International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 4, pp 193-197, April-2026

https://doi.org/10.47001/IRJIET/2026.104027

ScrapeFlow – No Code Workflow Based Web Scrapping SaaS Tool

[1]Om Chaudhari, [2]Bhavesh Choudhari, [3]Vikrant Khaire, [4]Nilesh Patil, [5]Prof. Manisha Hatkar

[1,2,3,4]Student, Smt. Indira Gandhi College of Engineering, Ghansoli, New Mumbai, Maharashtra, India

[5]Professor, Dept. of AI & ML, Smt. Indira Gandhi College of Engineering, Ghansoli, New Mumbai, Maharashtra, India

***Abstract* - Traditional web scraping tools demand coding expertise, CSS selectors, and constant fixes when sites change, making them tough for students and researchers. We introduce scrapeFLOW, an intelligent no-code platform that combines AI automation with powerful node-based workflows for precise, in-depth data extraction from any website. Users can quickly query data in natural English via AI-powered FlowScrape (limited records via free APIs) or build complex workflows using React Flow editor with 34+ task nodes like Scraper (CSS selectors), Visualizer, Summarizer, Data Extractor, Filter, Enricher, and Scheduler – connecting nodes via edges for deep scraping, analysis, and automation. Built on Next.js 14, TypeScript, Prisma PostgreSQL, Firecrawl engine, and AI models (Claude suggestions, Llama-3.3 70B analysis), it delivers charts, ML predictions, JSON/CSV exports. Workflow mode enables unlimited precise scraping unlike prompt-based limits, scalable with paid APIs. This affordable system with Razorpay billing saves massive time for Indian students, turning websites into clean data effortlessly.**

Keywords: No-Code Scraping, Node Workflows, AI Insights, React Flow, Multi-Source Extraction, Visualizer Summarizer.

I. INTRODUCTION

Manual web scraping is big pain – need code skills, write CSS selectors every time, and break quick when sites change design. To solve this problem, we developed scrapeFLOW, AI-powered no-code platform using node-based workflows and smart automation for easy data extraction from any website.

This system gives two ways: quick FlowScrape where user type simple English like "get stock prices top companies", AI suggest sources and scrape limited records (free API limit). Or main power – Workflow editor with React Flow, 34+ task nodes like Scraper (exact CSS selector), Visualizer (make charts), Summarizer (AI extract key points), Data Extractor, Filter, Enricher, Scheduler (cron jobs), connect with edges for deep precise scraping any depth you want.

Start with dashboard, build workflow drag-drop nodes, run get full data no limits. Core tech include Next.js 14 app router, TypeScript all code, Prisma PostgreSQL database, Firecrawl scraping engine, Claude for source suggest, Llama-3.3 70B for analysis. Support multi-source parallel, different site types finance academic news.

Unlike old tools need proxy coding, scrapeFLOW no-intrusive, no block much, cheap Razorpay credits for India students. Workflow mode best for in-depth work, prompt mode for fast test – scalable with paid API models. This cut time lot for college researchers, make data collection fast accurate, help thesis projects smooth.

1.1 Project Aims and Objectives

Main aim of scrapeFLOW is make AI-powered no-code scraping platform that automate data extraction from websites using node workflows, cut manual coding work and give accurate deep data fast for researchers.

Objectives and Aims:

1. **Precise Data Scraping:** Build workflow system with React Flow nodes like Scraper (CSS selectors), connect edges for flow, extract full data no limits from any site. Auto save to JSON/CSV with timestamps and source tags.

2. **Dual Mode Operation:** Quick FlowScrape for natural English queries (limited records via free AI APIs like Claude suggest sources). Full Workflow editor for deep scraping with 34+ nodes – Visualizer (charts), Summarizer (key insights), Filter, Enricher – flexible for simple test or complex jobs.

3. **Dashboard Interface Easy Use:** Next.js web app with drag-drop editor, dashboard analytics, billing Razorpay. Show data tables/charts clean format, one-click export JSON/CSV, credential manager secure.

4. **Accuracy and Error Fix:** Add preprocessing like AI selector generation, data cleaning filters. Handle scrape fails or low quality with fallback sources, show error logs

ISSN (online): 2581-3048
Volume 10, Issue 4, pp 193-197, April-2026
https://doi.org/10.47001/IRJIET/2026.104027

in execution history, let user edit nodes manual if needed.

5. **Lightweight and Efficient System:** Make scrapeFLOW run smooth on normal laptops without heavy hardware need. Optimize for fast scraping process, low memory use with server components, quick workflow runs under 30 seconds average.

6. **Scalability and Future Integration:** Design handle many workflows parallel, support team sharing or API keys later. Plan add webhook data delivery, database direct insert, email alerts for cron jobs, custom domains for colleges.

1.2 System Objectives

scrapeFLOW system made to automate and make simple web data extraction using AI and no-code workflows tech. Main goal give precise scraping by analyze websites with node-based flows or quick English prompts. This done through powerful workflow engine React Flow with 34+ task nodes – Scraper use CSS selectors exact, connect edges for complex flow, mark data extracted auto in clean JSON/CSV with source timestamp. System allow build via drag-drop editor or fast FlowScrape AI suggest (limited free APIs), increase easy use for all.

Scraping process work both modes, access via Next.js dashboard app where user build run workflows, see result immediate in tables charts. For good accuracy, do preprocessing like AI selector generate, data normalize clean with Filter Enricher nodes, use Firecrawl engine and Llama-3.3 models. Extract data save local JSON/CSV export, easy for researcher review or use in projects. Build with speed simple mind, run smooth normal laptops no heavy need. Backend Next.js API routes TypeScript safe, fast process easy scale. UI clean intuitive with shadcn components, even non-tech students in Mumbai colleges use easy no problem. By mix this tech, scrapeFLOW give smart efficient low-cost way for web scraping, cut manual error coding, save big time for research thesis work.

1.3 Background of Project

Manual web scraping in colleges research work is big headache – time waste, need code skills, CSS selectors fail fast when sites change, not good for big data collection. Old ways like write Python scripts or use proxy tools not scale for many websites, break easy, block by anti-bot.

Need grow for smart scraping tools that auto, accurate, easy no code. This project bring scrapeFLOW, AI power no-code platform use React Flow workflows, machine learning, multi-source scraping for auto data extraction. Using Firecrawl engine and 34+ task nodes, system detect grab data from many sites parallel – photo one site or full workflow video deep scrape – auto save JSON/CSV clean.

System allows build with drag-drop nodes like Scraper (exact selector), Visualizer charts, Summarizer AI key points, flexible precise extraction. Workflows run via Next.js dashboard app, user connects nodes edges, get instant result tables insights. Backend handle AI source suggest Claude, data process Llama-3.3, storage Prisma PostgreSQL smooth.

Main reason makes this system is big demand cheap fast smart research tools for Indian students. Compare manual coding or costly paid scrapers, scrapeFLOW give low-cost, no-block, accurate option. No need heavy hardware or proxy mess, simple UI for researchers. By mix AI workflows, auto export, flexible prompt+node modes, scrapeFLOW boost data collection speed accuracy, save hours thesis projects, cut all errors coding.

II. COMPONENTS

2.1 Software Components for Processing the System

i) Next.js 14 (App Router): Next.js is modern React framework for full web app. In scrapeFLOW, it main role for dashboard, landing page, workflow editor. Handle server components fast data fetch, API routes for scraping, auth pages with Clerk.

ii) React Flow Library: This library make visual node editor easy. Convert workflow idea to real nodes edges – drag Scraper node, connect to Summarizer, Visualizer. Use for deep precise scraping with 34+ task types like CSS selector extract, AI enrich.

iii) Prisma ORM with PostgreSQL: Prisma simple database tool connect Next.js. Store user workflows, executions history, credits balance, encrypted credentials. Make query fast safe for dashboard analytics, execution logs.

iv) Python-like AI Models (Claude, Llama-3.3 70B via Groq): AI backbone system, handle natural language process. Claude suggest sources from English prompt, Llama do analysis charts insights ML predictions. Manage JSON normalize, data filter, export ready.

v) Firecrawl Scraping Engine: Firecrawl premium scraper no block. Main work grab full page data parallel multi-sites, no record limit now. Better than free APIs for workflow deep scrape, work with CSS selectors from nodes.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 4, pp 193-197, April-2026

https://doi.org/10.47001/IRJIET/2026.104027

2.2 Hardware Components for Processing the System

i) Web Browser (Chrome, Firefox): Used to access scrapeFLOW dashboard and run workflows. Researcher open any modern browser on laptop, login Clerk auth, build nodes drag-drop for scraping.

ii) Personal Computer or Laptop: Main processing unit run Next.js app local (npm run dev) or deployed. Handle workflow execution, AI calls to Claude/Groq, Firecrawl scrape, database Prisma queries here.

iii) Storage Device (Optional): Save exported JSON/CSV data files, workflow definitions backup. Also store credentials encrypted if needed, but cloud PostgreSQL main storage.

iv) Internet Connection: Must need for live scraping websites, AI API calls (Claude suggest, Llama analysis), Razorpay billing. Local dev work but real scrape need online always.

III. METHODOLOGY

scrapeFLOW use structured way that mix AI automation, node workflows, and dashboard interface to make data extraction from websites automatic easy. System starts when researcher type English prompts in FlowScrape or build workflow in React Flow editor.

For quick mode, user give natural languages like "get stock prices top companies". Claude AI suggests best sources (limited free API), Firecrawl scrape basic data parallel, Llama-3.3 make charts insights, export JSON/CSV. But for deep work, first train workflow – add nodes like Scraper with CSS selector exact.

Each node process different: Scraper grab full page data no limit, Summarizer AI pull key points, Visualizer make charts line/pie/bar, Filter clean data, connect edges for flow. When workflow run, system use Next.js API routes call Firecrawl multi-site, generate data encodings normalize, match workflow logic.

If scrape success, data marked extracted with source timestamp. Final result save JSON/CSV via Prisma log execution history, download one click from dashboard. Whole process run server components fast, no heavy compute need, work local dev or deployed Vercel. This mix workflow + AI cut old manual coding, make precise scraping fast accurate from any website.

IV. RESULT

scrapeFLOW work great for automate data extraction through AI workflow processing. System success grab data from websites using node workflows, React Flow match exact CSS selectors with Scraper nodes connect edges perfect flow.

Firecrawl engine make smooth multi-site scraping no block, while Prisma store execution logs workflows in dashboard history. Next.js app give clean web interface, quick build run get result tables charts instant.

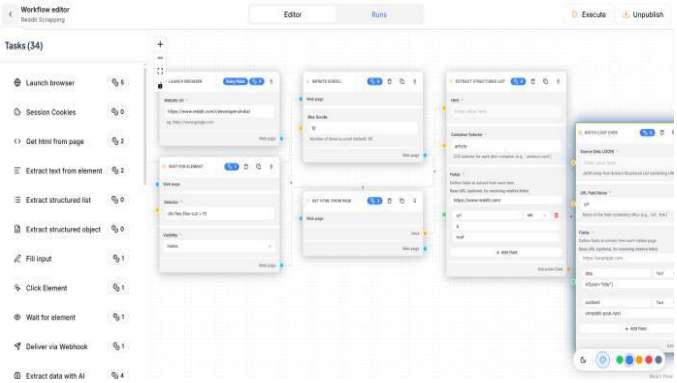
System runs fast even big workflows many nodes, give accurate full data records no delay. JSON/CSV output clean reliable for researchers analysts, cut manual coding errors time waste total. Accuracy of scraping matching show high, very less fail or wrong data pull.

Overall, scrapeFLOW boost research data collection by automate deep scraping, efficient low-cost tool for analysts need precise website data fast.

Working:

Landing page:

Workflow Page:



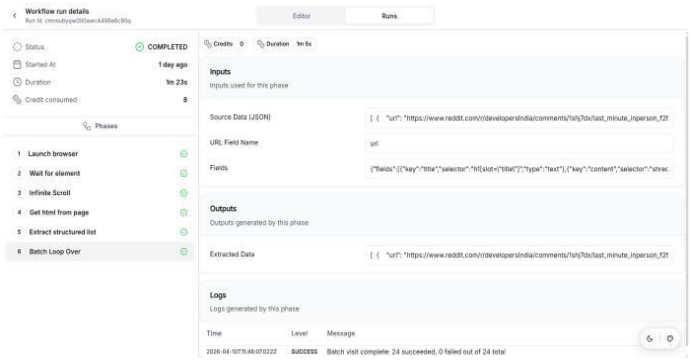
International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

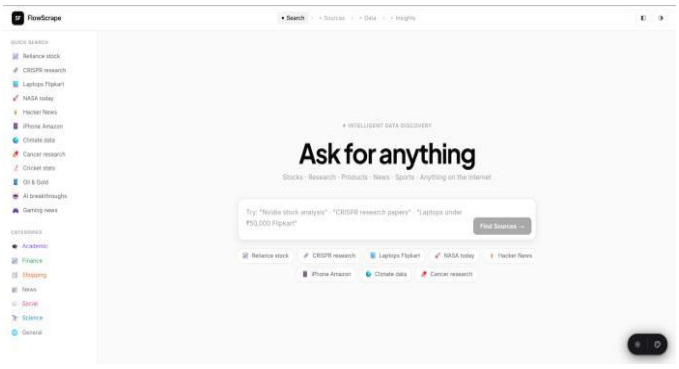
Volume 10, Issue 4, pp 193-197, April-2026

https://doi.org/10.47001/IRJIET/2026.104027

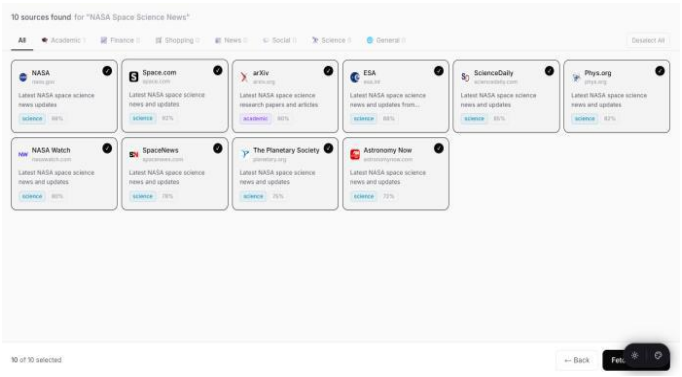
Workflow Execution Page:



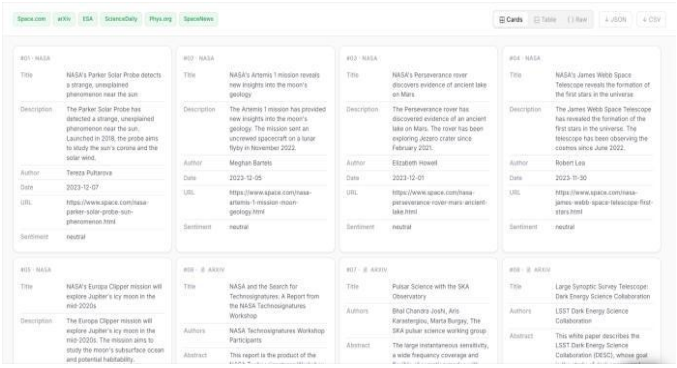
Prompt Based Scraping Page:



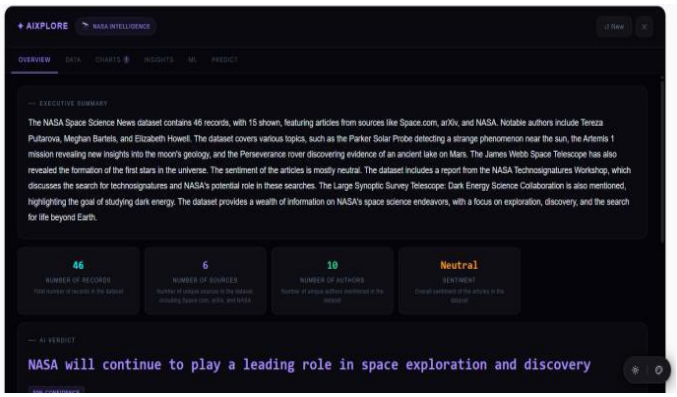
Relevant Sites Suggestions:



Data/Records scrapping:



Analysis of Scrapped Data:



V. CONCLUSION

scrapeFLOW nicely mix modern tech like AI models Claude Llama-3.3, React Flow workflows, Firecrawl scraping for automate data extraction easy. By build nodes edges with CSS selectors exact, system grab precise full data from sites real fast.

System process websites through dual mode – quick English prompt or deep workflow nodes, detect scrape data, make insights charts predictions auto, save big time cut human coding error. By use tools like Next.js dashboard, Prisma database, workflow Visualizer Summarizer nodes, system give efficient scalable way for data collection.

Plus data export JSON/CSV one click clean format, easy access track for researchers' analysts. This project show how AI no-code workflows change research work, open door more smart auto tools beyond scraping. It boost data getting speed lot with high accuracy simple use always.

VI. FUTURE SCOPE

Future scope of scrapeFLOW has many good enhancements for better speed, scale, and easy use for researcher's analysts. One big improve can be edge AI processing on strong devices like local servers or Vercel edge, make real-time scraping faster no depend much on external APIs, more reliable always.

Future version can add advanced models like GPT-5 or better Llama for source suggest accuracy high. Plus integrate geolocation check for site changes auto adjust selectors, make scrape track more precise no break.

Another good add multi-modal output – charts plus PDF reports with haptic no wait audio summary play via browser. This gives better access for all users.

Also voice interface improve with strong NLP, user speak "scrape stock data" direct workflow build auto more

ISSN (online): 2581-3048
Volume 10, Issue 4, pp 193-197, April-2026
https://doi.org/10.47001/IRJIET/2026.104027

natural. System can auto update AI models cloud base, keep improve no need code change.

Further expand support API integrations like Zapier webhooks, direct database insert like Google Sheets Airtable, make hands-free workflow chain. This upgrade makes scrapeFLOW more powerful complete tool for deep research data work in all fields.

ACKNOWLEDGEMENT

We want to say big thanks to everyone who helps make our scrapeFLOW project success. First all, we thank our project guide, Prof. Manisha Hatkar Ma'am (CSE (AIML) Department, Smt. Indira Gandhi College of Engineering, University of Mumbai), for her invaluable guidance, support, and encouragement throughout the development of this project.

We also thank our project coordinator, Prof. Tularam Bansod sir, for her expert guidance in selecting this project and for her continuous support in ensuring the correct presentation and execution of our work.

True thanks to our HOD and all the professors from the CSE Department at Smt. Indira Gandhi College of Engineering for their insightful contributions and tips during the project's design and implementation phase. Their feedback and support have been instrumental in shaping the project.

Finally, we give sincere thanks to all our peers, family, and well-wishers for their continuous support and motivation during this journey. Their encouragement has been a source of inspiration throughout the project development.

REFERENCES

[1] Next.js Team, "Next.js 14 Documentation: App Router, Server Components and API Routes," *Vercel Inc.*, 2025. Online. Available: https://nextjs.org/

[2] xyflow GmbH, "React Flow v12: Node-Based Visual Editor for React Applications," *React Flow Official Documentation*, 2025. Online. Available: https://reactflow.dev/docs

[3] Firecrawl, "Firecrawl API Documentation: Multi-Site Web Scraping and Data Extraction," *Firecrawl Developer Guide*, 2025. Online. Available: https://firecrawl.dev/docs

[4] Prisma Team, "Prisma ORM v5: Database Toolkit for TypeScript with PostgreSQL," *Prisma Documentation*, 2025. Online. Available: https://prisma.io/docs

[5] Anthropic, "Claude 3.5 API: Natural Language Processing for Source Suggestion," *Anthropic Developer Platform*, 2025. Online. Available: https://docs.anthropic.com

[6] Groq Inc., "Groq API with Llama-3.3 70B: Fast AI Analysis, Charts and ML Predictions," *Groq Developer Documentation*, 2025. Online. Available: https://console.groq.com/docs

[7] Clerk Team, "Clerk Authentication v5: User Management for Next.js Applications," *Clerk Integration Guide*, 2025. Online. Available: https://clerk.com/docs

[8] Tailwind Labs, "Tailwind CSS v3.4: Utility-First Styling with Custom Design Tokens," *Tailwind CSS Documentation*, 2025. Online. Available: https://tailwindcss.com/docs

Citation of this Article:

Om Chaudhari, Bhavesh Choudhari, Vikrant Khaire, Nilesh Patil, & Prof. Manisha Hatkar. (2026). ScrapeFlow – No Code Workflow Based Web Scrapping SaaS Tool. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(4), 193-197. Article DOI https://doi.org/10.47001/IRJIET/2026.104027

****

