

# AI-Powered User Management via UI Automation (No API) For CloudEagle

Submitted by: Shreya Upadhyay

Date: 12 April 2026

Demo: [https://drive.google.com/file/d/1JMYycjJWiBL\\_k-OltYlfzS9nRnsxxJ-d/view?usp=sharing](https://drive.google.com/file/d/1JMYycjJWiBL_k-OltYlfzS9nRnsxxJ-d/view?usp=sharing)

Code: <https://github.com/shreyaup12/saas-ui-automation>

## 1. Understanding the Problem

Modern SaaS platforms require user management actions such as adding users (provisioning), removing users (deprovisioning), and monitoring user activity. However, a significant segment of the SaaS ecosystem does not expose APIs for user management and provisioning.

Several structural and strategic factors drive this gap:

- **Legacy Architecture & Technical Debt**

Many SaaS applications were built before API-first design became standard. Their backends tightly couple UI and business logic, making it costly to expose clean API endpoints without major rewrites.

- **Market Positioning & Monetization**

API access is often gated behind enterprise-tier plans. Smaller or mid-market SaaS vendors deliberately withhold API functionality to drive upsells or prevent commoditization of their platform.

- **Niche & Vertical SaaS Vendors**

Specialized tools (design tools, creative suites, niche CRMs) often serve SMBs and deprioritize developer-facing infrastructure. Their product roadmaps focus on UI features rather than integration capabilities.

- **Security & Compliance Concerns**

Exposing user management APIs introduces security surface area. Some vendors consciously avoid this to reduce liability, especially in regulated industries.

- **Resource Constraints**

Many SaaS startups lack engineering bandwidth to build, maintain, version, and document robust APIs alongside their core product.

### Key Challenges in UI-based Automation

- **Brittle bots:** Traditional RPA/screen-scraping breaks on even minor UI changes (CSS updates, A/B tests, JavaScript re-renders).
- **Authentication complexity:** MFA/2FA (TOTP, email OTP, push notifications), session timeouts, and anti-bot measures (Cloudflare, reCAPTCHA, behavioral fingerprinting).
- **Dynamic & varied UIs:** Pagination, infinite scroll, role-based views, pop-ups, and app-specific flows make one-size-fits-all scripting impossible.
- **Scalability & security:** Running hundreds of sessions securely, avoiding rate-limits, storing credentials compliantly, and handling CAPTCHA/bot detection at scale.
- **Maintenance overhead:** Every SaaS update requires bot fixes, creating technical debt.

### Core Insight

Even when APIs exist, real-world workflows often happen through the UI. Therefore, a reliable, scalable UI automation system is critical for CloudEagle to deliver complete user lifecycle management.

## 2. Research on Available Technologies

Research was conducted using AI chatbots (ChatGPT, Claude, Grok, Gemini) as specified in the assignment.

### Headless Browsers

- **Playwright** (recommended) – Fast, reliable, built-in stealth mode and excellent support for dynamic SaaS portals.
- Selenium – Functional but slower for modern UIs.

### RPA Tools

- UiPath and Robocorp – Strong for rule-based UI simulation but brittle when interfaces change.

### AI-driven Web Automation

- **LangChain / LangGraph** – For building stateful, multi-agent workflows with reasoning.
- **AutoGPT-style agents & OpenAI function calling** – Enable natural-language intent to drive browser actions.
- **Gemini LLM (gemini-2.0-flash) or OpenAI**– Used in the AI-driven implementation for converting natural-language tasks into structured Playwright actions.

### Authentication Handling

- Session persistence via saved browser contexts
- MFA/OTP support through manual first-time login + session reuse
- CAPTCHA solvers and human-in-the-loop fallback

The recommended stack combines **Playwright** with an **AI Reasoning Layer** (deterministic rules or Gemini LLM) for the best balance of reliability and adaptability.

## 3. Proposed Solution

### AI-Driven Web Automation Agent

A hybrid system that performs user data extraction and account management directly from SaaS UIs without APIs. It supports two implementations:

1. **Deterministic Automation** (Playwright + semantic selectors)
2. **AI-Driven Automation** (Gemini LLM + Playwright)

### Workflow for Scraping User Data

- Login via persisted session
- Navigate to Members/Users page
- Anchor on stable semantic elements (e.g., role buttons)
- Extract structured data: Name, Username, Role, Last Active

### Workflow for Provisioning and Deprovisioning

- **Provisioning**: Click "Invite Workspace members" → enter email → handle chip conversion → Send invite
- **Deprovisioning**: Locate user row → click Remove → handle confirmation modal → validate removal
- Different SaaS flows are handled through configurable intent-based actions

### Handling Challenges

- Dynamic UI → Semantic anchors + AI decision layer
- Multi-step actions → Sequential execution with validation
- UI variations → Flexible selectors (get\_by\_text, get\_by\_role)

## 4. Scalability & Automation

### Multi-SaaS Handling

- Intent layer is UI-agnostic (“Invite user”, “Remove user”)
- Anchor-based extraction on semantic elements (role buttons, action buttons)
- Works across platforms like Notion, Slack, Jira, Dropbox, HubSpot, and Trello

### Role of AI in Resilience

- Deterministic: High reliability with rule-based reasoning
- AI-Driven: Gemini LLM converts natural-language intent into actions, enabling self-healing and adaptability to unseen UI changes

The system scales horizontally and integrates directly into CloudEagle’s dashboard.

## 5. Testing & Proof of Concept

**Tested on:** Trello Workspace (real SaaS admin portal)

### POC Implementation

Two approaches were tested on the same Trello workspace:

1. **Deterministic Automation** – Pure Playwright with semantic selectors
2. **AI-Driven Automation** – Gemini LLM for task-to-action mapping + Playwright execution

```
C:\Users\Dell\Desktop\trello-demo>python main.py
Login and open Workspace → Members page, then press Enter...

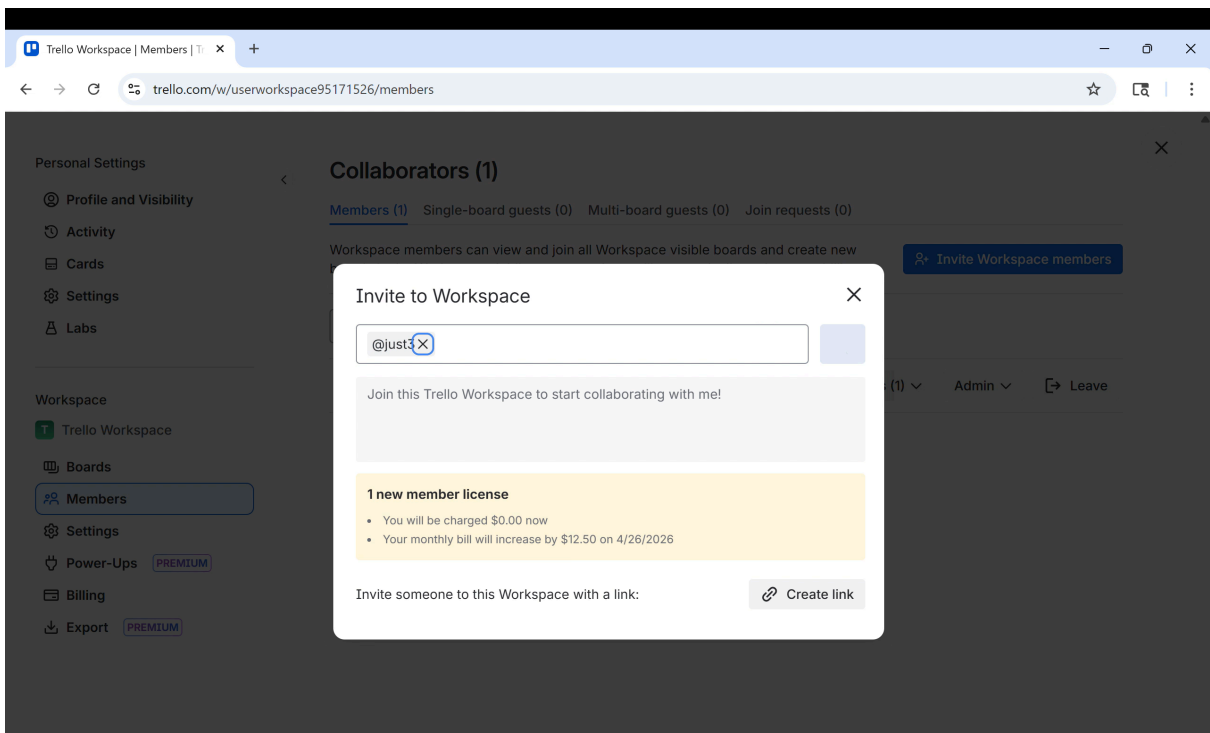
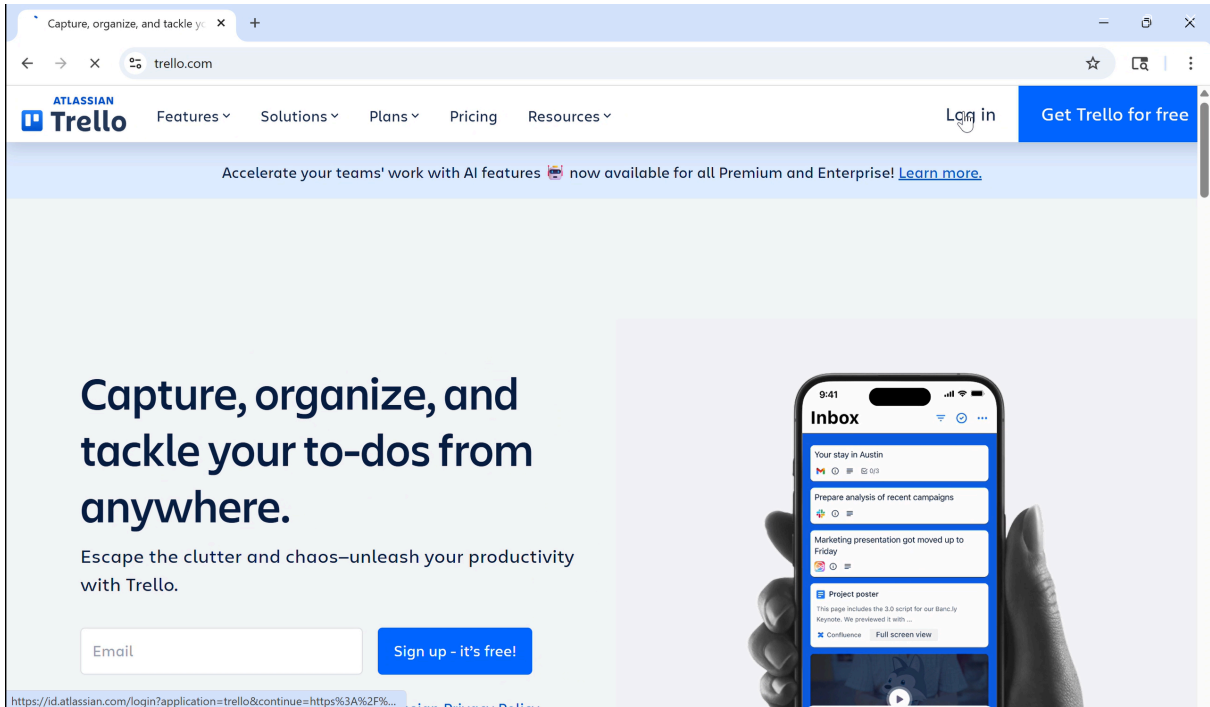
--- STEP 1: SCRAPING USERS ---
Found 1 user(s)

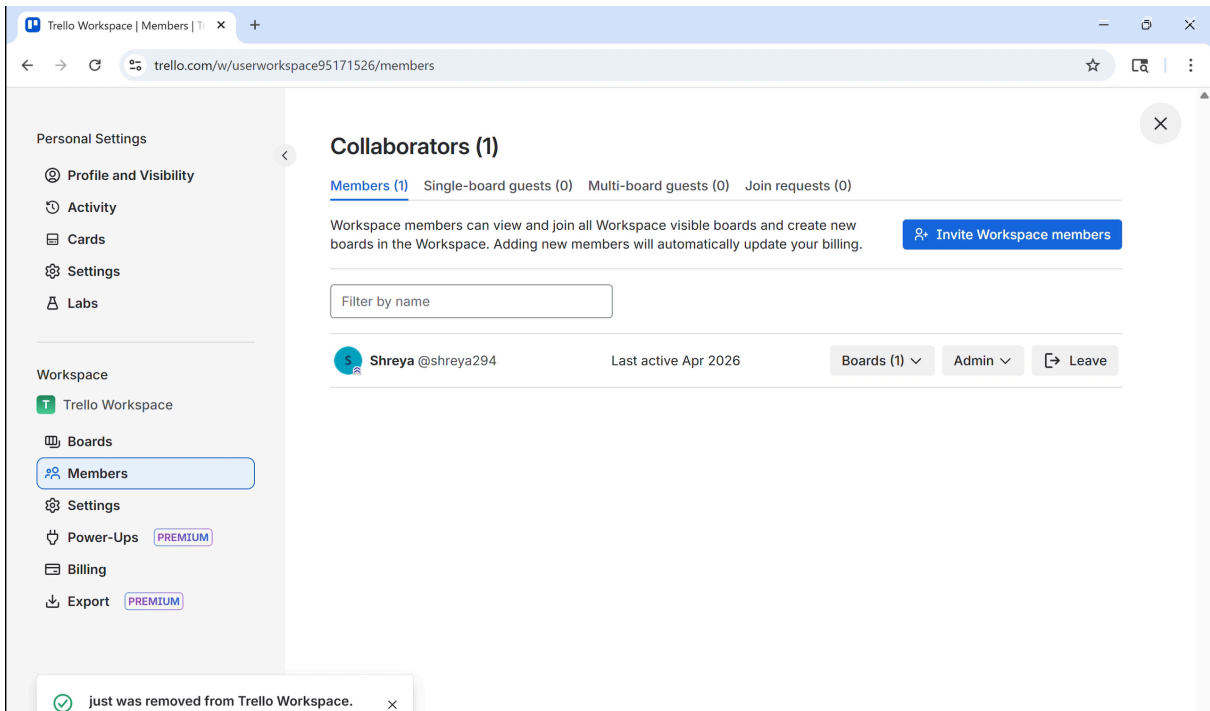
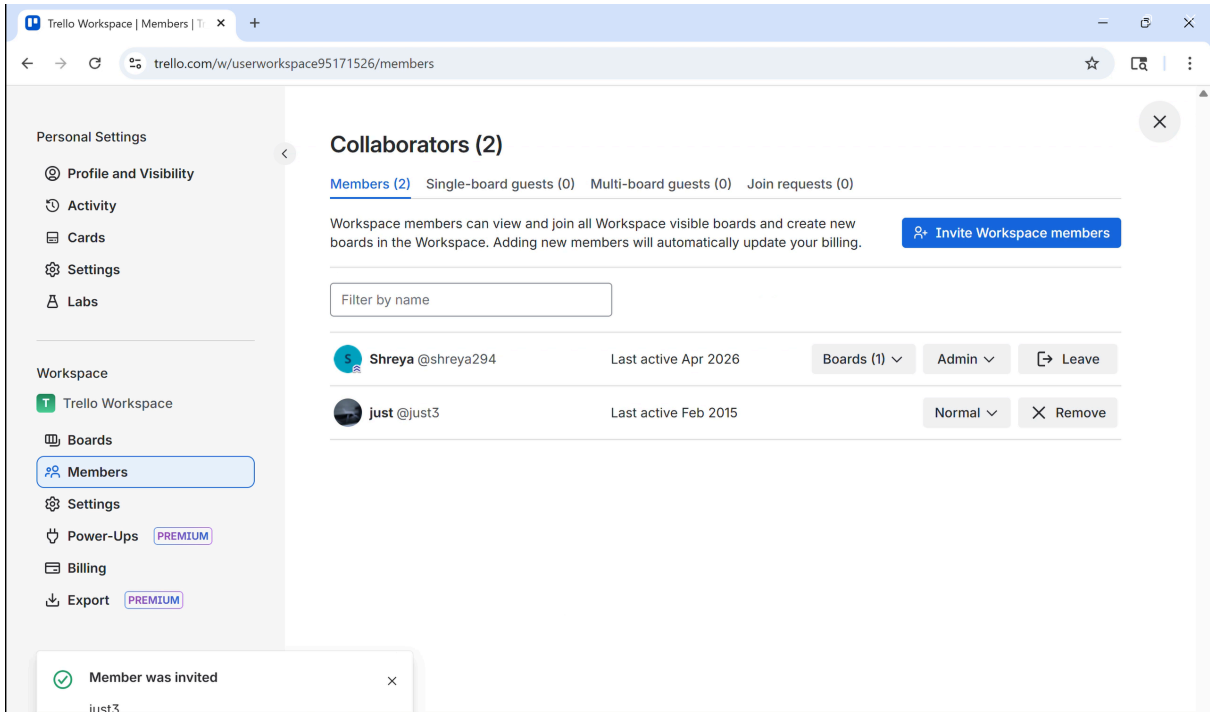
User Data:
Name: Shreya
Username: @shreya294
Role: Admin
Last Active: Last active Apr 2026
Email: Not available via UI
-----

--- STEP 2: INVITING USER ---
Invite triggered

--- STEP 3: REMOVING USER ---
User found: @just3
Confirming removal
User removed successfully

Press Enter to close...
```





### What Worked

- Accurate user data scraping (Name, Username, Role, Activity)
- Full invite workflow automation
- Deprovisioning with modal handling and post-action validation

### POC Results

- Extracted user: Shreya (@shreya294, Admin, Last Active)
- Successfully completed invite and remove cycles

### What Didn't Work

- Email extraction (hidden by Trello privacy settings)
- Fully automated login (manual first-time due to OTP; session saved thereafter)

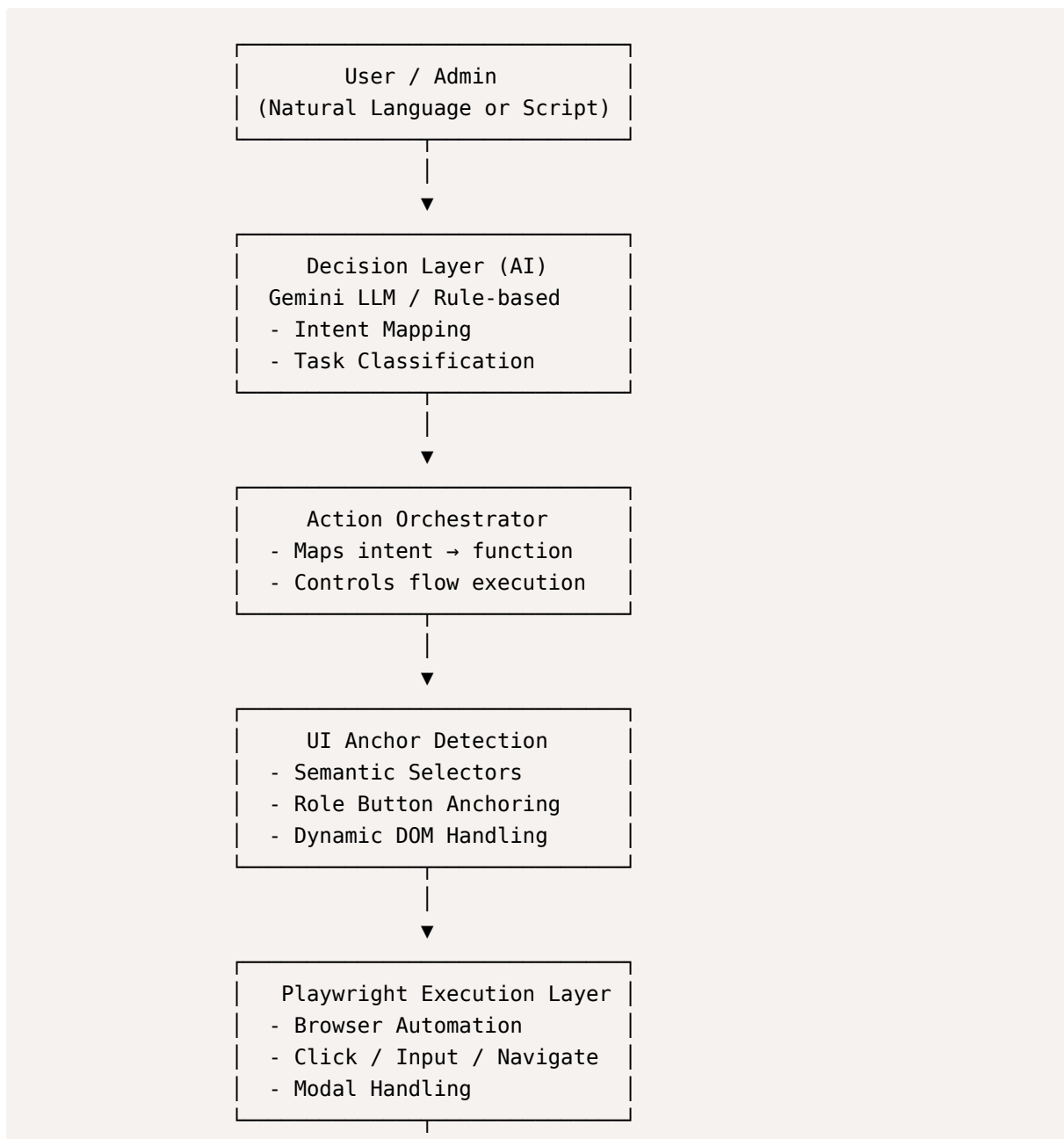
### Challenges Faced & Solutions

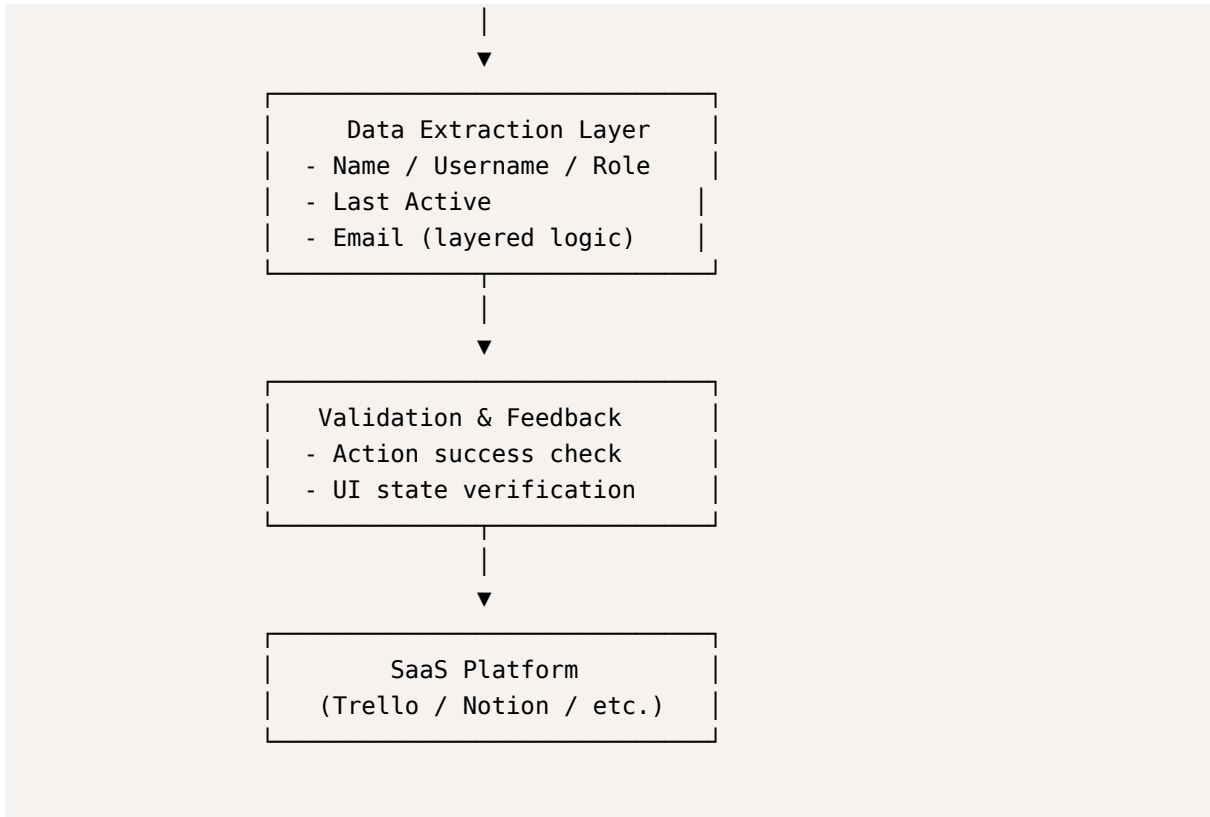
- Dynamic DOM → Solved with semantic anchors (button:has-text('Admin'))
- Multi-step flows → Handled via sequential validation
- Selector instability → AI reasoning (Gemini) or rule-based fallback

## 6. Python Code : <https://github.com/shreyaup12/saas-ui-automation>

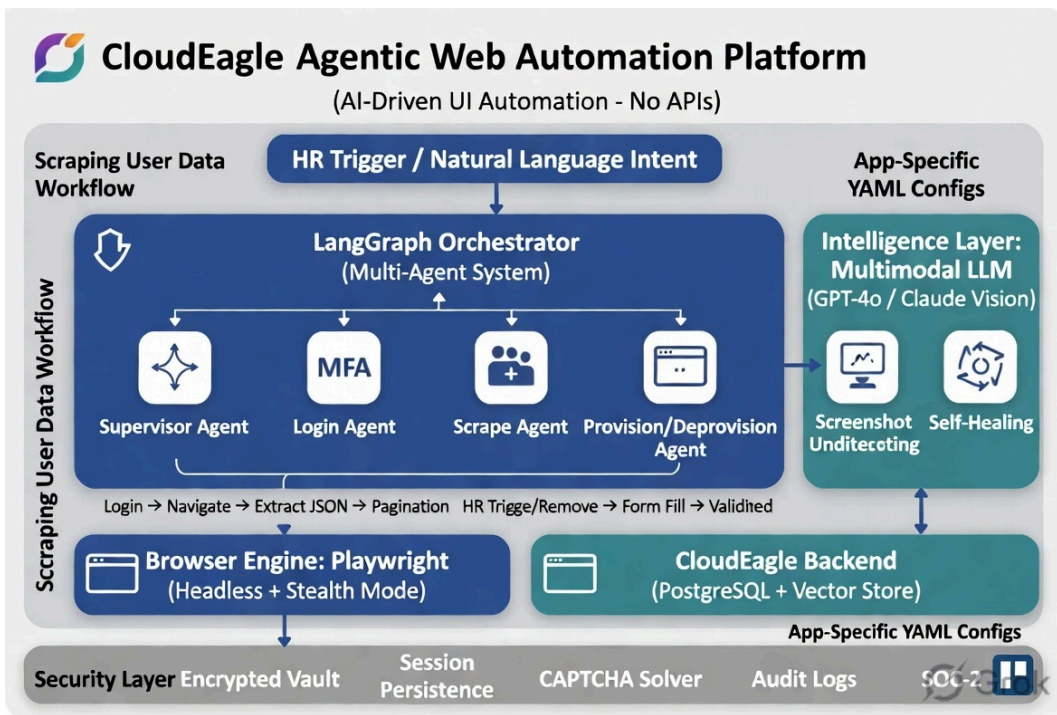
1. **Deterministic Automation** – Pure Playwright with semantic selectors
2. **AI-Driven Automation** – Gemini LLM for task-to-action mapping + Playwright execution

### AI-Driven Extension (Gemini LLM)





High level System Architecture with LangGraph



**Comparison**

Feature	Deterministic	AI-Driven (Gemini)
Reliability	High	Medium (LLM dependent)
Flexibility	Low	High

Feature	Deterministic	AI-Driven (Gemini)
Scalability	Medium	High
Debuggability	Easy	Moderate

## Conclusion

This project demonstrates a practical and scalable approach to automating SaaS user management purely via UI. By combining Playwright with semantic anchors and an AI reasoning layer (Gemini LLM), the system moves from rigid scripts to intelligent, adaptive agents. The Trello POC validates immediate value for CloudEagle and highlights the path to production across multiple SaaS platforms.