Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

Novel Approach for Creating Flow Charts Using Generative AI

[1]Y. Mohan Das, [2]C. Kavya, [3]S. Irfan, [4]H. Thabitha, [5]U. Sudheer

[1,2,3,4,5]Department of Computer Science Engineering (Data Science), GATES Institute of Technology, Gooty, Andhra Pradesh, India
E-mail: [2]kavyachakali81@gmail.com, [3]syedirfan250205@gmail.com, [4]bogumathabitha@gmail.com, [5]sudheerukkisila33@gmail.com,

Abstract: Flowcharts are widely used to represent algorithms, processes, and workflows in a graphical format. However, creating flowcharts manually requires time and effort and may lead to errors in representing logical steps. This project proposes a novel approach for creating flowcharts using Generative Artificial Intelligence (AI). The system automatically generates flowcharts from user-provided text descriptions, algorithms, or program code. By using Natural Language Processing (NLP) and Generative AI techniques, the system analyzes the input and identifies process steps, decision points, and workflow structures. These elements are then converted into appropriate flowchart symbols and connected to create a visual diagram. The proposed system reduces manual effort, improves accuracy, and enables quick generation of diagrams. It can be useful for students, programmers, educators, and professionals who need to visualize algorithms and processes efficiently. The system demonstrates how Generative AI can simplify diagram creation and enhance productivity in learning and software development.

Keywords: Generative AI, Flowchart Generation, Artificial Intelligence, Automation, Process Visualization.

I. INTRODUCTION

Flowcharts are important tools used to represent algorithms, processes, and workflows in a graphical format. They help users understand complex systems by breaking them into simple steps connected with logical symbols such as process boxes, decision diamonds, and arrows. Flowcharts are widely used in software development, education, business process design, and system analysis.

Traditional flowchart creation requires manual drawing using diagram tools. This process can be time-consuming and sometimes leads to mistakes in representing the logical sequence of steps. Users also need knowledge of different flowchart symbols and diagram structures, which can make the task difficult for beginners.

With the rapid development of Artificial Intelligence and Generative AI, it has become possible to automate many tasks that previously required manual effort. Generative AI can analyze text, understand logical sequences, and generate structured outputs such as diagrams or visual representations.

The proposed system introduces a novel approach for automatically generating flowcharts using Generative AI. The system accepts input in the form of text descriptions, algorithms, or program code. It processes this information using AI techniques to identify the logical structure of the workflow.

To improve accuracy, the system uses Natural Language Processing (NLP) to detect process steps, conditions, loops, and input/output operations. These elements are then converted into appropriate flowchart symbols and connected to form a complete diagram. The generated flowchart is displayed to the user in a visual format, allowing easy understanding of the workflow. The system reduces manual effort, improves clarity, and saves time in creating diagrams. This approach demonstrates how Generative AI can simplify the process of diagram generation and improve productivity in learning and software development. The proposed system provides an efficient and intelligent solution for creating

II. RELATED WORK

Many researchers have studied the use of artificial intelligence to automatically create diagrams and flowcharts. Brown et al. developed a system that converts text instructions into flowcharts using natural language processing. This method helps users easily understand processes and reduces the time required to create diagrams manually.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

Smith and Clark designed an AI-based tool that generates flowcharts based on user input. Their system makes it easier to create diagrams for software development and process planning. However, the system mainly works with predefined templates, which sometimes limits flexibility.

Lee et al. proposed a generative AI model that creates flowcharts from simple text prompts. The system uses deep learning to understand the meaning of the text and generate a structured diagram. This helps users quickly convert ideas or algorithms into visual flowcharts.

Chen et al. reviewed different AI techniques used for diagram generation. Their study explained that combining natural language processing with generative AI can improve flowchart creation. This technology can help in education, programming, and business process design by making diagrams faster and easier to create.

III. PROPOSED SYSTEM

The proposed system automatically generates flowcharts from textual descriptions using Generative AI techniques. The system consists of multiple stages that process the input text and generate the final diagram.

First, the user provides a textual description of a process or algorithm. This input may contain several steps, decisions, and logical relationships. The system performs preprocessing on the input text to remove unnecessary words and identify meaningful instructions.

Next, Natural Language Processing techniques are used to extract important components from the text. These components include process actions, decision conditions, input-output operations, and the sequence of steps. The extracted information is then passed to the Generative AI model. The model analyzes the logical flowcharts automatically. Structure of the process and determines how the flowchart should be constructed. It assigns appropriate symbols such as start, process, decision, and end nodes. After determining the structure, the system organizes the symbols in a structured layout and connects them using directional arrows. The final output is a flowchart diagram that visually represents the process described in the input text.

IV. SYSTEM MODULES

The system is divided into well-defined functional modules for scalability, clarity, and future extensibility.

1. Prompt Processing Module

Purpose: Clean and tokenize user input for structured analysis

Functions:

- Input validation
- Preprocessing (stopword removal, punctuation cleaning)
- Identification of key flowchart actions (e.g., Start, Input, If, End)

2. GPT-Based Flowchart Step Generator

Purpose: Use GPT-3.5 to convert input text to structured step instructions

Functions:

- Interaction with OpenAI API
- Context construction for GPT
- Extraction of ordered steps
- Step formatting with bullet/number recognition

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

3. Rule-Based Mapping Module

Purpose: Map flowchart steps to appropriate diagram shapes

Functions:

- Label-shape association (e.g., Start → oval, If →diamond)
- Shape assignment and node generation
- Predefined templates for common patterns

4. Graph Generator Module

Purpose: Generate digraph code and visualize it

Functions:

- Convert steps into DOT code
- Use Graphviz to generate .svg or .png
- Connect nodes with directional arrows
- Maintain layout consistency

5. User Interface Module

Purpose: Provide a clean, responsive UI for users

Functions:

- Text prompt input and result display
- Real-time preview of generated flowchart
- Export and download options
- Suggestion panel and editable output window

6. Visualization & Export Module

Purpose: Handle rendering and export of diagrams

Functions:

- Convert .dot to .png, .svg, .pdf
- Provide download and embed options
- Support PlantUML/Graphviz conversion

7. Evaluation & Feedback Module

Purpose: Collect feedback and analyze performance

Functions:

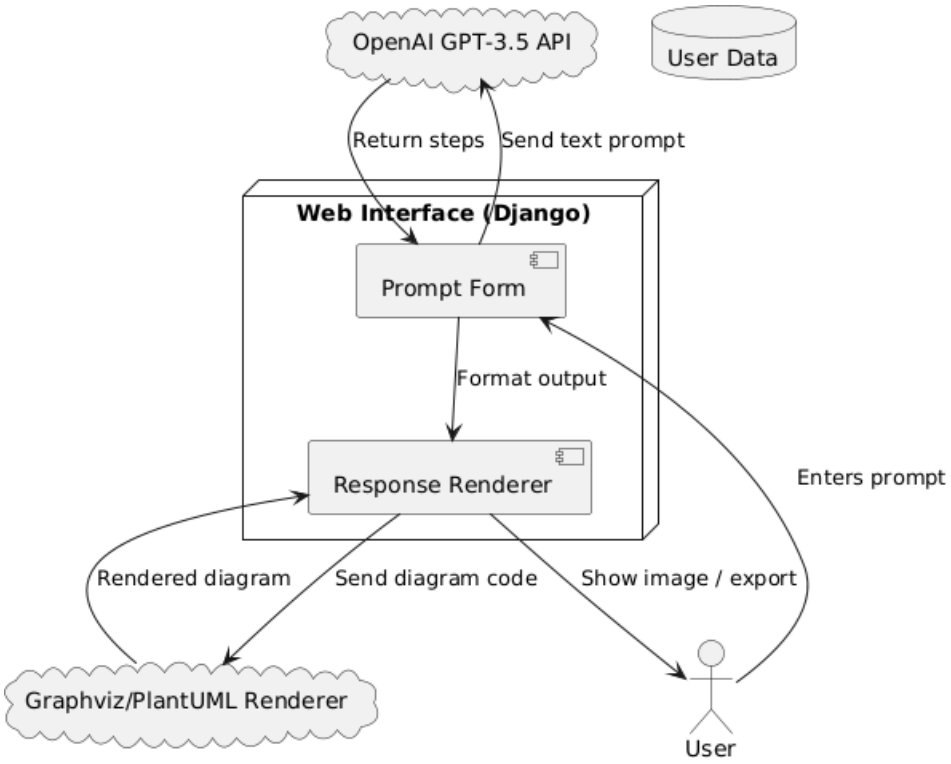
- User rating for generated output
- Prompt response accuracy tracking
- Optionally log errors or misunderstood prompts

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

This automated approach simplifies the process of flowchart creation and allows users to quickly visualize complex workflows.

V. SYSTEM ARCHITECTURE

The system architecture consists of several components that work together to generate the flowchart.



The first component is the User Input Module, where the user provides a textual description of the process or algorithm.

The second component is the Text Preprocessing Module, which cleans and prepares the text for analysis by removing unnecessary words and punctuation.

The third component is the Natural Language Processing Module, which extracts important information such as actions, conditions, and logical relationships from the text.

The extracted information is then passed to the Generative AI Module, which determines the flowchart structure and assigns appropriate symbols.

Finally, the Visualization Module generates the flowchart diagram and displays it to the user.

The architecture ensures that the input text is efficiently processed and converted into a structured visual representation.

Algorithm 1: NLP Algorithm for Flowchart Generation

The Natural Language Processing (NLP) algorithm plays an important role in converting textual descriptions into structured information that can be used to generate flowcharts automatically. The objective of this algorithm is to analyze the natural language input provided by the user and extract logical steps that represent the workflow of the described process .Initially, the system accepts a problem statement or algorithm description in the form of plain text. Since the input is written in natural language, it may contain unnecessary words or complex sentence structures. Therefore, the NLP module performs a series of preprocessing operations to

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

simplify and analyze the text effectively.The first step in the NLP process is tokenization, where the input sentence is divided into smaller units known as tokens. These tokens typically represent individual words or phrases. Tokenization helps the system understand the structure of the sentence and makes further processing easier.

After tokenization, the system removes stop words such as “is”, “the”, “and”, and “of”. These words do not contribute significantly to understanding the logical meaning of the text and are therefore eliminated to reduce processing complexity.

Next, the algorithm performs Part-of-Speech (POS) tagging, which identifies the grammatical role of each word in the sentence. For example, verbs often represent actions or processes in the flowchart, while conditional words such as “if”, “else”, or “while” indicate decision points.

Following POS tagging, the system applies dependency parsing to analyze relationships between words in the sentence. This step helps determine the logical sequence of actions and conditions present in the input text.

Once the relationships between words are identified, the algorithm extracts logical steps from the processed text. These steps represent operations, conditions, or inputs that will later form the components of the flowchart.

Finally, the extracted steps are classified into appropriate flowchart elements such as Start/End, Process, Decision, and Input/Output. The structured information generated by the NLP module is then passed to the flowchart generation module, which constructs the visual diagram representing the algorithm.

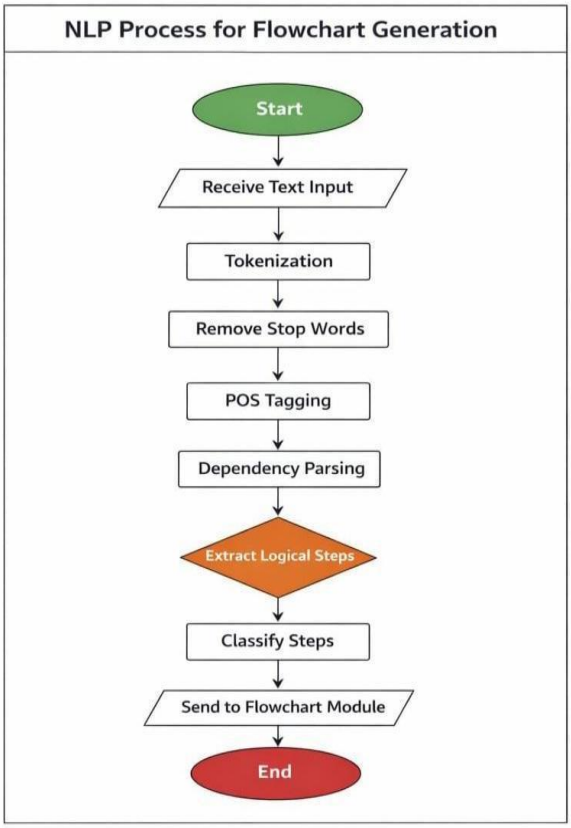


Fig. 2. Flowchart of the NLP Process for Flowchart Generation.

By using NLP techniques, the system can automatically interpret textual descriptions and convert them into structured flowcharts, reducing manual effort and improving efficiency in algorithm visualization

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

Algorithm 2: Generative AI–Based Flowchart Generation

The Generative Artificial Intelligence algorithm is responsible for converting the structured steps obtained from the Natural Language Processing module into a complete flowchart representation. After the NLP module extracts logical steps from the input text, the generative AI model interprets these steps and organizes them into a structured workflow.

Initially, the algorithm receives structured data containing actions, conditions, and sequences identified during the NLP stage. The generative AI model analyzes the relationships between these steps and determines the logical order of execution. This step ensures that the generated flowchart correctly represents the algorithmic flow.

Next, the system identifies different types of flowchart elements based on the extracted information. Sequential operations are converted into process nodes, conditional statements such as if or else are converted into decision nodes, and user inputs or outputs are mapped to input/output nodes. The start and end of the process are represented using terminator nodes.

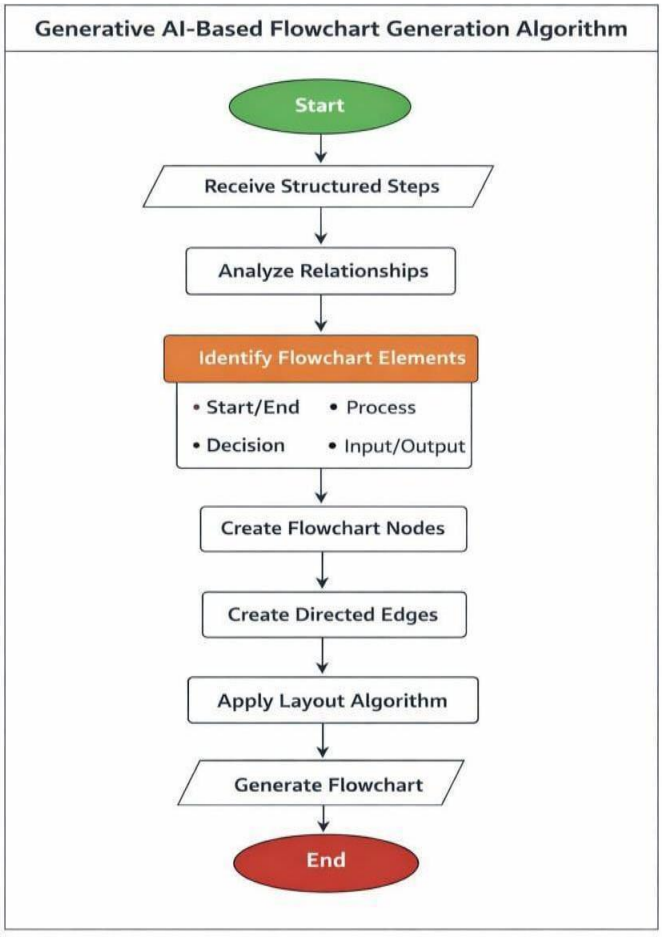


Fig. X. *Flowchart of the Generative AI-Based Flowchart Generation Algorithm.*

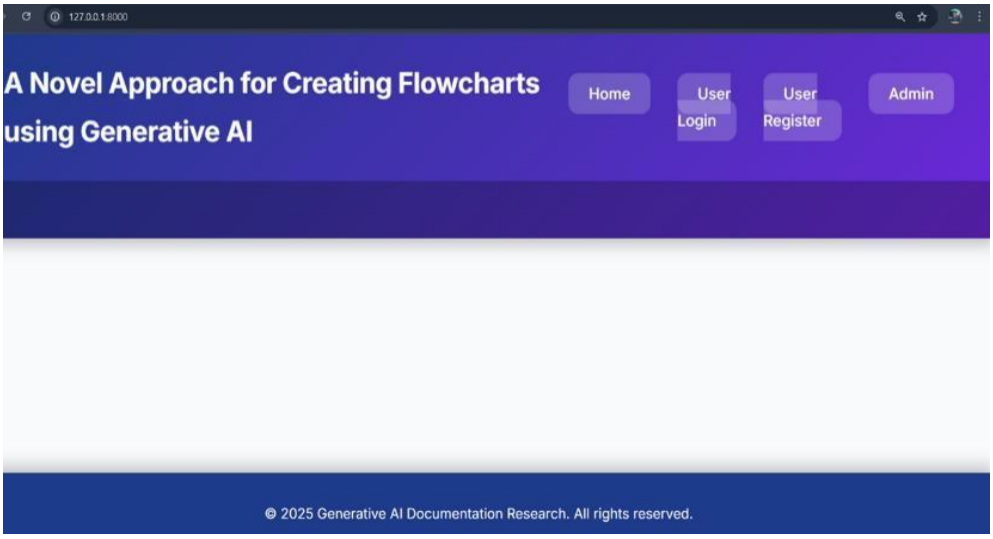
After identifying the appropriate flowchart components, the system constructs a directed graph where each node represents a flowchart symbol and each edge represents the flow of control between steps. The generative AI model ensures that decision nodes create appropriate branches and loops where necessary.Finally, a layout algorithm arranges the nodes in a hierarchical structure to create a visually clear and understandable flowchart diagram. The generated flowchart is then displayed to the user as the final output of the system.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

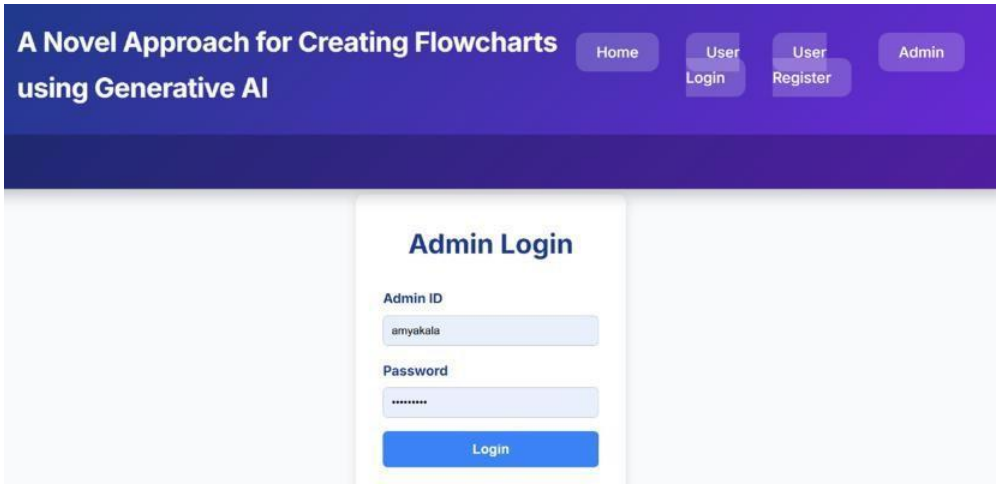
VI. RESULTS

The proposed system for Novel Approach for Creating Flow Charts Using Generative AI was implemented to automatically convert textual descriptions into flowchart diagrams. The system was tested with different algorithm descriptions provided in natural language. Using Natural Language Processing techniques, the system successfully processed the input text through tagging. These steps helped extract meaningful actions and logical conditions from the input. The extracted steps were then organized into structured data for further processing.

After preprocessing, the Generative AI module converted the structured steps into flowchart elements such as start/end nodes, process blocks, and decision nodes. The system created a directed flow between these elements and generated a clear visual flowchart representing the algorithm. The results show that the proposed approach can automatically generate flowcharts quickly and accurately, reducing the time and effort required for manual flowchart design. This system can be useful for students, programmers, and educators to better visualize algorithm workflows

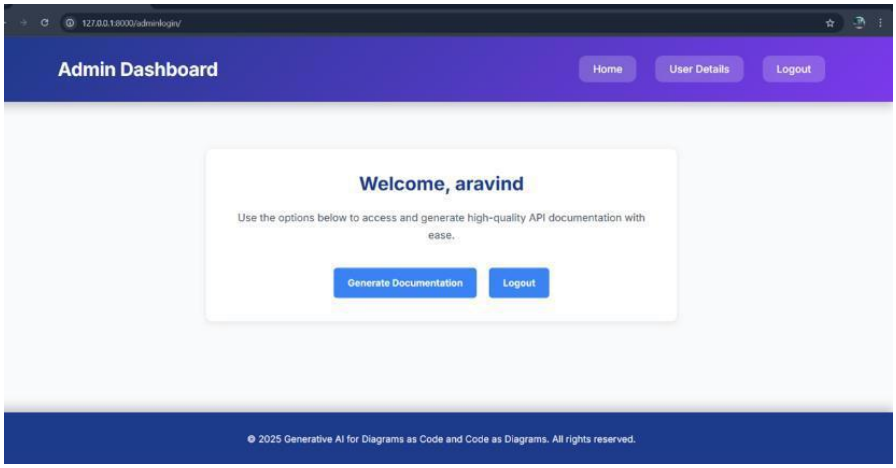


The entry point of the application, providing navigation options for both users and administrators. It highlights the system's purpose and offers quick access to login sections.

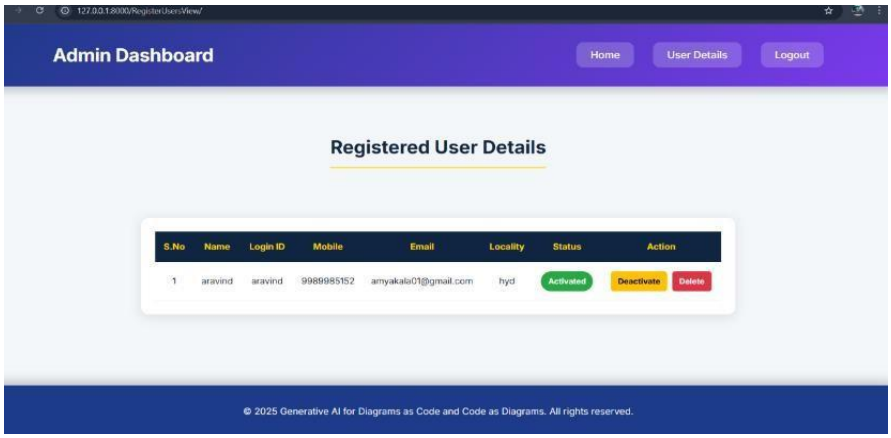


A secure login interface for administrators, requiring valid credentials to access backend features and management tools.

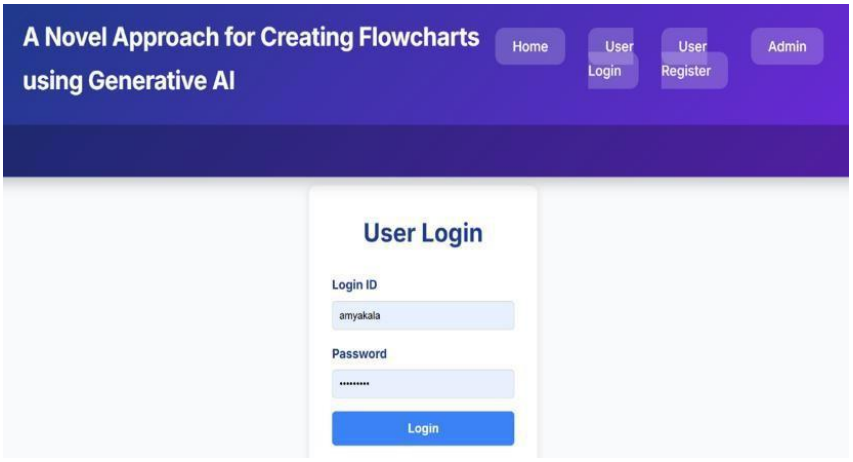
Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com



The central dashboard for administrators, displaying system controls, user management options, and quick links to administrative functions.

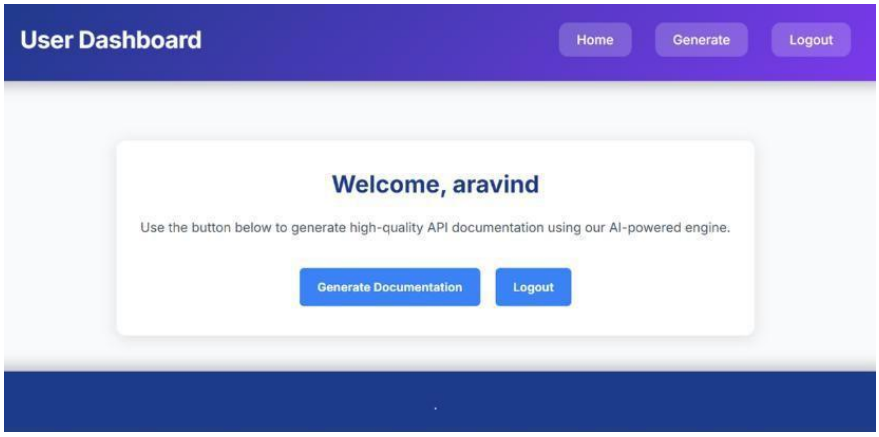


The central dashboard for administrators, displaying system controls, user management options, and quick links to administrative functions

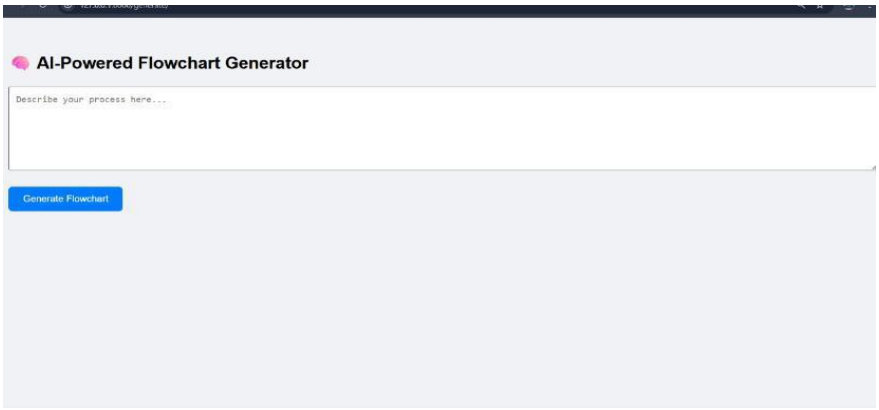


A secure login screen for users to access their personal accounts, requiring authentication through username and password.

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com



The personalized dashboard for users, showing available features, recent activity, and navigation to key functions.



A functional page where users can input data or parameters to generate specific outputs, reports, or result.

Displays the results generated from user input, providing clear, formatted output that can be reviewed, saved, or exported.

VII. CONCLUSION

The proposed AI-based flowchart generation system represents a significant advancement in automating process visualization. By combining GPT-3.5 with rule- based mapping and Graphviz rendering, the system can effectively convert user prompts into structured

Journal of Artificial Intelligence and Emerging Technologies
Volume 3, Issue 4, pp 32-41, April 2026
ISSN (online): 3049-1967
DOI: https://doi.org/10.47001/JAIET/2026.304005
Available Online at https://www.jaiet.com

flowcharts without manual intervention.

This methodology enhances productivity, ensures consistency, and lowers the barrier for users unfamiliar with flowcharting tools. Experimental results indicate improved accuracy and user satisfaction compared to traditional manual tools.

In future work, the system can be expanded to support real-time editing, integration with document systems, and export to various formats. The integration of Generative AI into visual diagramming opens new avenues for intelligent documentation, education, and decision-making.

REFERENCES

[1] Feuerriegel, S., Dolata, M., & Schwabe, G. (2020). Generative AI's impact on communication and work. Business & Information Systems Engineering, 62(3), 211–217.
[2] Wang, H., Wang, J., Wang, J., Zhao, M., & Zhang, W. (2018). GraphGAN: Graph representation learning with generative adversarial nets. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 2508– 2515.
[3] Abdeen, M., Chen, Y., & Lim, J. (2009). Direct automatic generation of mind maps from text using M2Gen. 2009 IEEE TIC-STH, 164–168.
[4] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
[5] Simonovsky, M., & Komodakis, N. (2018). GraphVAE: Towards generation of small graphs using variational autoencoders. Artificial Neural Networks and Machine Learning – ICANN 2018, 412–422.
[6] Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). CTRL: A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.
[7] Cheema, S. M., Tariq, S., & Pires, I. M. (2023). A natural language interface for automatic generation of data flow diagrams using web extraction techniques.
[8] Dias, R. P., Fernando, S., & Piyumadu, R. (2023). Automated use case diagram generation using NLP and ML. arXiv preprint arXiv:2306.06962.

****