IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 10 ISSUE : 12 Print / Issue Publication Date: April 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v10i12.014

Indexed In

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 105-107
Published Online April 2026 in IJEAST (http://www.ijeast.com)

AI-POWERED WORKFLOW AUTOMATION USING MODEL CONTEXT PROTOCOL

Narada Ravi, Aviti Abhishek, Tirumalaraju Haripriya, Tammineni Kalyan Kumar, Dr. Sk. Althaf Rahaman
Dept. of Computer Science and Systems Engineering
Lendi Institute of Engineering and Technology Vizianagaram, Andhra Pradesh, India

Abstract—**This paper presents an AI-powered workflow automation system integrating Model Context Protocol (MCP) with Cursor IDE. The system enables natural language-based automation of tasks such as email handling, scheduling, and data retrieval. By leveraging Large Language Models (LLMs) and structured communication, the system reduces manual effort, minimizes context switching, and improves productivity. Experimental evaluation demonstrates real-time task execution, scalability, and efficient workflow automation. Index Terms—Workflow Automation, MCP, Artificial Intelligence, LLM, API Integration**

Keywords— **AI Workflow Automation, Model Context Protocol (MCP), Cursor IDE, Large Language Models (LLMs), Tool Integration, API-Based Automation, Intelligent Systems**

I. INTRODUCTION

Modern software development workflows require continuous interaction with multiple tools such as email systems, scheduling platforms, APIs, and development environments. This constant switching between tools reduces efficiency, increases cognitive load, and slows down overall productivity. Developers often spend significant time on repetitive tasks like debugging, documentation, communication, and data handling, which limits their ability to focus on innovation and problem-solving.

Traditional automation systems provide limited support by relying on predefined rules and scripts, making them rigid and difficult to adapt to dynamic requirements. On the other hand, AI systems, especially Large Language Models (LLMs), offer strong capabilities in understanding natural language and generating intelligent responses. However, these systems lack direct execution ability, as they cannot independently interact with external tools or perform real-world actions.

To bridge this gap, the Model Context Protocol (MCP) has been introduced as a standardized framework that enables secure and structured communication between AI models and external tools. MCP allows AI systems to access real-time data, invoke APIs, and execute tasks across multiple platforms while maintaining contextual consistency.

In this paper, we propose an AI-powered workflow automation system that integrates MCP with Cursor IDE to enable intelligent task execution. The system allows users to perform complex workflows using natural language prompts, automating tasks such as email handling, scheduling, and data retrieval. By combining the reasoning capability of LLMs with the execution capability provided by MCP, the proposed system reduces manual effort, minimizes context switching, and improves overall productivity.

II. LITERATURE SURVEY

Existing automation tools such as Zapier provide rule-based workflows that can connect multiple applications, but they lack intelligence and adaptability to dynamic user requirements. These systems operate on predefined triggers and conditions, making them limited in handling complex or context-aware tasks.

Recent AI-based frameworks such as Tool former and ReAct enhance decision-making by enabling Large Language Models (LLMs) to select and use tools during reasoning. While these approaches improve the capability of AI systems to interact with external resources, they lack a standardized and secure mechanism for structured execution across multiple platforms.

To overcome these limitations, the proposed system integrates the Model Context Protocol (MCP) with LLMs. MCP provides a structured and secure communication framework that enables seamless interaction between AI models and external tools. This approach supports scalable, context-aware, and real-time workflow automation across diverse applications.

III. SYSTEM ARCHITECTURE

The architecture of a system built using the Model Context Protocol (MCP), which enables smooth interaction between users, AI models, and external services. The system consists of three main components: MCP Clients, MCP Host, and MCP Servers. MCP Clients include interfaces such as IDEs, chat applications, and mobile apps through which users provide input in the form of prompts.

The MCP Host acts as the central processing unit powered by a Large Language Model (LLM). It interprets user requests, manages context, and decides which tools or

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 10, Issue 12, ISSN No. 2455-2143, Pages 105-107
Published Online April 2026 in IJEAST (http://www.ijeast.com)

services need to be used. It also handles communication between the client and the servers using secure API calls.
MCP Servers provide access to various tools and services, including data servers, tool servers, and action servers. These servers connect to external platforms such as Gmail, LinkedIn, and web APIs, enabling the system to perform real-world tasks. After execution, the results are sent back to the MCP Host, processed by the LLM, and returned to the user in a clear and readable format.

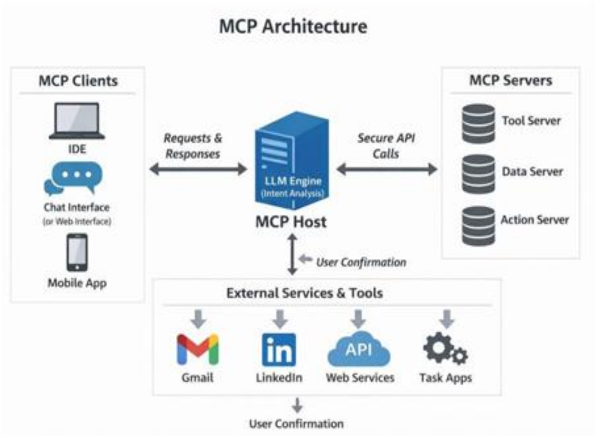

Fig. 1. MCP System Architecture

IV. METHODOLOGY

The system follows a structured workflow:

A. Input Processing

User provides natural language input in MCP Host..

B. Task Identification

LLM analyzes intent and selects required tools dynamically.

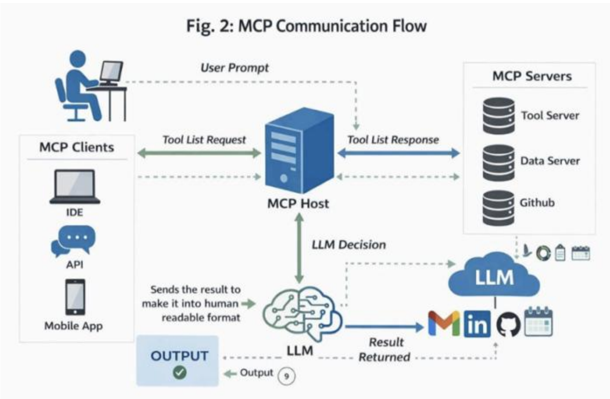

Fig. 2. MCP Communication Flow

C. Execution

MCP communicates with APIs to execute tasks securely.

D. Output Generation

Results are processed and displayed in human-readable format.

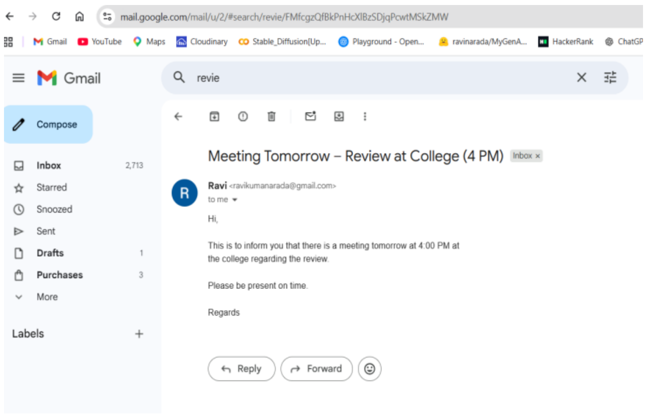

Fig. 3. Automated Email Generation using MCP

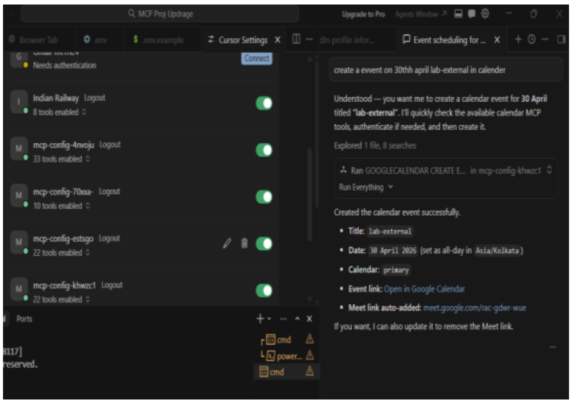

Fig:4 Event creation in calendar in MCP Host

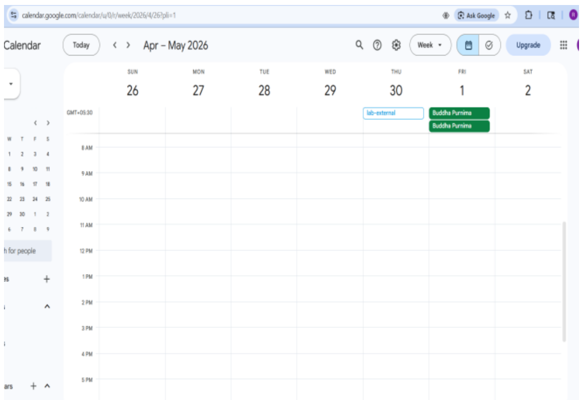

Fig:5 Lab-external event created in calendar

V. COMMUNICATION FLOW

1. User enters a prompt in Cursor IDE.
2. MCP client sends the request to the MCP Host.
3. MCP Host fetches available tools from the MCP Server.

4. User prompt + tool list are sent to the LLM.
5. LLM analyzes the request and selects the required tool.
6. MCP Host sends execution request to MCP Server.
7. MCP Server performs the action using external services (Gmail, LinkedIn, APIs).
8. Result is returned to the MCP Host.
9. LLM formats the response into readable output.
10. Final result is displayed to the user.

VI. IMPLEMENTATION

The proposed system is implemented using Streamlit, the Model Context Protocol (MCP) framework, and Large Language Model (LLM) APIs. Streamlit is used to build an interactive user interface for input handling and output visualization.

The MCP framework is integrated to enable structured communication between the LLM and external tools. External services such as Gmail, LinkedIn, and web APIs are connected through MCP servers to support real-time task execution.

The LLM processes user prompts, selects appropriate tools, and coordinates actions through MCP. This implementation ensures seamless automation, efficient workflow execution, and real-time interaction with multiple platforms.

VII. RESULTS AND DISCUSSION

A. System Output

The system successfully automates real-world tasks using natural language input. As shown in the figures, the system generates emails automatically and retrieves train ticket availability using integrated APIs.

B. Performance Analysis

The system demonstrates:
• Real-time task execution
• Reduced manual effort
• Improved workflow efficiency
• Accurate API-based responses

C. Discussion

The results confirm that MCP enables seamless integration between AI models and external tools. The system handles multiple tasks efficiently while maintaining accuracy and scalability. Performance depends on API latency and network conditions.

VIII. CONCLUSION

The results confirm that MCP enables seamless integration between AI models and external tools. The system handles multiple tasks efficiently while maintaining accuracy and scalability. Performance depends on API latency and network conditions

IX. REFERENCE

[1]. Lacity M, Willcocks L, Craig A, “Robotic Process Automation” Journal of Information Technology, Vol. 30, Nov 2, 2015, pp. 130-145.
[2]. Brown A, Davis R, Wilson P, “Artificial Intelligence Techniques for Workflow Automation” International Journal of Computer Applications (IJCA), Vol. 178, Nov 25, 2019, pp. 10-15.
[3]. Brown T. B., Mann B., Ryder N, “Language Models are FewShot Learners” Advances in Neural Information Processing Systems (NeurlPS), 2020, pp. 1877-1901.
[4]. Smith J, Johnson M, Lee K, “AI-Powered Decision Making in RPA Workflows: The Rise of Intelligent Decision Engines” IEEE Conference on Intelligent Systems, 2021, pp. 45-50
[5]. Nakano R, Hilton J, “WebGPT: Browser-assisted Question Answering with Human Feedback” arXiv preprint arXiv:2112.09332, 2021, pp. 120.
[6]. Yao S, Zhao J, Yu D, “ReAct: Synergizing Reasoning and Acting in Language Models” arXiv preprint arXiv:2210.03629, 2022, pp.1-18.
[7]. Karpas E, “MRKL Systems: Modular Reasoning, Knowledge and Language” arXiv preprint arXiv:2205.00445, 2022, pp. 1-12.
[8]. Schick T, Dwivedi Y, “Toolformer : Language Models Can Teach Themselves to Use Tools” arXiv preprint arXiv:2302.04761, 2023, pp. 1-20.
[9]. Significant Gravitas, “AutoGPT: Autonomous AI Agents” Github Repository / Technical Documentation, 2023, pp. 1-10.
[10]. Harrison Chase, “LangChain: Building Applications with LLMs” Technical Documentation / White Paper, 2023, pp. 1-15.
[11]. J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of NAACL-HLT, 2019.
[12]. A.Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems (NeurIPS), 2017.

2455-2143