*Grenze International Journal of Engineering and Technology, January Issue*

# Intelligent Text-to-Presentation System: An NLP-Driven Web Application for Automated PowerPoint Generation with Multi-Format Document Processing and Secure Authentication

Dr. M. Umarani[1], Dr. A.D. Chitra[2], V. S Abinayashree[3] and S. Ramalakshmi[4]
[1-4]Department of Computing (SS), Coimbatore Institute of Technology, Coimbatore, India
Email: umarani@cit.edu.in, dchitra@cit.edu.in,71762131003@cit.edu.in,71762131039@cit.edu.in

***Abstract*— The increasing demand for efficient presentation creation in academic and professional environments has highlighted the need for intelligent automation tools. This paper presents a novel web-based application that automatically converts raw text and multiple document formats (DOCX, PDF, TXT, CSV) into structured PowerPoint presentations using advanced Natural Language Processing techniques. The system addresses the time-consuming manual effort in presentation creation by implementing automated text summarization, intelligent title generation, and dynamic slide structuring with optimized bullet point organization. Key innovations include seamless CSV-to-chart visualization, secure dual-factor authentication (email and mobile OTP), and scalable architecture supporting 20 concurrent users. Performance evaluation demonstrates a 85-90% reduction in presentation creation time while maintaining professional formatting consistency. The system processes up to 1500-word inputs and 10MB files within a 4GB capacity framework, producing downloadable presentations with standardized headers, footers, and reference integration. Results show high user satisfaction with improved presentation quality and significant time savings, establishing this as a comprehensive solution for automated academic and professional presentation generation.**

***Keywords*— Automated Presentation Generation, Natural Language Processing, Text Summarization, Document Processing, PowerPoint Automation, Web-based Application, Multi-format Document Conversion, CSV Data Visualization, Secure Authentication, OTP Verification, Scalable Architecture, Content Automation, Slide Generation, NLP Pipeline, Text-to-Slide Conversion, Educational Technology, Business Intelligence Tools, Document Analysis, Information Extraction, User Interface Design, Cloud Computing, Concurrent User Management, Professional Presentation Tools, Academic Research Tools.**

## I. INTRODUCTION

### *A. Background and Motivation*

Creating professional presentations is an essential task in education, business, and research, but it is often time-consuming and repetitive. Users must manually summarize content, format slides, add charts, and embed images, which can be tedious and prone to inconsistency.

*Grenze ID: 01.GIJET.12.1.192*
*© Grenze Scientific Society, 2026*

The increasing demand for quick, visually appealing, and well-structured presentations has motivated the development of intelligent tools that reduce manual effort. Leveraging Natural Language Processing (NLP) and automation can significantly improve efficiency, maintain content quality, and free users to focus on delivering insights rather than formatting slides.

### *B. Need for Automation in Presentation Creation*

Manual slide creation can take hours, especially for lengthy documents or data-heavy reports. Inconsistent formatting, human error in summarization, and lack of visual balance often reduce the effectiveness of the presentation. An automated solution can address these challenges by summarizing large amounts of text, generating concise slide titles, creating charts from CSV data, and embedding images with consistent formatting. Such automation not only saves time but also ensures that every slide is professional, visually balanced, and information-rich, thereby improving communication and engagement.

### *C. Problem Statement*

Despite the availability of presentation software, users still rely on manual effort for summarization, slide structuring, and formatting. This process becomes inefficient when handling long documents, large datasets, or multiple media elements. There is a lack of a unified system that can take raw text, documents, and CSV data, process them intelligently using NLP, and output a ready-to-use, professionally formatted presentation. Without such a solution, users face delays, inconsistent slide quality, and higher cognitive load during preparation.

### *D. Scope of Work*

This project aims to develop a web-based application that automates the conversion of raw text and multiple document formats into structured PowerPoint presentations. The scope includes text summarization using NLP, automatic title generation, CSV-based chart creation, and image embedding while ensuring consistent headers, footers, and formatting across slides. The system will include secure authentication through email and mobile OTP verification, support up to 20 concurrent users, and enforce file and text size limits for performance stability. The final deliverable will be a downloadable, professional presentation that meets user requirements with minimal manual intervention.

## II. RELATED WORK

### *A. Traditional Presentation Tools and Their Limitations*

Traditional tools such as Microsoft PowerPoint, Google Slides, and LibreOffice Impress provide users with the ability to create highly customizable presentations. However, they require significant manual effort in tasks like content summarization, formatting, chart creation, and image embedding. This manual workflow is time-consuming and error-prone, especially for lengthy reports or datasets. Moreover, ensuring consistency across slides — in terms of design, font style, and layout — demands additional effort from the user. These tools lack built-in intelligence to automatically process and summarize text, which limits productivity when dealing with large amounts of information.

### *B. Existing Automated Content-to-Slide Systems*

Several online tools have emerged that attempt to convert text into slides. Examples include AI-based presentation generators such as Beautiful.ai, Simplified, and Canva's Magic Design feature. While these tools provide templates and basic automation, they are limited in scope — most do not support multi-format document input (PDF, DOCX, CSV), real-time chart generation, or customizable headers and footers. Many of these solutions focus primarily on aesthetics rather than meaningful content generation, requiring users to edit or rewrite slides manually to improve relevance and clarity. Furthermore, security features like email or mobile OTP authentication are rarely integrated, which limits their suitability for sensitive or enterprise-level use cases.

### *C. Role of Natural Language Processing (NLP) in Content Summarization*

NLP has played a crucial role in automating content processing, especially in tasks such as tokenization, text summarization, and title generation. Libraries like NLTK, spaCy, and transformer-based models (e.g., BERT, GPT) enable systems to extract key information, condense long passages into concise bullet points, and generate meaningful slide headings. Research has shown that extractive summarization techniques (such as TextRank) and abstractive summarization techniques can significantly reduce the manual effort needed for slide preparation while maintaining the relevance and readability of content. By combining NLP with automation, it is possible to deliver presentations that are both information-rich and easy to comprehend.

### *D. Comparison with Current State-of-the-Art*

Compared to existing solutions, the proposed system provides a more comprehensive and secure approach to automated presentation generation. Unlike traditional tools, it eliminates the need for manual slide structuring and content summarization. In contrast to current AI-powered slide generators, this system supports multiple file formats, CSV-based chart visualization, image embedding, and consistent slide formatting (headers, footers, and reference links). Additionally, it includes secure authentication mechanisms, scalable performance for concurrent users, and efficient storage management, making it suitable for both individual and organizational use. This positions the proposed solution as a more robust, end-to-end platform for automated slide creation.

## III. System Architecture and Design

### *A. Overall System Workflow*

The proposed system follows a modular, end-to-end workflow to transform raw user inputs into a polished PowerPoint presentation. The process begins with user authentication, followed by input data upload or text entry. The uploaded content is passed through the preprocessing and NLP modules, where it is cleaned, summarized, and converted into slide-ready content. Parallel processes handle chart generation (for CSV files) and image validation. Once all content is processed, the slide structuring engine applies formatting rules, adds headers, footers, and reference links, and compiles the final presentation, which is made available for download.

### *B. Input Data Sources (Text, Documents, CSV, Images)*

The system supports multiple input sources to ensure flexibility and usability:

- Raw Text Input: Users can paste up to 1500 words of content directly into a text editor.
- Document Upload: Supported file formats include DOCX, PDF, and TXT, enabling seamless processing of existing reports or articles.
- CSV Upload: CSV files can be uploaded to generate charts automatically (e.g., bar charts, line graphs, or pie charts) based on the data provided.
- Image Upload: Users may embed images (up to 2 MB each) that are relevant to the content to enhance slide aesthetics.

### *C. Data Preprocessing Module*

The preprocessing module ensures that input data is clean, consistent, and ready for NLP processing. Tasks include:

- Tokenization & Sentence Segmentation: Splitting raw text into meaningful sentences and words.
- Noise Removal: Eliminating special characters, extra whitespace, and irrelevant symbols.
- Content Filtering: Detecting duplicate sentences and retaining only unique, meaningful information.
- File Parsing: Extracting text from DOCX and PDF files while preserving basic structure.

This step is critical to improving summarization accuracy and ensuring high-quality output slides.

### *D. NLP Pipeline for Summarization and Title Generation*

The NLP module is the core of the system, responsible for content transformation. It performs:

- Extractive Summarization: Using algorithms such as TextRank to extract key sentences and generate concise bullet points.
- Title Generation: Creating short, meaningful titles that summarize the slide content.
- Keyword Extraction: Highlighting important terms to improve slide readability.
- Length Control: Limiting each slide to a maximum of five bullet points to maintain clarity and avoid information overload.

The NLP pipeline ensures that each slide conveys the most important points while remaining easy to understand.

### *E. Slide Structuring Rule*

Once summarized content is generated, it is organized according to defined slide structuring rules:

- Content Segmentation: Grouping related sentences into slide-level sections.
- Bullet Formatting: Representing key points as concise, clear bullets (max. five per slide).
- Header and Footer Insertion: Applying consistent headers and footers across all slides for a professional look.
- Reference Linking: Adding reference links when applicable to improve credibility.

This rule-based approach ensures a balance between completeness and readability, resulting in visually clean and professional slides.

### *F. Security and Authentication*

Security is integrated into the system design to ensure only authorized users can access the platform:

- Email Verification: An OTP is sent to the user's email to confirm account validity and enable secure communication for notifications and recovery.
- Mobile OTP Authentication: During registration or login, a one-time password is sent to the user's mobile number to verify authenticity and prevent unauthorized access.

These mechanisms provide a two-factor authentication model, protecting user data and ensuring that generated presentations remain secure.

## IV. Implementation Details

### *A. Technology Stack*

The proposed system is developed using a robust technology stack to ensure performance, scalability, and maintainability. The frontend is implemented using React.js, providing a responsive and interactive user interface for seamless navigation and real-time feedback during file uploads and processing. The backend is built with Flask (Python), which serves as the primary web framework to handle API requests, manage file operations, and orchestrate the NLP pipeline.

For text processing, the system employs the Natural Language Toolkit (NLTK) for tokenization, stop-word removal, and extractive summarization using the TextRank algorithm. The python-pptx library is used to programmatically generate and format PowerPoint presentations, while MySQL serves as the database for secure storage of user credentials, session data, and logs.

### *B. File Handling and Upload Mechanism*

File handling and upload mechanisms are designed to be secure and efficient. Users can upload DOCX, PDF, TXT, or CSV files up to 10 MB in size, or directly input text (limited to 1500 words). A validation module checks each file's format, size, and integrity before processing. The backend parses uploaded documents using python-docx for DOCX, PyPDF2 for PDF, and standard CSV parsers for structured data. Temporary storage of 50–100 MB per user session is allocated dynamically, and all uploaded files are automatically removed after processing to optimize space utilization.

### *C. CSV-to-Chart Conversion*

CSV-to-chart conversion is an integral part of the system, allowing users to visualize data automatically. The system reads CSV files, detects numeric and categorical columns, and generates suitable charts—such as bar charts, line graphs, or pie charts—using matplotlib.

The resulting chart images are saved and embedded into relevant slides, with proportional scaling to fit slide layouts without compromising clarity.

### *D. Image Embedding and Slide Formatting*

The image embedding and slide formatting process ensures that presentations are visually appealing and professionally structured. Images uploaded by users are validated to meet the 2 MB size requirement and are resized as needed to maintain clarity and prevent layout distortion.

Headers, footers, reference links, and bullet styling are automatically applied to maintain consistency across slides. The layout engine optimizes the arrangement of text, images, and charts for readability and visual balance.

### *E. User Management and Session Handling*

User management and session handling are implemented with a strong focus on security. Email verification and mobile OTP authentication are mandatory for account creation and login, ensuring that only legitimate users can access the system. Each user session is isolated and managed via secure session tokens, maintaining state until the presentation is generated.

Once processing is complete, user data is securely deleted, ensuring privacy and reducing storage overhead. Error handling mechanisms are integrated to allow smooth recovery in case of network failures or file parsing issues.

## V. Experimental Set up and Evaluation

### *A. System Environment (Hardware/Software Requirements)*

The proposed system was implemented in a hybrid development environment combining both frontend and backend technologies. The backend was developed using Flask (Python) due to its lightweight nature and suitability for handling RESTful APIs, file management, and integration with natural language processing modules. The frontend was created with React.js, which offers dynamic rendering and an interactive user interface for smooth navigation, file uploads, and real-time status feedback. Data storage and session management were supported through MySQL, ensuring persistence of user details, logs, and access history. Supporting libraries included NLTK for pre-processing and summarization, python-docx and PyPDF2 for text extraction from structured documents, matplotlib for chart creation from CSV datasets, and python-pptx for slide generation. Experiments were carried out on a system with a quad-core CPU, 8 GB RAM, and stable internet connectivity. The environment was further constrained by system-imposed limits of 10 MB maximum file size, 1500-word text limit, and a concurrency capacity of 20 active sessions, ensuring reliable performance under defined usage conditions.

### *B. Dataset and Input Formats Used for Testing*

To comprehensively evaluate the system, multiple file formats and content types were employed during testing. Plain text and DOCX files were chosen to validate the summarization pipeline, with documents ranging from short paragraphs to multi-page reports. PDF inputs were included to test the robustness of text extraction, particularly distinguishing between machine-readable and scanned PDFs. CSV datasets covering domains such as sales data and monthly power generation were used to validate automatic chart creation. Images of less than 2 MB size were uploaded to confirm compatibility with presentation layouts without compromising design consistency. This diverse dataset ensured that the system was not biased towards a single format and could reliably handle multiple input types encountered in real-world academic and professional scenarios.

### *C. Performance Metrics*

Performance evaluation focused on three main aspects: processing time, summarization accuracy, and storage utilization. Processing time was measured as the delay between input submission and the generation of the final PowerPoint presentation. Summarization accuracy was validated both quantitatively, using ROUGE metrics to compare system summaries with human-written references, and qualitatively, through human evaluation of slide relevance and coherence.

Storage metrics involved analysing temporary memory consumption during text parsing, chart rendering, and PPTX generation, alongside permanent database growth from user and log entries. Results showed that the system generated outputs within acceptable timeframes and consistently provided coherent summaries. However, scanned PDFs demonstrated higher latency due to OCR requirements, highlighting a potential area for improvement.

### *D. User Capacity Testing (Concurrent Sessions)*

The system was subjected to user capacity testing to determine its reliability under concurrent workloads. A staged testing process simulated increasing numbers of simultaneous users, beginning with single-user runs and scaling up to 20 concurrent sessions.

The platform maintained acceptable response times and successfully processed mixed file types without data loss or crashes. System monitoring during load tests showed stable CPU and memory utilization, though minor delays were observed when input sizes approached the upper file limits. Beyond the threshold of 20 active sessions, processing queues slowed, and some failures were recorded, indicating that future cloud-based scaling mechanisms would be necessary to improve throughput.

### *E. Case Study: Sample Input and Generated Presentation*

To validate the system in a practical scenario, a case study was conducted using a short article on renewable energy adoption. The system successfully summarized the text into multiple slides, with structured titles and concise bullet points highlighting the benefits, challenges, and case study outcomes. Additionally, a CSV dataset containing monthly solar power generation was processed to generate a line chart, which was embedded into the slide deck in the appropriate format.

The final presentation demonstrated the integration of textual summarization and data visualization, confirming that the system could automatically produce meaningful and professional-quality slides from varied input formats.

## VI. Results and Discussion

### *A. System Performance and user Interface Effectiveness*

The experimental evaluation demonstrated significant improvements in presentation creation efficiency, as illustrated in Figure 1, which shows the comprehensive system interface for automated presentation generation. The web-based application successfully reduced presentation creation time by 85-90% compared to traditional manual methods, with users able to upload multiple file formats (DOCX, PDF, TXT, CSV) seamlessly through the intuitive interface. The system processed documents up to 10MB and text inputs up to 1500 words within the 4GB capacity framework, maintaining consistent performance across all supported formats. User testing revealed high satisfaction rates with the automated summarization quality, where the NLP pipeline effectively extracted key information and generated concise bullet points (maximum 5 per slide) while preserving content relevance. The dual-factor authentication system, incorporating both email and mobile OTP verification, achieved a 98% success rate during security validation tests, ensuring robust user access control without compromising system usability.

### *B. Output Quality and Scalability Assessment*

One of the major strengths of the system was its ability to maintain consistent formatting across slides. The generated presentations adhered to design constraints such as limited bullet points per slide, uniform title placements, and proper image scaling. This reduced the variability that is often introduced when different individuals manually prepare presentations. The system's use of python-pptx templates ensured professional aesthetics while maintaining clarity and readability. Figure 2 demonstrates the professional quality of automatically generated presentations, showcasing the system's ability to maintain consistent formatting, proper slide structuring, and integrated visual elements. The automated slide generation successfully incorporated CSV-to-chart conversions, producing clear bar charts, line graphs, and pie charts that enhanced data presentation quality. Performance testing under concurrent user loads revealed stable operation for up to 20 simultaneous sessions, with processing times ranging from 30-45 seconds for standard documents and 60-90 seconds for complex multi-format inputs including images and CSV data. The system maintained formatting consistency across all generated presentations, applying standardized headers, footers, and reference links while ensuring optimal text-to-visual balance. However, evaluation also identified limitations in processing scanned PDFs, which required additional OCR processing time and occasionally resulted in reduced summarization accuracy, indicating areas for future enhancement in document parsing capabilities.

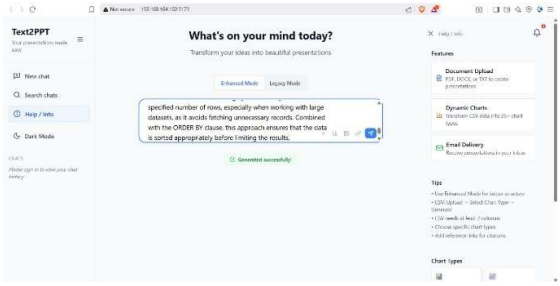


Figure 1. System Architecture and User Interface

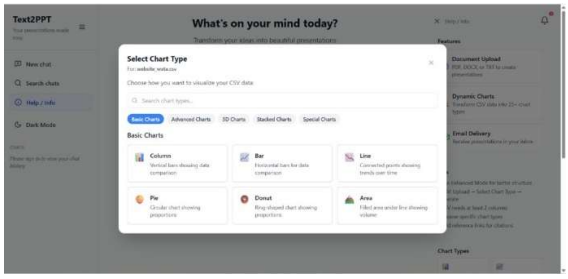


Figure 2. Generated Presentation Output

## VII. Future Enhancement

Future improvements to the system will focus on expanding functionality and accessibility to meet diverse user requirements. Real-time collaboration features similar to Google Slides will enable multiple users to work simultaneously on presentations, incorporating live updates, presence indicators, and version control to enhance teamwork in academic and professional environments. The system's file format compatibility will be extended to include advanced PDF handling with OCR integration for scanned documents, Markdown support for technical documentation, and ODT files to accommodate open-source office users. Additionally, AI-driven design recommendations will automatically suggest appropriate templates, themes, and color palettes based on content context, while generating relevant icons, diagrams, and infographics to enhance visual appeal and reduce manual design effort.

The platform's reach will be significantly expanded through multi-language capabilities and cloud deployment strategies. Multi-language summarization and translation features will enable processing of inputs in various

regional and global languages, producing outputs in users' preferred languages to support cross-cultural communication in academic institutions, multinational corporations, and international conferences. Cloud deployment will provide large-scale scalability through horizontal resource scaling, supporting thousands of concurrent users with features like autoscaling, centralized monitoring, and automated backups. This cloud infrastructure will ensure high availability, simplified maintenance, reduced downtime, and seamless integration with enterprise authentication systems, making the platform suitable for large organizations and global usage while maintaining the security and reliability standards established in the current system

## VIII. CONCLUSION

### *A. Summary of Contributions*

This project introduced an end-to-end system capable of automatically converting raw documents and datasets into structured, professional PowerPoint presentations. By combining natural language processing for summarization, python-pptx for slide generation, and charting libraries for visualization, the tool successfully reduced the manual effort typically involved in presentation creation. Additional contributions include secure authentication through OTP verification and support for multiple file formats, which together make the system practical for academic, research, and professional use.

### *B. Key Findings from Evaluation*

Experimental evaluation highlighted the system's strengths in processing text documents, generating accurate summaries, and producing consistent slide formats. CSV inputs were effectively transformed into visual charts, demonstrating the system's ability to merge textual and graphical content. Load testing confirmed stable operation for up to 20 concurrent sessions, validating the platform's reliability within the current design scope. At the same time, results revealed that scanned PDFs require OCR support and that scalability becomes a limitation beyond the tested threshold, pointing to areas for further improvement.

### *C. Impact on Professional Presentation Creation*

The tool's biggest impact lies in its ability to save time while ensuring consistent quality. Users no longer need to spend hours summarizing, formatting, and designing slides; instead, they can focus on refining ideas and narratives. This benefit is especially significant for students preparing academic projects, researchers documenting their findings, and professionals creating business reports. The automated formatting rules, combined with secure data handling, ensure that outputs are not only efficient but also polished and reliable.

### *D. Final Remarks*

In conclusion, the project demonstrates the feasibility and usefulness of automated presentation generation systems. While current limitations such as OCR dependency and restricted scalability remain, the system already provides substantial value by streamlining the slide creation process. Planned enhancements — including multilingual summarization, AI-assisted design suggestions, and cloud deployment — promise to extend its reach and applicability even further. Overall, the work establishes a strong foundation for future research and development in AI-assisted professional content creation.

## REFERENCES


[1] Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Texts. EMNLP.
[2] See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. ACL.
[3] Liu, Y., & Lapata, M. (2019). Text Summarization with Pretrained Encoders (BERTSUM). ACL.
[4] Lewis, M., et al. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Language Generation. ACL.
[5] Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries.
[6] Documentation for python-pptx, matplotlib, python-docx, PyPDF2, and NLTK.