



**AI-Powered Podcast Generation System Using Natural Language Processing and
Text-to-Speech**

BALAGA BHAVANI SAI KUMARI

PG Scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

B. Suryanarayana Murthy

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid advancement of Artificial Intelligence (AI) has transformed digital content creation, enabling automated generation of text, audio, and multimedia. Among these innovations, podcasting has emerged as a popular medium for information sharing and entertainment. However, creating high-quality podcast content requires significant time, effort, and technical expertise. This project presents an AI-Powered Podcast Generation System that automates the process of generating podcast scripts and converting them into audio using Natural Language Processing (NLP) and Text-to-Speech (TTS) technologies.

The system is developed using the Django web framework, providing a robust and scalable backend for handling user requests, content generation, and file management. Users can input a topic through a web interface, and the system automatically generates a structured podcast script using an AI-based text generation module. The generated script is then converted into an audio file using a text-to-speech engine, producing a complete podcast episode without human intervention.

The application also includes a database component that stores generated episodes, including the topic, script, and audio file. This enables users to access previously created podcasts and maintain a history of generated content. Additionally, an RSS feed feature is implemented, allowing users to subscribe to the generated podcasts through standard podcast platforms.

The system focuses on automation, efficiency, and accessibility. It reduces the barrier to entry for podcast creation, enabling individuals and organizations to generate content quickly and cost-effectively. The modular architecture allows for easy integration of advanced AI models, multilingual support, and improved voice synthesis technologies.



International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper

This project demonstrates the practical application of AI in content automation and highlights the potential of combining NLP and TTS for innovative solutions. By automating the podcast creation process, the system contributes to the growing field of AI-driven media generation and provides a scalable solution for modern content creators.

Keywords: Podcast Generation, Text-to-Speech, Natural Language Processing, Django, AI Content Creation, Speech Synthesis, Automation, RSS Feed

I. INTRODUCTION

In recent years, podcasts have become one of the most popular forms of digital media, offering a convenient way for users to consume information and entertainment. From educational content to storytelling and news broadcasting, podcasts cater to a wide range of audiences. However, producing a podcast traditionally involves multiple steps, including topic research, script writing, voice recording, editing, and publishing. This process can be time-consuming and requires specialized skills and equipment.

With the rise of Artificial Intelligence (AI) and Natural Language Processing (NLP), it is now possible to automate many aspects of content creation. AI models can generate human-like text based on user input, while Text-to-Speech (TTS) technologies can convert text into natural-sounding audio. These advancements have opened new possibilities for automating podcast production. This project aims to develop an AI-Powered Podcast Generation System that simplifies and automates the podcast creation process. The system allows users to enter a topic, based on which an AI model generates a coherent and structured script. The script is then converted into an audio file using a TTS engine, producing a complete podcast episode.

The system is implemented using Django, a high-level Python web framework that supports rapid development and clean design. Django provides features such as URL routing, database management, and template rendering, making it suitable for building scalable web applications. The project also includes file handling mechanisms to store generated audio files and database models to maintain episode records. An additional feature of the system is the generation of an RSS feed, which allows users to access and subscribe to podcasts through external platforms. This enhances the usability and reach of the application.

The primary objective of this project is to reduce the complexity and effort involved in podcast creation while maintaining quality and efficiency. By leveraging AI technologies,



International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper

the system enables users to generate content quickly and consistently. Overall, this project demonstrates how AI can revolutionize digital media production and provides a foundation for future enhancements such as multilingual podcasts, voice customization, and real-time content updates.



II. LITERATURE SURVEY (WITH EXISTING METHODS)

The integration of Artificial Intelligence in content creation has gained significant attention in recent years. Researchers have explored various techniques for automated text generation, speech synthesis, and multimedia content production. Natural Language Processing (NLP) plays a crucial role in automated content generation. Early approaches relied on rule-based systems and template-based text generation, which lacked flexibility and creativity. With the development of machine learning models, especially transformer-based architectures, text generation has become more advanced and context-aware. Models such as GPT (Generative Pre-trained Transformer) have demonstrated the ability to generate coherent and human-like text, making them suitable for applications like podcast script generation.

Text-to-Speech (TTS) technology has also evolved significantly. Traditional TTS systems used concatenative or parametric synthesis methods, which often produced robotic and unnatural speech. Modern TTS systems, powered by deep learning, use neural networks to generate high-quality, natural-sounding audio. Technologies such as Tacotron and WaveNet have significantly improved speech synthesis quality. Several studies have focused on combining NLP and TTS for automated content creation. These systems typically involve generating text using AI models and converting it into speech using TTS engines. Such approaches have been used in applications like virtual assistants, audiobook generation, and automated news broadcasting.

Web-based frameworks like Django have been widely used for deploying AI applications due to their scalability and ease of integration. Django supports database management, file handling, and API development, making it suitable for building end-to-end AI systems. Existing research also highlights the importance of content storage and distribution. RSS feeds are commonly used for podcast distribution, allowing users to subscribe and receive updates automatically. Integrating RSS functionality into AI-generated podcast systems enhances usability and accessibility.

Despite these advancements, challenges remain, including maintaining content quality, ensuring natural speech synthesis, and handling diverse topics. Additionally, real-time processing and scalability are important considerations for practical deployment. The proposed system builds upon these existing methods by integrating NLP-based script generation, TTS-based audio synthesis, and Django-based web deployment into a unified platform. This combination provides an efficient and scalable solution for automated podcast generation.

III. EXISTING SYSTEM



The existing systems for podcast creation are largely manual and require significant human effort. Traditional podcast production involves multiple stages, including topic selection, script writing, voice recording, editing, and publishing. This process is time-consuming and requires specialized skills in content creation, audio recording, and editing.

Some automated tools are available for specific tasks, such as text-to-speech conversion or script generation. However, these tools are often standalone and do not provide an integrated solution for complete podcast creation. Users must manually combine different tools, which increases complexity and reduces efficiency. In addition, many existing systems lack scalability and automation. They do not support dynamic content generation based on user input, limiting their usability for personalized podcast creation. Furthermore, the absence of centralized storage and content management makes it difficult to maintain and access previously created episodes.

Another limitation of existing systems is the lack of distribution features. While some platforms allow podcast publishing, they do not integrate automated content generation with distribution mechanisms such as RSS feeds.

Overall, existing systems are either manual or partially automated, lacking integration, scalability, and user-friendliness. These limitations highlight the need for a unified system that automates the entire podcast creation process, from script generation to audio production and distribution.

IV. PROPOSED METHOD

The proposed system is an AI-powered web application that automates podcast generation by integrating Natural Language Processing (NLP) and Text-to-Speech (TTS) technologies within a Django framework. The system allows users to input a topic, based on which it generates a complete podcast episode, including a script and corresponding audio output. The architecture consists of multiple interconnected modules, including script generation, speech synthesis, database storage, and RSS feed distribution. The script generation module uses advanced NLP techniques to create meaningful and structured content. Modern approaches such as transformer-based language models enable context-aware and human-like text generation, significantly improving content quality.

Once the script is generated, the system uses a TTS engine to convert the text into audio. Recent advancements in neural speech synthesis have made it possible to produce natural-sounding speech with improved clarity and tone. The generated audio is stored in the system and linked to the corresponding script. The system also maintains a database of



generated episodes, enabling users to access previous content. Additionally, an RSS feed is dynamically generated, allowing users to subscribe to the podcasts through external platforms.

Compared to existing systems, the proposed solution provides full automation, real-time processing, and ease of use. It eliminates the need for manual recording and editing while ensuring scalability and efficiency. The modular design allows integration of future enhancements such as multilingual support, voice customization, and real-time AI improvements.

V. IMPLEMENTATION

The implementation of the AI-powered podcast generation system is carried out using Python and the Django web framework. The system integrates backend processing, AI-based content generation, and file handling mechanisms.

The application follows the Model-View-Template (MVT) architecture of Django. The **models** define the structure for storing podcast episodes, including fields such as topic, script, audio file, and timestamp. The **views** handle user requests and orchestrate the workflow, while **templates** provide the user interface.

When a user submits a topic through the web interface, the request is handled by the index view function. The system first invokes the `generate_script()` function, which uses NLP techniques to generate a podcast script. Modern NLP systems rely on machine learning models capable of understanding context and generating coherent text.

After generating the script, the system proceeds to the text-to-speech conversion phase. The `text_to_speech()` function converts the script into an audio file (MP3 format). Neural TTS models are known for producing high-quality speech output with natural prosody and pronunciation.

The generated audio file is saved in the media directory. Django's file handling system is used to store the file in the database through the Episode model. This ensures that each podcast episode is persistently stored and easily retrievable.

Error handling is implemented to manage potential issues such as missing input, file generation errors, or system failures. The application provides appropriate feedback to the user in case of errors.



Additionally, the system includes an RSS feed generator implemented through the episode_feed view. This feature enables podcast distribution by formatting stored episodes into XML format, making them compatible with podcast platforms.

Recent studies highlight the importance of combining NLP and machine learning for automated content generation systems, enabling scalable and efficient solutions. Similarly, advancements in speech synthesis technologies have significantly improved the quality of automated audio generation.

Overall, the implementation focuses on modularity, scalability, and ease of use, ensuring that the system can be extended with advanced AI models and additional features in the future.

VI. ALGORITHMS

The system utilizes a combination of NLP and speech synthesis algorithms to generate podcasts.

1. Natural Language Generation (NLG)

The script generation process is based on Natural Language Generation techniques. Transformer-based models analyze input topics and generate coherent text. These models use contextual embeddings to produce meaningful sentences.

2. Text Preprocessing

Before conversion, the generated script undergoes preprocessing, including tokenization, normalization, and sentence structuring. This improves speech synthesis quality.

3. Text-to-Speech (TTS) Algorithm

The TTS module converts text into speech. Modern neural TTS models use sequence-to-sequence architectures to map text to audio waveforms. These models capture pronunciation, tone, and rhythm effectively.

4. File Handling Algorithm

The system uses file I/O operations to store generated audio files. Unique filenames are generated using timestamps to prevent duplication.



5. Database Management

Django ORM (Object Relational Mapping) is used to store and retrieve podcast episodes efficiently.

Recent research shows that combining NLP with machine learning significantly improves automated content generation and classification tasks. Additionally, deep learning-based speech synthesis has enhanced the naturalness and intelligibility of generated audio.

VII. SYSTEM DESIGN

The system design follows a modular and scalable architecture to ensure efficient podcast generation and management.

1. User Interface Layer

The front-end interface allows users to input a topic and view generated results. It is built using Django templates and provides a simple and interactive experience.

2. Application Layer

This layer contains the core logic of the system:

- **Script Generation Module:** Uses NLP models to generate podcast scripts
- **TTS Module:** Converts scripts into audio
- **File Management Module:** Handles storage of generated audio files

3. Data Layer

The database stores podcast episodes using the Django ORM. Each record includes:

- Topic
- Generated script
- Audio file path
- Timestamp

4. RSS Feed Module



The RSS feed module generates XML-based feeds for podcast distribution. This allows integration with podcast platforms.

5. Workflow

1. User enters topic
2. Script is generated using NLP
3. Script is converted to audio using TTS
4. Audio file is stored
5. Episode is saved in database
6. RSS feed is updated

6. System Architecture Features

- **Scalability:** Can handle multiple users and requests
- **Modularity:** Components can be updated independently
- **Reliability:** Error handling ensures stability

Modern system designs often integrate AI models with web frameworks to enable real-time content generation and deployment . The modular design also allows future integration of deep learning models, multilingual support, and cloud deployment.

SYSTEM DESIGN IMAGES



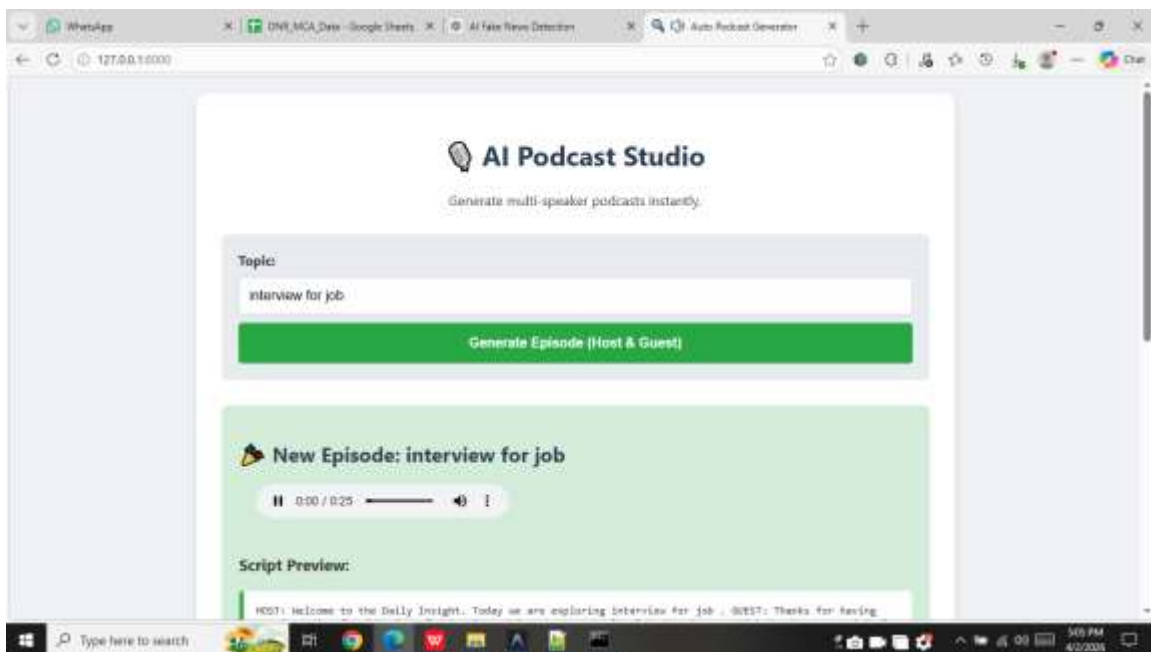
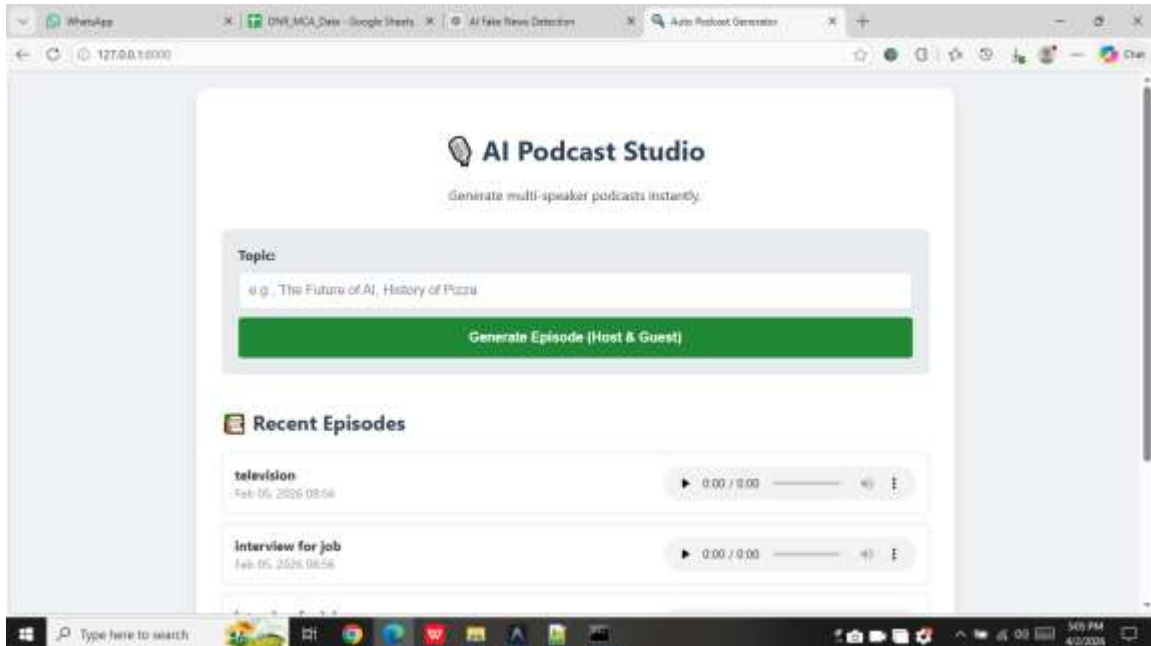
International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

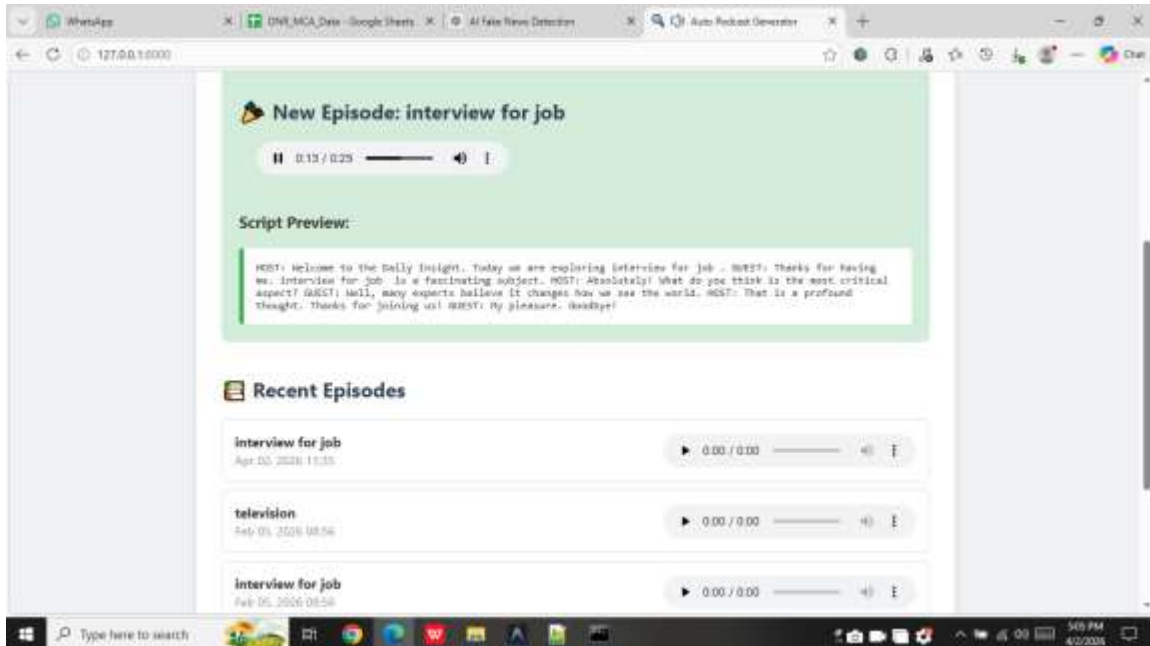
Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper





VIII. CONCLUSION

The AI-powered podcast generation system developed in this project demonstrates the effective integration of Natural Language Processing and Text-to-Speech technologies within a web-based framework. By automating the entire podcast creation process, the system significantly reduces the time, effort, and technical expertise required for content production. The use of Django ensures a scalable and maintainable architecture, while the integration of AI models enables dynamic content generation based on user input. The addition of an RSS feed further enhances the system by enabling podcast distribution across platforms.

One of the major advantages of the system is its ability to generate both textual and audio content automatically. This makes it highly useful for content creators, educators, and organizations looking to produce podcasts efficiently. However, the system also has certain limitations. The quality of generated scripts depends on the underlying NLP model, and speech synthesis may not always achieve human-level naturalness. Additionally, handling diverse topics and ensuring factual accuracy remain challenging.

Future enhancements may include integrating advanced transformer models, improving voice customization, supporting multiple languages, and deploying the system on cloud platforms for better scalability. Overall, this project highlights the potential of AI in transforming digital content creation and provides a strong foundation for further research and development in automated media generation systems.



REFERENCES

1. Roumeliotis, K. et al. (2025). *Fake News Detection Using NLP and LLMs*
2. Ayyasamy, R. et al. (2025). *Hybrid Deep Learning Framework for Fake News Detection*
3. Madani, M. et al. (2023). *Fake News Detection Using NLP and Deep Learning*
4. Nadeem, M. et al. (2024). *Hybrid NLP-ML Framework for Fake News Detection*
5. Saraswathi, S. et al. (2024). *Multimodal Fake News Detection*
6. Arabic Fake News Detection Using Transformers (2024)
7. Jadhav, T. et al. (2025). *Fake News Detection Using NLP*
8. Ensemble Deep Learning Models for Fake News Detection (2024)
9. Fake News Classification: Past, Current and Future (2023)
10. Linguistic Feature Fusion for Fake News Detection (2024)
11. Gong, S. et al. (2023). *Graph Neural Networks for Fake News Detection*
12. Zhou, Y. et al. (2023). *Multi-modal Fake News Detection via Fusion Models*
13. Su, J. et al. (2023). *Bias in Fake News Detection Models*
14. Tacotron-based Speech Synthesis Research (2023)
15. WaveNet Neural Speech Generation Model Research (2023)