E-ISSN: 2583-9667
Indexed Journal Peer Reviewed Journal
Received: 08-01-2025
https://multiresearchjournal.theviews.in
Accepted: 19-03-2025

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH IN MULTIDISCIPLINARY

Volume 3; Issue 2; 2025; Page No. 390-393

AI Browser Automation Using Gemini API and Web UI

[1]Venkata Reddy V and [2]Dr. Sheela

[1]Student, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu, India
[2]Assistant Professor, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu, India

DOI: https://doi.org/10.5281/zenodo.15623544

Corresponding Author: Venkata Reddy V

Abstract

This project, titled "AI Browser Automation Using Gemini API and Web UI," integrates artificial intelligence with browser automation to perform web tasks efficiently. Using the Browser-Use framework, Playwright, and a Gradio Web UI, the AI agent interprets user instructions through the Gemini API, navigating websites, extracting data, and executing actions like searches and transactions. It supports persistent sessions, custom configurations, and high-definition screen recording for monitoring. The system demonstrated over 90% task accuracy, showcasing its potential in data scraping, autonomous browsing, and AI-powered web automation, forming a foundation for future AI-driven automation tools.

Keywords: AI Browser Automation, Gemini API, Playwright Framework, Gradio Web UI, Large Language Model (LLM)

1. Introduction

This project introduces an AI-powered browser automation system that performs web tasks intelligently using natural language instructions. The AI agent, powered by the Gemini API, interprets user commands and controls browser actions through the Playwright framework. It can search, extract data, and simulate transactions on websites, offering real-time task execution with high accuracy. A Gradio-based Web UI ensures user-friendly interaction, while features like persistent sessions and high-definition screen recording enable monitoring and debugging. Designed to be efficient, scalable, and easily extendable, the system demonstrates practical AI applications in data scraping, autonomous browsing, and automated online operations.

2. Purpose of the project

The primary goal of this project is to develop an AI-powered browser automation system that can perform web-based tasks using natural language instructions, improving efficiency in data handling and online operations. It integrates the Gemini API with the Browser-Use framework and Playwright for executing tasks like searches, data extraction, and simulated transactions. The system aims to minimize manual intervention by enabling real-time, intelligent web interactions through a Gradio-based Web UI. Designed to be scalable, user-friendly, and educational, this project offers students hands-on experience in AI, web automation, and large language model integration while promoting practical AI-driven solutions for everyday web-based tasks.

3. Existing System

Traditional browser automation systems are typically built using tools like Selenium or basic Playwright scripts, primarily relying on static, predefined code to perform web actions. These systems can execute simple tasks like form filling or clicking buttons but lack the ability to intelligently understand or respond to natural language instructions. Most require extensive manual configuration for each specific website and are prone to failure with dynamic web content, CAPTCHAs, or multi-step workflows. They also lack user-friendly interfaces, real-time monitoring, screen recording, or flexible session management. Additionally, existing solutions are developer-centric, non-scalable, and unsuitable for non-technical users, highlighting the need for a smarter, AI-powered, automated browsing solution.

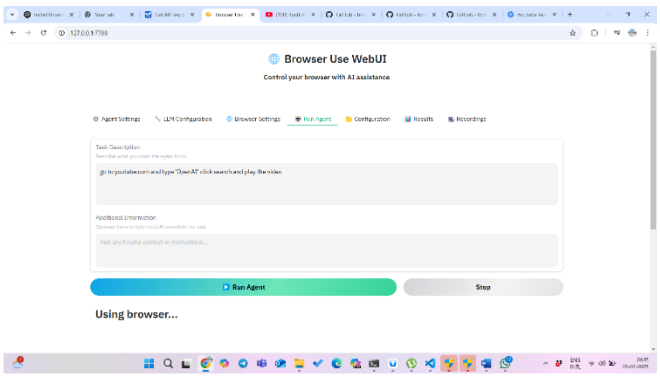
4. Proposed System

The proposed system introduces an AI-powered, smart, and automated browser automation solution designed for enhanced efficiency, scalability, and real-time response. Centered around the Gemini API and the Browser-Use framework, it interprets natural language commands and performs web-based tasks without human intervention. When a task is issued, the system automatically executes web actions via the Playwright framework, retrieving data, navigating pages, or simulating transactions. A Gradio Web UI provides live status updates, while the entire system supports persistent sessions and screen recording for monitoring. The platform is modular, lightweight, and easy to deploy across different systems. It is cost-effective, user-friendly, and designed for continuous autonomous operation. Additionally, its components are easy to maintain and upgrade, with future support for multiple AI models and APIs, making it adaptable for evolving web automation needs.

5. Software Requirements

The AI Browser Automation system is built using the Gemini API, a large language model capable of processing natural language inputs and generating context-aware responses. Paired with Python, it enables seamless integration with browser automation frameworks like Playwright and Browser-Use. Python offers a beginner-friendly syntax, extensive libraries, and strong support for AI, web scraping, and automation tasks. The Gradio library provides an intuitive Web UI for interacting with the AI agent. Key Python modules include playwright for browser control, requests for handling web requests, and json for data processing. The system runs efficiently on any Windows, Linux, or macOS environment with Python 3.x installed. While web-based by default, it can be extended with APIs or databases for real-time data storage and remote task management.
Python offers modular coding with custom functions and extensive library support, making it flexible, beginner-friendly, and suitable for everything from learning projects to AI-powered automation systems. It encourages rapid prototyping and experimentation—perfect for both educational and real-world applications **like browser** automation. Python's compatibility with AI APIs and automation frameworks makes it a practical, scalable tool for dynamic web interactions.

6. Hardware Requirement


127.0.0.1:7788

Browser Use WebUI
Control your browser with AI assistance

Task Description
Describe what you want the agent to do
go to youtube.com and type 'OpenAI' click search and play the video

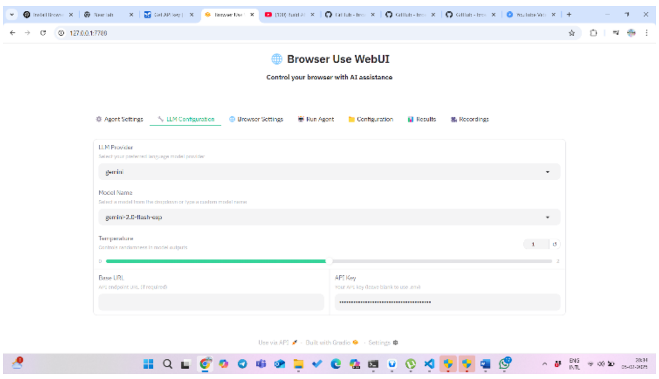
Additional Information
Optional hints to help the LLM complete the task
Add any helpful context or instructions...

Using browser...

Fig 1: Arduino Uno

The core AI agent processes user instructions and controls the browser automation tasks, including web navigation, data extraction, transactions, and real-time monitoring, ensuring intelligent, efficient, and autonomous online operations.

.


127.0.0.1:7788

Browser Use WebUI
Control your browser with AI assistance

LLM Provider
Select your preferred language model provider
gemini

Model Name
Select a model from the dropdown or type a custom model name
gemini-2.0-flash-exp

Temperature
Controls randomness in model outputs
1

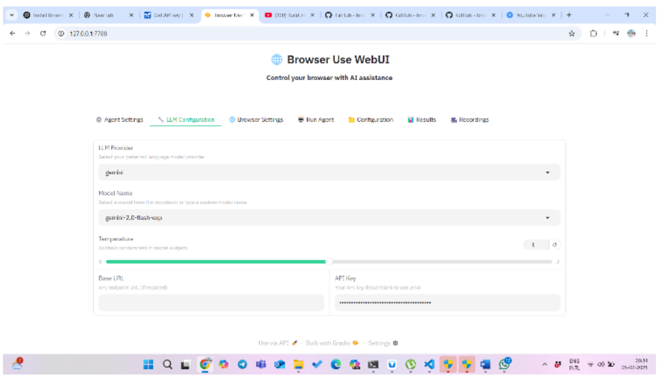
Base URL
API endpoint URL (if required)

API Key
Your API key (leave blank to use .env)
••••••••••••••••••••••••••••••••••••••

Fig 2: AI

Allows the AI agent to control browser actions like clicking buttons, submitting forms, or navigating pages based on user instructions and real-time website responses.

.


127.0.0.1:7788

Browser Use WebUI
Control your browser with AI assistance

LLM Provider
Select your preferred language model provider
gemini

Model Name
Select a model from the dropdown or type a custom model name
gemini-2.0-flash-exp

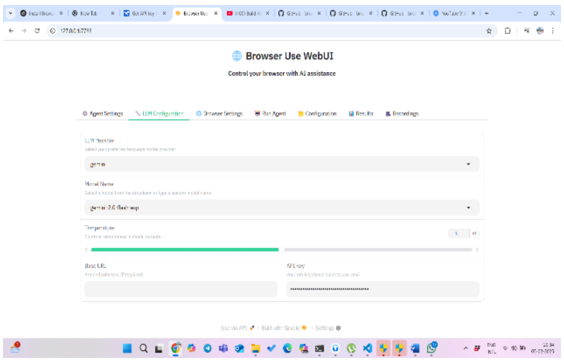
Temperature
Controls randomness in model outputs
1

Base URL
API endpoint URL (if required)

API Key
Your API key (leave blank to use .env)
••••••••••••••••••••••••••••••••••••••

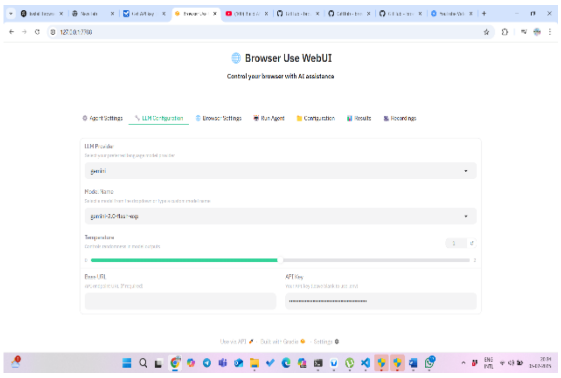
Fig 3: AI Web

Displays real-time task status, browser activity, and AI responses through a user-friendly Gradio Web UI. The interface simplifies interaction by allowing plain language instructions and clear visual feedback.


Browser Use WebUI
Control your browser with AI assistance
LLM Provider: gemini
Model Name: gemini-2.0-flash-exp
Temperature: 1

Automatically performs web actions like clicking buttons, submitting forms, or closing browser tabs when instructed, helping automate repetitive online tasks efficiently and reliably.

.


Browser Use WebUI
Control your browser with AI assistance
LLM Provider: gemini
Model Name: gemini-2.0-flash-exp
Temperature: 1

Alerts users with on-screen notifications or status messages when tasks fail, complete, or encounter errors, enabling quick review, correction, or further action.

.

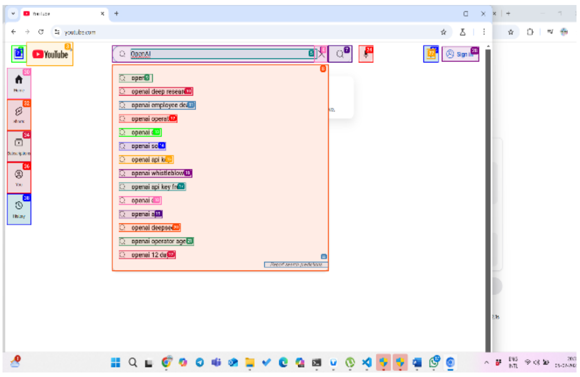
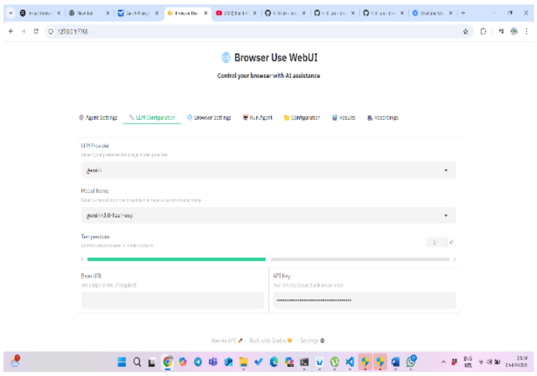


Fig 6: Exhaust Fan

Alerts users with on-screen notifications or status messages when tasks fail, complete, or encounter errors, enabling quick review, correction, or further action.

.


Browser Use WebUI
Control your browser with AI assistance
LLM Provider: gemini
Model Name: gemini-2.0-flash-exp
Temperature: 1

Displays real-time task status, browser activity, and AI-generated responses through a Gradio Web UI. The simplified interface reduces user effort by enabling natural language input and clear visual feedback.

.

7. Implementation

System implementation involves integrating AI models, browser automation frameworks, and a user interface to enable intelligent web-based task execution. The system consists of key components such as the Gemini API, Browser-Use framework, Playwright automation library, and Gradio-based Web UI. The AI agent serves as the central unit, interpreting user instructions and controlling browser actions based on predefined logic. Using Python, the system is programmed to continuously process natural language inputs and translate them into browser commands. When a task is received, the AI agent automatically navigates the browser, retrieves data, submits forms, or simulates transactions. The Gradio UI displays task status, browser activity, and AI responses in real-time. The entire system operates within an event-driven architecture, initializing browser sessions, executing tasks, and monitoring outputs in a continuous loop. After software setup and integration, the system is thoroughly tested to ensure reliable and accurate task execution. Issues such as API errors, browser delays, or instruction misinterpretation are identified and resolved to guarantee dependable, real-time AI-driven automation for diverse web applications.

.

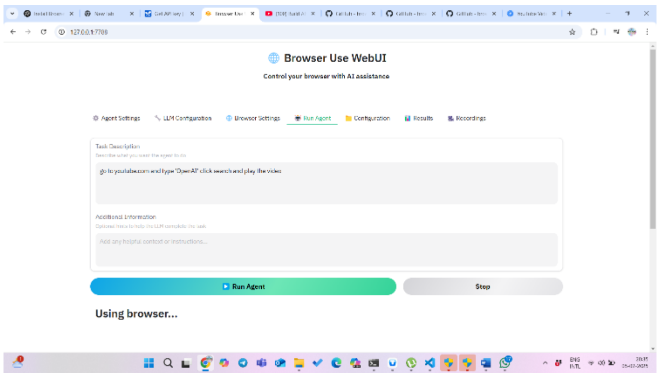

Browser Use WebUI
Control your browser with AI assistance
Task Description: go to youtube.com and type 'OpenAI' click search and play the video
Additional Information: Add any helpful context or instructions...
Using browser...

Fig 8: AI SAMPLE

The system architecture diagram for the browser automation framework illustrates how each module, AI agent, and web automation component is connected and coordinated. to The system architecture for the AI browser automation framework illustrates how each component is connected and coordinated to perform its function. The AI agent, powered by the Gemini API, serves as the central controller, receiving user instructions via the Gradio Web UI and determining the required browser actions. When a task is received, the AI processes the instruction and sends commands to the Browser-Use framework and Playwright, which navigate to the specified web pages and perform actions like clicking buttons, retrieving text, or filling forms. The Gradio interface provides real-time feedback, displaying task status and browser responses to the user, while on-screen alerts notify users of any issues or completed actions. The system architecture ensures persistent browser sessions and high-definition screen recording for activity monitoring. All components are carefully integrated using Python, ensuring efficient communication between the AI agent, automation frameworks, and user interface. A suitable runtime environment with Python 3.x, required libraries, and API keys powers the entire system, ensuring smooth, reliable, and intelligent browser automation.

.

8. Conclusion

The AI-based browser automation system using the Gemini API and Web UI offers an efficient and intelligent solution for automating web interactions in both personal and enterprise environments. It uses the Gemini AI agent to interpret user instructions, navigate websites, retrieve data, and perform actions such as filling forms, extracting URLs, and completing transactions. This automated workflow significantly reduces manual effort, improves operational speed, and minimizes errors during repetitive online tasks.

.

The AI-based browser automation system using the Gemini API and Web UI offers an intelligent, efficient, and reliable solution for automating web-based tasks in both personal and professional environments. By integrating the Gemini AI agent with frameworks like Playwright and Browser-Use, the system can navigate websites, perform data extraction, and execute predefined actions like form submissions or online transactions autonomously. This automated workflow improves efficiency, reduces manual effort, and enhances online productivity.

The AI browser automation system applies AI principles and offers practical learning opportunities in AI agent integration and web automation frameworks. Its adaptability allows it to interact with multiple websites and perform varied tasks, making it suitable for personal productivity, data scraping, and enterprise automation. Its autonomous operation ensures task completion without constant user oversight, reducing manual errors and increasing efficiency. Overall, the system is a scalable, cost-effective, and impactful solution for streamlining online workflows and demonstrating the future potential of AI-powered browser automation.

.

9. References

1. International Journal of Engineering Research & Technology (IJERT). Articles on AI-based gas Artifical systems and Automation integration [Internet]. IJERT; [cited 2025 Jun 9]. Available from: https://www.ijert.org/
2. International Research Journal of Engineering and Technology (IRJET). Research papers related to AI Agent and Automation System mechanisms [Internet]. IRJET; [cited 2025 Jun 9]. Available from: https://www.irjet.net/
3. IEEE Xplore Digital Library. Technical papers on realtime monitoring systems and Artifical applications [Internet]. IEEE; [cited 2025 Jun 9]. Available from: https://ieeexplore.ieee.org/
4. Hanwei Electronics Co. Ltd. MQ-5 Gas Sensor Datasheet [Internet]. Hanwei; [cited 2025 Jun 9]. Available from: https://www.winsen-sensor.com/
5. Arduino. Arduino official website: Programming Uno, pin configuration, hardware libraries [Internet]. Arduino.cc; [cited 2025 Jun 9]. Available from: https://www.arduino.cc/
6. CircuitDigest. Tutorials and circuit examples for interfacing sensors, relays, and servo motors with Arduino [Internet]. CircuitDigest; [cited 2025 Jun 9]. Available from: https://circuitdigest.com/
7. Electronics Hub. Guides on building AI systems with LCDs, buzzers, and automation logic [Internet]. ElectronicsHub; [cited 2025 Jun 9]. Available from: https://www.electronicshub.org/