback.

3. Technologies Used

• Frontend: JavaScript (React or Vue), HTML, and CSS Python (Flask or Django) for the backend Hugging Face Transformers (BERT, T5, GPT), spaCy, and NLTK are AI/NLP models. MongoDB and PostgreSQL are the databases. AWS S3 and Firebase Storage are the storage options.
• Implementation: Heroku, Docker, AWS, and Render

4. Data Flow Diagram (DFD)

Content is uploaded or entered by the user, processed by the preprocessing module, and thereafter Key summaries are produced by the NLP model; postprocessing enhances and organizes the note; The finished product is either downloaded or rendered in the frontend.

Backend Development:

The AI Smart Notes system's backend is in charge of managing data, responding to user inquiries, and executing the AI models that produce smart notes. It was created with Python and makes use of frameworks such as Flask or FastAPI to create RESTful APIs that are responsive and scalable. Features like file uploads, note creation, user authentication, and feedback submission are made possible by these APIs, which facilitate smooth communication between the frontend and the server.

The Natural Language Processing (NLP) engine, which is at the heart of the backend, incorporates strong transformer-based models like T5, BART, or BERT through the Hugging Face Transformers library. These models are used to process user input, documents, or lecture notes and provide clear, short summaries. Tokenization, text cleaning, model inference, and formatting are among the activities that the backend manages to guarantee that the output is understandable and organized.

The system uses either MongoDB or PostgreSQL as its database to store and manage user data and created notes. JWT tokens are used to securely handle user authentication, enabling customized access to saved notes. Additionally, version tracking and indexing of generated notes for quick retrieval are supported by the backend. Moreover, text is extracted from uploaded PDF or DOCX files using programs like PyPDF2 and python-docx, readying them for summary.

The backend employs asynchronous task processing with Celery and Redis for improved security and speed, particularly for lengthy summary tasks. Rate restriction is used to control traffic, and input data is cleaned to avoid security flaws. The solution is scalable and usable in the real world because the whole backend is containerized using Docker and can be set up on cloud platforms like Heroku, Render, or AWS.

Database Design & Management:

The AI Smart Notes system's database architecture is designed to provide effective management, retrieval, and storage of user information, submitted files, and generated summaries. Depending on scalability requirements, either a NoSQL database like MongoDB or a relational database like PostgreSQL can be utilized. The model is made to accommodate numerous users, each of whom has a unique history of activity logs, written notes, and uploaded files. Strong data integrity and easy access to crucial information are maintained by the structure of tables and collections.

Models like Users, Documents, Summaries, and Feedback form the foundation of the database. Metadata, preferences, and user credentials are kept in the Users table. File format, upload time, and processing status are all stored in the Documents table, which also manages uploaded files and associates them with a particular user. When a summary is created, it is saved in the Summaries database with timestamps and a link to the source document. Users can read and download their summary at any moment thanks to this structure.

While the metadata of the documents is indexed in the database, the actual documents may be stored in a cloud bucket (such as AWS S3). In order to enhance performance for large-scale operations, the database design additionally includes indexes on commonly queried fields like user_id, document_type, or created_at. Constraints and data validation are used to minimize redundancy and guarantee consistency.

During development and maintenance, tools like pgAdmin, MongoDB Compass, or DBeaver are used to manage and monitor data. To protect user data and guarantee seamless upgrades, backup plans and migration scripts are used. All things considered, the database system is built with efficiency, security, and scalability in mind; it facilitates smooth backend integration and offers a strong basis for the workflow that generates notes using AI.

Real-Time Feedback and Interaction:

The **Real-Time Feedback and Interaction** module plays a vital role in enhancing user experience and learning efficiency in the AI Smart Notes system. This feature allows users to interact dynamically with the platform while receiving immediate feedback on their inputs, note requests, or queries. By incorporating real-time capabilities, users are not only engaged but also empowered to fine-tune the output based on their preferences.

The WebSocket protocol, also known as server-sent events, is one of the key elements that makes this possible. It keeps the line of communication open between the client and server. Instant response messages and progress indications appear when a user uploads a document or asks a summary. This removes the ambiguity that is frequently present in asynchronous systems and provides a sense of responsiveness. Additionally, to effectively guide the user during text entry or query formulation, live typing feedback, autocomplete, and suggestions are offered.

Additionally, the system has feedback methods following the generation of a note or summary. Users can request a rewording or a different tone (formal or casual), score the output's quality, and point out places that need work. The AI instantaneously processes this input and modifies its subsequent output accordingly. This makes the system smarter and more user-centric over time by helping to tailor the results according to the user's interaction history.

Depending on the tech stack, solutions like Socket.IO, Firebase Realtime Database, or Pusher can be used to manage these dynamic interactions. Low-latency communication and seamless synchronization between user input and system reactions are guaranteed by these techniques. All things considered, real-time feedback fosters a responsive, flexible, and captivating learning and content production environment, which enhances usability and increases confidence in the system's capabilities.

RELATED WORKS:

Artificial intelligence integration into educational systems has been the subject of numerous studies and applications, especially in the areas of intelligent note production and summarization. Early studies concentrated on conventional extractive summarizing strategies, which chose important phrases from papers without fully comprehending their context. More context-aware summarization tools started to appear as machine learning, particularly deep learning models like LSTMs and GRUs, gained popularity. These tools greatly increased the relevance and coherence of outputs.

The field has undergone a revolution because to recent innovations like the Transformer architecture. Models such as BERT and GPT showed remarkable abilities in comprehending natural language and producing text that resembled that of a person. Smart note recommendations, automatic content summarizing, and even the creation of customized learning materials are all features that these models have been integrated into educational applications. AI can help students quickly understand key concepts from extensive academic texts, as demonstrated by platforms like QuillBot, SMMRY, and technologies coupled with ChatGPT or Google's T5.

AI-based platforms have begun to prioritize improving personalization and interactivity in addition to summarization. To accommodate the needs of many learners, systems now include voice input, multilingual processing, real-time feedback mechanisms, and style customization. These characteristics improve the accessibility and engagement of educational resources. The significance of explainable AI (XAI) in education has also been emphasized by research, which shows that openness in AI judgments increases user confidence and instructional value.

There is still a lack of platforms that integrate real-time interaction, excellent AI-generated summaries, customisation, and user-friendly design into a single framework, despite the fact that numerous tools and research initiatives have advanced this sector. In

order to close this gap, this project expands on current technologies and provides a unified, AI-powered smart notes system that is easy to use and interactive.

METHODOLOGY:

The AI Smart Notes system was developed utilizing an iterative, controlled process that focuses on integrating AI with web development and natural language processing technologies. The main objective is to provide an intuitive platform that can produce intelligent, real-time summaries from instructional materials while allowing for interactive elements like customization and feedback.

We identified essential features including text input, summarization, user interaction, and real-time processing during the first step, which entailed requirement analysis and data collection. To train and assess the AI model, datasets comprising scholarly papers, lecture notes, and online instructional materials were collected. Tokenization, stop-word removal, and normalization were among the preprocessing methods used to clean and get the data ready for model training.

We then used transformer-based models, like BERT and GPT, that were refined on instructional materials to produce succinct and insightful summaries throughout the AI and NLP integration phase. To guarantee contextual relevance and coherence, extractive and abstractive techniques were used to further improve the summarization logic. The summarizing algorithms were implemented and evaluated using open-source tools such as NLTK and Hugging Face Transformers.

Modern web technologies were used to construct the frontend and backend. The backend was driven by Node.js and Express.js, while the interactive user interface was created using React.js. WebSockets and asynchronous processing were used to provide real-time interaction. Because of its scalability and adaptable schema structure, which accommodate changing content types, MongoDB was selected for database administration.

Lastly, user input gathering, performance testing, and unit testing were all part of the testing and evaluation phase. Accuracy benchmarks and usability measurements were used to improve the system. In addition to being technically sound, this approach guarantees that the AI Smart Notes system is user-focused and flexible enough to accommodate a range of educational use cases.

IMPACTS & BENEFITS:

The AI Smart Notes system offers a revolutionary way for teachers and students to interact with course content. Through the use of sophisticated natural language processing, the platform drastically cuts down on the amount of time needed to comprehend and edit academic material. It helps readers rapidly and effectively understand important concepts by distilling long texts into succinct, insightful summaries. This encourages active learning and improves understanding, especially when it comes to difficult subjects.

Since a variety of users, including those with learning disabilities or language difficulties, may utilize the instrument, one of the main effects is the democratization of education. A personalized learning experience is made possible by the interactive elements and real-time feedback that let users alter the style and depth of their notes. This flexibility aids in accommodating different learning preferences, including interactive, textual, and visual.

From an academic standpoint, the tool improves performance and efficiency, particularly while preparing for tests or quickly reviewing material. It facilitates note-taking, reduces information overload, and improves recall of the material. It provides teachers with a means of pre-processing vast amounts of text into easily readable material for online platforms or classroom delivery.

Last but not least, the system offers scalable, AI-driven solutions for knowledge management, training modules, and documentation, which could be useful in corporate and institutional training settings. It is a useful tool in contemporary digital education and learning ecosystems because it automates the note-taking and summary process, which lowers human labor and increases overall efficiency.

RESULTS:

The AI Smart Notes system performed admirably when it came to automating the process of creating notes and summarizing a variety of academic material. The system successfully transformed long paragraphs, articles, and lecture materials into succinct, pertinent, and understandable notes by utilizing deep learning models and natural language processing.

The platform was tested using a variety of inputs, including research papers, lecture transcripts, and chapters from textbooks. The core of the original information was preserved thanks to the generated summaries' excellent accuracy and semantic relevance. Many users praised the AI-generated notes' clarity and utility during revision sessions, and many noted a discernible increase in study efficiency.

Furthermore, real-time interaction elements like keyword-based highlighting and adjustable note length greatly increased user comprehension and engagement. The platform was a flexible tool for students of different skill levels because it could modify its output according to user preferences, whether they required a detailed summary or a quick overview.

All things considered, the system was successful in facilitating a more targeted and customized learning experience while also streamlining the process of consuming knowledge. According to the findings, there is a good chance that it will be widely adopted in educational and professional settings.

CONCLUSION:

The AI Smart Notes system uses artificial intelligence to create succinct and insightful summaries of difficult academic material, offering a revolutionary approach to contemporary education. By combining real-time user interaction, machine learning, and natural language processing, the platform makes taking notes easier, improves comprehension, and saves both professionals and students a significant amount of time.

Through the delivery of tailored, pertinent, and organized notes, the project effectively addressed major issues with material comprehension and information overload. Its practical utility in a variety of educational environments is further enhanced by its user-centric design and flexible features, which encourage efficiency and engagement.

AI-driven technologies like Smart Notes have the ability to transform conventional teaching strategies and make knowledge acquisition more efficient and accessible as education continues to change in the digital age. This system has the potential to be a useful tool in corporate training settings, self-learning platforms, and classrooms with further development and wider use.

In the end, AI Smart Notes raises the bar for intelligent educational support by enhancing learning quality and enabling users to concentrate more on critical thinking and less on manual summary.

REFERENCES:

In AI SMART NOTES USING NLP, a variety of resources and references have been consulted to ensure its successful design and implementation. These references include books, journals, and online platforms that provide valuable insights into competitive coding, system design, machine learning, and best practices in software development. The resources range from authoritative textbooks on algorithms and programming languages to research papers on online platforms and coding systems. Additionally, documentation from reputable websites has been referenced to stay updated on the latest trends and technologies relevant to the project's development. These references have played a crucial role in guiding the platform's architecture, ensuring its scalability, and supporting the integration of key features.

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is All You Need*. Proceedings of the 31st International Conference on Neural Information Processing Systems, 30, 6000–6010.

Introduced the Transformer architecture, which is foundational for many modern NLP models powering AI-based note-generation tools.

https://doi.org/10.5555/3295222.3295349

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL-HLT 2019, 4171–4186.

Presented BERT, a bidirectional transformer model widely used for summarization and question-answering in smart note systems.

https://doi.org/10.18653/v1/N19-1423

[3] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-training*. OpenAI.

Laid the foundation for generative pre-trained transformers (GPT), enabling context-rich summarization for note generation tools.

https://openai.com/blog/language-unsupervised

[4] Zhang, B., & Wang, H. (2020). *AI-Assisted Personalized Learning: A Systematic Review*. Educational Technology Research and Development, 68(5), 2247–2273.

Explores how AI personalizes the learning experience, aligning with Smart Notes' goal of individualized content delivery.
https://doi.org/10.1007/s11423-020-09797-3

[5] Chen, M., Mao, S., & Liu, Y. (2014). *Big Data: A Survey*. Mobile Networks and Applications, 19(2), 171–209.
Provides insight into data processing techniques relevant for large-scale educational content summarization.
https://doi.org/10.1007/s11036-013-0486-3

[6] Kumar, A., & Bansal, M. (2020). *AI-Based Smart Classroom Systems: A Survey*. International Journal of Engineering and Advanced Technology, 9(4), 4381–4386.
Surveys AI tools in educational settings, laying groundwork for integrating smart notes into smart classroom ecosystems.
https://doi.org/10.35940/ijeat.D7545.129420

[7] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, 51–56.
Introduces essential Python libraries like Pandas, which support backend development and data processing in AI note systems.
https://doi.org/10.25080/Majora-92bf1922-00a

[8] Joulin, A., Grave, E., Mikolov, T., Ranzato, M. A., & Lécun, Y. (2017). *Bag of Tricks for Efficient Text Classification*. arXiv:1607.01759.
Proposes efficient NLP techniques that help accelerate smart note generation and classification.
https://arxiv.org/abs/1607.01759

[9] Smith, S., & Jackson, M. (2019). *The Role of AI in Modern Education: Enhancing Teaching and Learning Outcomes*. Journal of Educational Computing Research, 57(6), 1320–1345.
Discusses the integration of AI into education and its effectiveness in improving comprehension through tools like Smart Notes.
https://doi.org/10.1177/0735633119844239

[10] Li, X., & Li, H. (2019). *A Survey of Deep Learning in the Field of Natural Language Processing*. International Journal of Computer Science and Information Security, 17(7), 225–235.
Covers deep learning advancements in NLP, forming the technical base of AI-powered summarization tools.
https://www.ijcsis.org