International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Automation Builder: Natural Language-Driven Workflow Generation Using Large Language Model and N8N

inothraj P, Prof. Mrs. X. Anitha Sarafin, Prof. Dr. A Geethapriya, Alagumalai N, Mukesh Jain M.D
Department of Computer Science and Engineering
Misrimal Navajee Munoth Jain Engineering College,
Chennai, Tamil Nadu – 600097

Abstract **- Among many workflow automation tools like n8n, Zapier, and make.com helps to connect different type of application and automate task to increase productivity. To create these automation needs technical knowledge, manual configuration and need to understand complex platform logic, so many non-technical user feels hard to use . To solve this problem, this paper presents a natural language-driven workflow generation system using large language models and the n8n platform. This system allows users to describe their requirements using simple English. This system uses the natural language processing ability of large language models to identify user intent, triggers, actions from the given text. These extracted statements are then converted to the json format that is required in n8n. This system automatically checks for errors and validates the workflow before deploying in n8n enabling minimal-effort automation process. This system architecture includes a React based user interface, Node.js backend server, LLM-based NLP and a Docker-based n8n engine make the result easier and more scalable. Experimental results show that this approach reduces the time that is needed to create the workflows and lowers the technical difficulty of using automation platforms compared to manual workflow creation methods. Hence the results says that natural language-based workflow automation can make automation accessible to wide range of users and can applied I different areas such as business process automation, data handling, and productivity involvement. This system also provides a strong base for future enhancements such as multi-language support and self-improving workflows.**

Keywords - Natural language processing, Large Language Models, Workflow Automation, Natural Language-Driven Systems, Minimal Automation, Intelligent Automation, N8N Platform, AI-Based Workflow Generation.

I. INTRODUCTION

Automation has become an essential component in modern software systems, which helps organizations to reduce the repeated manual work, It also improves efficiency and ensures consistent task execution. Workflow automation platforms like n8n, Zapier, and Make.com help users to connect different applications and automate different tasks along with the notifications, data synchronization, document processing and scheduled operations.

These tools have significantly improved productivity across all the domains in this modern technology. Even being promoted as low-code or no-code solutions, most automation platforms still require technical knowledge. Which was the user must manually need to configure triggers, actions, data mapping, dependencies, and error-handling logic. While workflow has more complexity then this becomes more difficult to design, maintain and debug. As a result, non-technical users often struggle to fully adopt automation tools.

In real-world Scenarios, user understand what they need to automate but feel it is hard to transform their requirements into specific workflows. Although users can easily describe what they want to automate in plain language, building the actual workflow requires technical knowledge. This gap between the user and the system design is the big problem of current automation platforms

Recent advances in AI, Especially in natural language processing, made it easier to solve this problem. Large Language Models can understand human language, identify the intention of the user and produce good quality outputs. Because of this , they are well suited for automating tasks that usually need manual setup and technical skills, experience and knowledge.

This paper proposes a natural language-driven workflow generation system using large language models and the n8n automation platform. The proposed system allow users to create complete automation workflows by describing their requirements in simple English. LLM are used to extract the triggers, actions, conditions, and dependencies from the user input, then it is converted into a structured JSON workflow compatible with n8n.

This system mainly focuses on being reliable, accurate and easy to use. The workflows are created by the system are checked before setting up properly. After validation, the workflows are automatically deployed in n8n environment without any extra manual work from the user.

Experimented evaluation says that this approach reduces the workflow creating time and configuration errors compared to manual methods. This also lowers the technical barrier, The system makes workflow automation more accessible to non-technical users and supports a wide range of real-world automation scenarios.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Overall, this work shows that integrating large language models with automation platforms can greatly improve efficiency, accessibility and easy to use by making automation system more intelligent and user friendly.

II. LITERATURE REVIEW

In recent years, workflow automation systems have grown rapidly to help improve efficiency in business, data processing and enterprise tasks. Traditional automation platforms such as Zapier, Make.com, and n8n provide rule-based development mechanisms that allow users to connect multiple applications through triggers and actions. While these automation platforms reduce the need for manual programming, but also it still requires users to have the strong understanding on workflow logic, data mapping, and platform configuration, which reduces the adaptability among non-technical users.

[1] Several studies have explored low-code and no-code automation platforms as a solution to reduce software development complexity. These platforms rely on predefined components to simplify workflow creation. However, the authors highlight that when workflows become more complex, users find it harder to understand, make errors, and have trouble maintaining them. The study also shows that low code often move the complexity instead of removing it, especially for users without technical experience.

[2] Research on template-based workflow automation systems propose the use of reusable automation template to increase the workflow creation. While templates helps in reducing the initial setup time but they suffer from limited flexibility and low adaptable for the dynamic requirements. The authors conclude that template-driven system fail to support personalized automation scenarios and require manual intervention for customization, which restricts scalability.

[3] Modern advancement in natural language processing, researchers have investigated using natural language interface to make automation and software control are easier for users. Early NLP based systems focused on keyword matching and intent classification using rule-based techniques. Although these approach allows only basic command based automation, they had a difficulty in understanding unclear instruction, handling multi-step tasks. The study clearly shows that traditional NLP methods are not effective for managing complex automation logic.

[4] The introduction of large language models has greatly improved the ability of systems to understand natural language and user intention. Recent works explains about the effectiveness of LLMs in interpreting complex instructions, generating structured representation, and assigning a task and code generation. The author says that LLMs perform better than traditional NLP models in understanding commands with handling multiple tasks and remembering context in long instructions. However, most of these studies focus on conversational assistants and code generation rather than direct workflow automation.

[5] Several AI-assisted automation system integrated chatbots with workflow platforms to guide users during workflow creation. These system provide step-by-step recommendations and validation feedback while users manually configure workflows. Although such approaches reduce user errors, they still require repeated user interaction and do not eliminate manual workflow design. The authors says that this type of systems improve usability but stop easy achieving full automation.

[6] Research involves chat-based workflow execution systems, particularly by integrating messaging platforms such as telegram or stack with automation engines like n8n, focuses primarily on triggering and monitoring workflows through conversational interfaces. In these systems, workflows are pre-build manually, and natural language is used only for execution control rather than workflow generation. This limits their contribution to workflow generation automation.

[7] Studies on JSON-based workflow representation explain how automation platforms use structured configuration files to manage and run workflows internally. These works highlight the importance of schema validation, dependency management and error handling for reliable automation execution. However they rely on structured or semi-structured inputs and do not address the challenge of generating workflow definitions directly from free-form natural language instruction.

[8] Recent survey on intelligent automation and AI-driven workflow systems explained the need for automation platforms that can adapt to user intention dynamically. The authors identify a research gap in systems that can automatically translate human language into executable workflows without intermediate manual steps. They also have the importance for validation, explainability and scalability in real-world automation deployments.

Even though automation tools, NLP interface, and AI-based systems have improved a lot, existing research shows that there are very few solutions that can fully create workflows from natural language and deploy them automatically. Most current approaches depend on predefined templates, need user to guide each step, or focus only on running workflows instead of creating them. In addition, very little work has been done on combining LLM-based intent understanding with real automation platforms like n8n in a scalable and practical way.

This works solves these problems by introducing a system that creates workflows from plain English using Large Language Models and n8n. Users only need to describe what they want to automate, and the system automatically builds and deploys a working workflow without manual setup. This approach moves automation closer to being fully intelligent and user friendly.

III. PROPOSED SOLUTION

A. Natural Language-Driven Workflow Generation approach

- This approach introduces a natural language-centric automation paradigm that allows users to define complex workflows using plain English instructions. Unlike existing workflow automation platforms that require manual configuration of triggers, actions, and condition logic, the proposed approach focuses on intent understanding and logical reasoning rather than user-driven configuration.

- The main idea of this solution is that workflow automation can be treated as a process of converting user intent into executable logic, where human instructions are transformed into structured steps. This change enables non-technical users to express automation requirements without knowledge of platform-specific syntax or workflow deign principles. This solution operates in three conceptual phases,

1.Intent Interpretation: User instructions are treated as the high-level goals. The system identifies the main purpose and identifies the needed triggers, actions, conditions and order of steps by understanding the meaning of the sentence not just matching keywords.

2. Logical Workflow Reasoning: This identified intent is converted into a general workflow plan. This plan explains what actions should happen and in what order, without depending on any specific automation tool. The system then organizes task dependencies, step order, conditions, and data flow between actions.

3. Executable Workflow Synthesis: The general workflow plan is converted into a format that the automation system can run. This keeps the user's original intent clear while allowing the workflow to run properly

This method reduces user effort, avoids setup mistakes, and supports complex multi-step tasks that are hard to build using traditional visual editors.

IV. SYSTEM ARCHITECTURE

A. System Overview

The System architecture is designed based on completely natural language-driven workflow automation, transforming user instructions into the fully functioning workflows with minima human intervention. This ensure scalability, extensibility and reliable execution

Automating process consists of five stages:

- Stage 1: User Interaction
- Stage 2: Natural language Processing
- Stage 3: Workflow Generation
- Stage 4: Automation Execution
- Stage 5: Monitoring and Feedback

Each stage works independently and shares information using structural data formats. This allows easy updates and smooth connection with other external services.

B. User Interaction Stage

The user interaction layer servers as the entry point to the system. It act as the interface through which users submit automation instructions using plain English. This interface can be implemented as the web-based client, REST API endpoint, or chat-style interface.

Key responsibilities of this layer include:

- Collects user instructions written in plain English.
- Checks and formats the input correctly
- Sends the request safely to the backend system.

This layer hides all technical details from the user and makes the automation process simple and easy to use.

C. Natural Language Processing Stage

The Natural Language Processing layer understands the user's instructions and converts them into a structure format. It uses Large Language Models to understand the meaning of the text instead of depending on fixed rules. The NLP layer performs the following functions:

- Extracts the user's main goal from the input,
- Identifies triggers, actions, conditions, and limits
- Clarifies unclear or implied workflow details

The result of this stage is a structured format that machine can understand, which is then sent to the workflow generation layer for the next step.

D. Workflow Generation Stage

This stage converts the structured intent representation into a workflow format that the automation platform can understand, It act as a bridge between what the user wants and how the system executes it.

Its responsibilities include:

- Mapping a workflow steps into automation action.
- Setting the correct order and dependencies
- Creating the workflow in the standard JSON format
- Checking the workflow is logically correct and complete

This layer makes sure that the generated workflow follows system rules and still keeps the original meaning of the user's request.

E. Automation Execution Stage

This stage is responsible for deploying and running workflows on the automation engine. In this system, n8n is used as the execution platform due to its flexibility, extensibility, and support for diverse integrations.

Key functions for this layer include:

- Deploys workflows automatically using REST APIs
- Activates triggers and handles events
- Runs connected services such as email, databases, cloud storage, and messaging apps.
- Manages errors and retries failed tasks when needed.

This layer makes sure the workflows run properly and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

reliably in real-world situations.

F. Monitoring and Feedback Stage:

This stage provides cisibility into workflow execution and system performance. It enables users and administrators to track workflow status, execution logs, and failure events.

Major feature include:

- Real-time execution status monitoring.
- Error reporting and diagnostic logs.
- Feedback mechanisms to refine workflow generation accuracy.

This supports continuous improvement by enabling iterative refinement of automation logic based on execution outcomes.

G. Data Flow Description

The end-to-end data flow of the system proceeds as follows:

1. The user submits the natural language instruction through the interface
2. This instruction is processed by NLP layer to extract intent and workflow components.
3. Then the structured intent is converted into a deployable workflow definition.
4. The automation execution layer deploys and execute the workflow on n8n.
5. Execution results and logs are captured by the monitoring layer and presented to the user.

This sequential flow ensures consistency, traceability, and reliability throughout the automation lifecycle.

H. Technology Stack

The complete system is implemented by technologies:

- Frontend/Interface: React.js and web-based UI
- Backend Services: Node.js with Express.js
- Natural Language Processing: Large Language Models accessed via API.
- Workflow Engine: n8n automation platform.
- Data Format: JSON for workflow definitions and API communication.
- Deployment Environment: Local or Clou-based server infrastructure.

I. Architectural Benefits

The proposed architecture offers several advantages:

Modularity: Each and every part of the system update without affecting the others.

Scalability: The system can handle more users and more workflows as it grows

Extensibility: New services and integration can be added easily.

Reliability: Automatic checking and monitoring help reduce errors while running workflows.

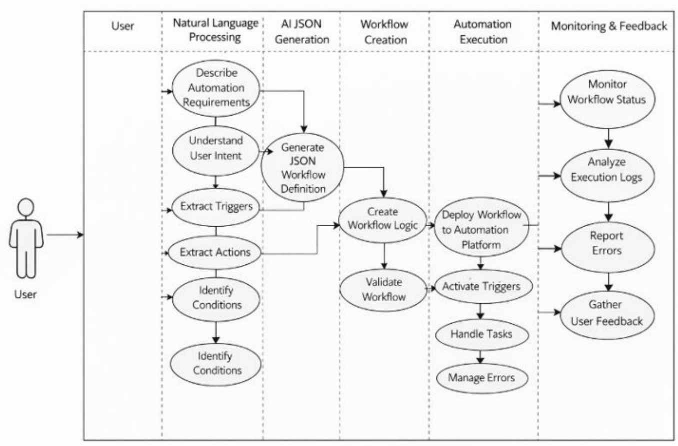

Fig. 1. UML Use Case Diagram for Automation System

Fig. 1. Flow Chart of Process Flow

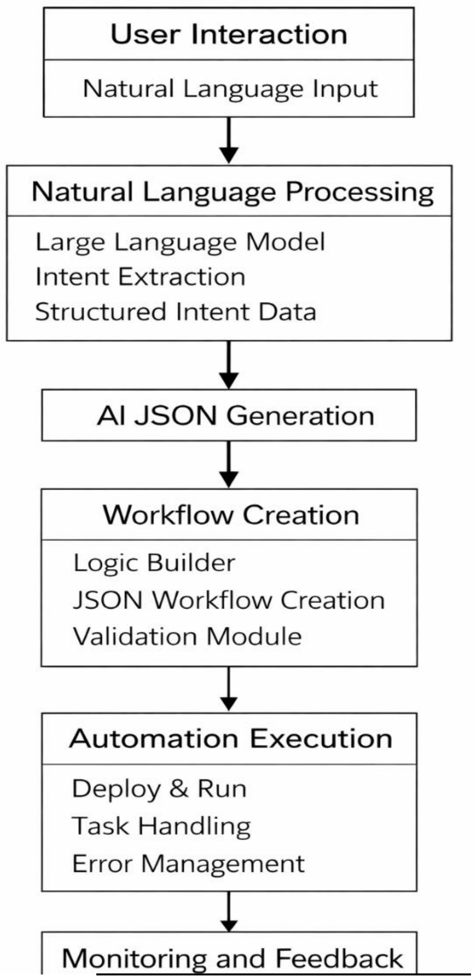

Fig. 2. System Architecture Diagram

V. METHODOLOGY

A. Research Design and Approach

The design-science research approach was used in this

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

work to build and test an intelligent system that can convert natural language instructions into executable automation workflows. This research mainly focusing on bridging the gap between human thoughts and machine-level workflow execution through the large language models (LLM) and automated workflow orchestrtion platforms.

The methodology is structured to ensure:

- Accurate understanding of user intent.
- Reliable workflow generation
- Smooth deployment and execution
- Measurable performance evaluation

B. *Personalized Treatment*

The system that input from users in simple English where they explain what kind of automation they need. These inputs can be short or long, simple or complex, and may be written in different ways.

Before the system starts processing, the input goes through minimal preprocessing so that the original meaning is not changed.

This includes:

- Removing extra spaces and unnecessary formatting.
- Converting time-related phrases into a standard format
- Standardizing names of application and file types so they are recognized correctly

This simple cleaning step helps the system understand the input clearly without changing what the user actually meant, and it makes the next steps more consistent and reliable.

C. *Intent Extraction Using Large Language Models*

The main methodological component involves leveraging a large language model to generae JSON structured information from natural input.
The LLM is prompted using a controlled prompt design to perform human intent identification, Trigger detection, Action extraction, Condition recognition, Condition recognition, Parameter interface.
The output is the structured intermediate representation containing all workflow elements in a machine-readable format. This representation serves as the foundation for subsequent workflow generation.

Prompt engineering technique used to guide the model so that:

- It gives consistent and predictable outputs.
- It follows a fixed structure of format (schema).
- It avoids generating incorrect, made-up or unclear information.

D. *Workflow Logic Construction*

The system converts the structured user request into a basic workflow plan. This plan explains how the automation should work step by step. It shows the order of tasks, handles conditions, manages how data moves between steps, and includes what to do if an error occurs.

After creating this plan, the system checks it carefully. It makes sure all necessary parts of the workflow are included, the steps are in the correct order , and there are no logical mistakes like circular loops. This ensure the workflow is clear, correct, and logically consistent before it is converted into a format that a specific automation platform can use.

E. *Workflow Translation to Executable Format*

Once validated, the abstract workflow logic is translated into an executable workflow definition compatible with the n8n automation platform. This involves:

- Mapping abstract actions to n8n nodes
- Defining node parameters and credentials
- Establishing data connections between nodes
- Assigning trigger configurations

The workflow is structured as JSON format, which serves as the deployment artifact.

F. *Automated workflow Deployment*

The generated workflow JSON is programmatically deployed to the n8n platform using authenticated REST APIs. This stage includes:

- Workflow registration
- Activation triggers
- Version control handling.
- Validation of deployment success

Automated deployment eliminates manual configuration and ensures rapid workflow availability

G. *Execution Monitoring and Error Handling*

After deployment, workflow execution is continuously monitored. The system captures:

- Execution logs
- Success and failures states
- Runtime errors and exceptions

In case of failures, the system identifies the error source and records diagnostic information. The feedback is used for the improvement of intent interpretation and workflow generation accuracy

VI. RESULTS AND EVALUATION

A. *Experimental Setup*

To evaluate the effectiveness of the proposed natural language-driven workflow generation system, a series of controlled using a set of automation scenarios. The experiments were designed to assess the system'sability to correctly interpret user intent, generate valid workflows, and execute them successfully on the automation platform. A total of multiple real-world automation tasks were defined, ranging from simple single-step workflows involving conditional logic and data transfer across applications. These tasks were expressed using natural

language instructions written in different styles and levels of detail to stimulate real user behavior.
The evaluation environment consisted of a backend service integrated with a large language model and an n8n workflow engine deployed in a local server environment.

B. Evaluation Metrics

The system was considering the evaluation using performance metrics:

- Workflow Generation Accuracy: Measures the percentage of user instructions that resulted in correctly structured and logically valid workflows.
- Execution Success Rate: Represents the proportion of generated workflows that were executed successfully without runtime errors on the automation platform.
- Intent Interpretation Accuracy: Evaluates how accurately the system extracted triggers, actions, and conditions from natural language input.
- User Effort Reduction: Assesses the reduction in manual configuration steps compared to traditional visual workflow creation
- Error Rate: Measures the frequency of logical or execution errors encountered during workflow execution.

These metrics collectively provides a comprehensive assessment of both system correctness and usability.

C. Test Case Categories

The test cases were grouped into three categories based on complexity:

- Simple Workflows: Single trigger and single action
- Moderate Workflows: Multiple actions with data transfer between steps.
- Complex Workflows: Conditional branching, scheduling, and multi-service integration

This categorization allowed the evaluation of system performance under increasing level of complexity.

D. Result Analysis

1. Workflow Generation Accuracy

The system performed high accuracy in converting natural language instructions into structured workflows. Simple and moderate workflows achieved near-perfect generation accuracy, while complex workflows showed slightly reduced accuracy due to ambiguous or implicit conditions in user input.

This results indicate that the intent-driven approach effectively captures user requirements even when instructions are expressed informally

2. Execution Success Rate

Most generated workflows executed successfully on the n8n platform. Execution failures were primarily attributed to:

- Missing or invalid third-party service credentials
- Unsupported user-reqested operations
- External service unavailability

3. Intent Interpretation Performance

The Large Language model demonstrate strong performance in identifying triggers, actions, and sequencing. The system performed particularly well when instructions were complete and explicit. Minor degradation was observed when instructions contained vague temporal expressions or implied conditions.

These findings highlight the importance of clear user input while also demonstrating the robustness of semantic understanding.

4. *User Effort Reduction*

Compared to traditional manual workflow creation using visual editors, the proposed system significantly reduced user effort. Users were able to define automation workflows using a single instruction rather than configuring multiple nodes and connections manually.

This reduction in effort directly contributes to improved accessibility for non-technical users.

5. Comparative Evaluation

A qualitative comparison was performed between the proposed system and traditional workflow automation approaches:

TABLE I. TRADITIONAL VS PROPOSED SYSTEM

Aspect	Traditional Tools	Proposed System
Workflow Creation	Manual Configuration	Natural Language Input
Technical Knowledge Required	High	Minimal
Error Proneness	High	Low
Time to Deploy	Long	Short
Scalability	Limited by user skill	High

This comparison demonstrates that the proposed approach offers substantial advantages in usability and efficiency

6. Summary of Results

The system evaluation explains about the proposed system:

- Accurately interprets natural language instructions
- Generates valid and executable workflows
- Reduces manual effort and configuration errors
- Scales effectively to complex automation scenarios

These results validate the effectiveness of the proposed methodology and support its suitability for real-world automation use cases.

VII. DISCUSSION

The result presented in previous section demonstrate the practical feasibility and effectiveness of the proposed natural language-driven workflow generation system. This discussion analyses these findings in depth, interprets their

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

implications, and situates the proposed approach within the broader context of existing workflow automation and intelligent systems research.

A. Interpretation of Key Findings

One of the most significant observations from the evaluation is the system's strong ability to translate natural language instructions into logically valid and executable workflows. This indicates that treating workflow automation as an intent-to-execution translation problem is a viable and effective approach.

The high workflow generation accuracy for simple and moderate automation tasks suggests that large language models are well-suited for capturing user intent, even when instructions are expressed informally or without strict structure. This confirms the central premise of this research: that sematic understanding can replace manual configuration in many automation scenarios.

For complex workflows, a slight reduction in accuracy was observed. This outcome is expected, as complex instruction often contain implicit assumptions, vague condition, or domain-specific references. The system was still able to generate functional workflows in most cases, highlighting the robustness of the logical reasoning layer.

B. Impact on usability and accessibility

A key contribution of the proposed system lies in its ability to significantly reduce the technical expertise required to create automation workflows. Traditional automation platforms demand familiarity with triggers, actions, data mapping, and execution logic, which limits their accessibility to non-technical users.

The evaluation results demonstrate that users can achieve the same automation outcomes by expressing requirements in a single natural language instruction. This reduction in user effort directly improves usability and lowers the barrier to adoption, supporting the goal of less process automation.

From a human - computer interaction prescriptive, the system aligns more closely with natural human communication patterns, making automation more intuitive and less error-prone.

C. Reliability and Execution Behavior

During testing, most of the generated workflows worked successfully. This shows that the system creates workflows that are properly structured and suitable for real-world automation platforms. Most of the failures were not caused by mistakes in the workflow logic. Instead, they happened due to external issues such as missing login credentials or limitations of third-party services.

This is important because it proves that the core system is reliable. Many of these failures can be reduced by improving system integration and providing better user guidance.

D. *Comparison with Existing Approaches*

Compared to traditional visual workflow tools and template-based systems, the proposed method has clear advantages, It is more flexible because it does not depend on fixed templates and can handle new types of automation tasks. It also reduces complexity by automatically generating workflows instead of requiring manual configuration. In addition, it can handle more complex workflows without increasing the effort required from the user.

However, traditional system may still be better in highly controlled environments where fully predictable behavior is necessary. Therefore, the proposed system should be seen as a complement to existing tools rather than a complete replacement.

E. Limitations and Challenges

The system has some limitations. The quality of the generated workflow depends on how clearly the user explains their request. If the instructions are unclear or incomplete, the output may not be perfect, especially for complex tasks.

Since the system relies on large language models, there are also challenges related to consistency. transparency, and cost. Although prompt design helps reduce these issues, further improvements are needed. Another challenge is handling integrations that are not directly supported by the automation platform, which may require custom development.

F. Implications for Future Automation System

This study shows that natural language-based automation can change how people use workflow systems. By removing technical complexity, it can make automation more accessible in areas like business processes, data management, and personal productivity.

The system's modular design also provides a strong base for future improvements, such as self-improving automation systems that learn from user feedback and execution results.

VIII. **CONCLUSION**

This paper presented a natural language-driven workflow generation system that integrates large language models with the n8n platform. The system allows users to create automation workflows by simply describing their requirements in plain English. The proposed approach effectively translates users intent into structured, executable workflows. Experimental evaluation showed high accuracy in workflow generation and strong execution success rates across different complexity levels. Most failures were due to external service limitations rather than issues in the core system logic.Compared to traditional visual and template-based tools, the system reduces manual configuration effort and lowers the technical barrier for non-technical users. It simplifies complex workflow creation while maintaining reliability and scalability.

Overall, the results confirm that natural language-based automation is a practical and efficient solution for modern workflow systems. The modular architecture also provides a solid foundation for future improvements, including

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

enhanced integrations and adaptive workflow refinement.

REFERENCES

[1] M. Sajid, A. Khalil, M. A. Iftikhar, M. H. Rehman, M. A. Shah, and S. Rho, "Disseminating the risk factors with enhancement in precision medicine using comparative machine learning models for healthcare data," *IEEE Access*, vol. 12, pp. 72 794–72 812, 2024.

[2] Y. Deng, C. Yang, H. Zhang, and Y. Zhang, "Personalized federated learning for in-hospital mortality prediction of multi-center icu," *IEEE Access*, vol. 11, pp. 11 652–11 663, 2023.

[3] T.-C. Do, H.-J. Yang, G.-S. Lee, S.-H. Kim, and B.-G. Kho, "Rapid response system based on graph attention network for predicting in- hospital clinical deterioration," *IEEE Access*, vol. 11, pp. 29 091–29 100, 2023.

[4] Manvi and A. Sridevi, "Automated medical diagnosis with deep learning," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2023.

[5] S. Gujjeti and P. Pal, "The transformative impact of deep learning on personalized medicine," *International Journal of Innovative Science and Research Technology*, vol. 9, no. 5, 2024.

[6] K. P, H. D and H. S, "Secure Data Sharing through Blockchain Integration with Robotic Process Automation," 2025 Global Conference in Emerging Technology (GINOTECH), PUNE, India, 2025, pp. 1-5, doi: 10.1109/GINOTECH63460.2025.11076625. https://ieeexplore.ieee.org/document/11076625.

[7] Kumar P and J. N (2024), "Synergistic Revolution with AI-Enabled Organ Matching and Blockchain Security in Transplantation Management," 2024 Second International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, Karnataka, India,2024,pp.1-6, doi: 10.1109/ICAIT61638.2024.10690290.

[8] S. Saravanan, M. Sivabalakrishnan, N. Duraimurugan, D. Divya, "Artificial Intelligence Security Model For Privacy Renitence In Big Data Analytics", Applied Mathematics & Information Sciences, 2022, Vol-16 [6], pp. 919-927, doi:10.18576/amis/160608.

[9] A. K, A. C and A. M, "LeNet – IoT Security Aware Intrusion Detection Framework Using Deep Learning," 2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2024, pp. 1-7, doi: 10.1109/ICSES63760.2024.10910309.

[10] N. Duraimurugan, R. A and R. S, "ResQ – Intelligent Real-Time Disaster Management Using Machine Learning and IoT," 2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2024, pp. 1-7, doi: 10.1109/ICSES63760.2024.10910703.

[11] M. Alazab, S. Venkatraman, P. Watters, M. Alazab, and R. Alazab, "Zero-Day Malware Detection based on Supervised Learning Algo- rithms of API Call Signatures," in *Proc. 9th Australasian Data Mining Conference*, 2011, pp. 171–182.

[12] A. K, A. C and A. M, "Security-Aware IoT Attack Detection Framework," 2025 International Conference on Computing and Communication Technologies (ICCCT), Chennai, India, 2025, pp. 1-6, doi: 10.1109/ICCCT63501.2025.11019030.