

# Instalily GTM Engineer Case Study: GenLeads.ai

Nicole Hao

May 15, 2025

## 1 General Approach & Goals

For the GTM engineer case study at Instalily AI, I prototyped **GenLeads.ai**. GenLeads.ai is an AI-powered lead generation and outreach platform for DuPont Tedlar's Graphics Signage team. The system automates finding and qualifying sales leads by researching industry events and trade associations where potential customers might be present.

I started with design an end-user workflow (how the user would interact with the AI agent) and AI agent workflow (how the agent responds to requests) based on the description, with an emphasis on scalable automation, so that one day it can be expanded to lines of business even beyond DuPont Tedlar's Graphics & Signage, for different sets of ICP that can be updated by the end-user. And more importantly, I assumed that as a GTM engineering, we want to achieve the following goals within **48 hours**:

- **Present** an interactive prototype to clients
- **Define** future steps, what can be added
- **Communicate** the implementation results both verbally and in written form **clearly** and in a **concise** fashion to clients and senior leadership.

Then, based on the end-user and ai agent workflows, I designed the UX first by sketching, then implemented the frontend with cursor using React with TypeScript, TailwindCSS, shadcn/ui components. Backend is in Node.js with Express. For more specifics of the tech implementation, refer to DOCUMENTATION.md. I went through multiple iterations to continue adding features and improve upon my design.

## 2 End-User and AI Agent Workflows

Instead of designing only the AI agent's workflow, I integrated the end user's actions with the AI agent workflow since the purpose is to assist our clients and create a 'human-in-the-loop' work process.

### 2.1 End-user Workflow

This focuses on how the user (e.g., Sales Ops, BDR, or PMM) would interact with the system:

1. **Login & ICP Selection:** User logs into the platform and selects or creates a relevant *Ideal Customer Profile*, e.g., "Architectural Signage Companies, \$10M–\$500M revenue, located in North America.". I designed a button that leads to a window for the end-user to add and tweak their ICP criteria for stability and future use purposes.
2. **Upload or Choose Target Event:** User selects from pre-loaded events/trade shows or uploads a CSV/list of companies/events (from web search, updated weekly).
3. **Configure Filters & Target Personas:** User specifies key decision-maker roles (e.g., "Director of Product," "VP of Innovation").
4. **Trigger Lead Generation:** User clicks "Generate Leads" — backend AI agent starts enrichment and lead qualification. Generate a list of lead profiles with corresponding decision maker's LinkedIn information. Clicking on each profile leads to a dashboard with graphs and stats showing how the company fits the ICP.
5. **Review Output Dashboard** User views a structured dashboard with:
  - Event → Company → Stakeholder → Reason for Fit → Key decision maker + window to generate and edit Personalized Outreach Message (with sources cited). User can approve, edit, or reject any AI-drafted outreach message.
6. **Export or Push to CRM** Export as CSV or auto-send to tools like HubSpot/Salesforce.
7. **Save ICP Template** Save new ICP for future campaigns across other lines of business.

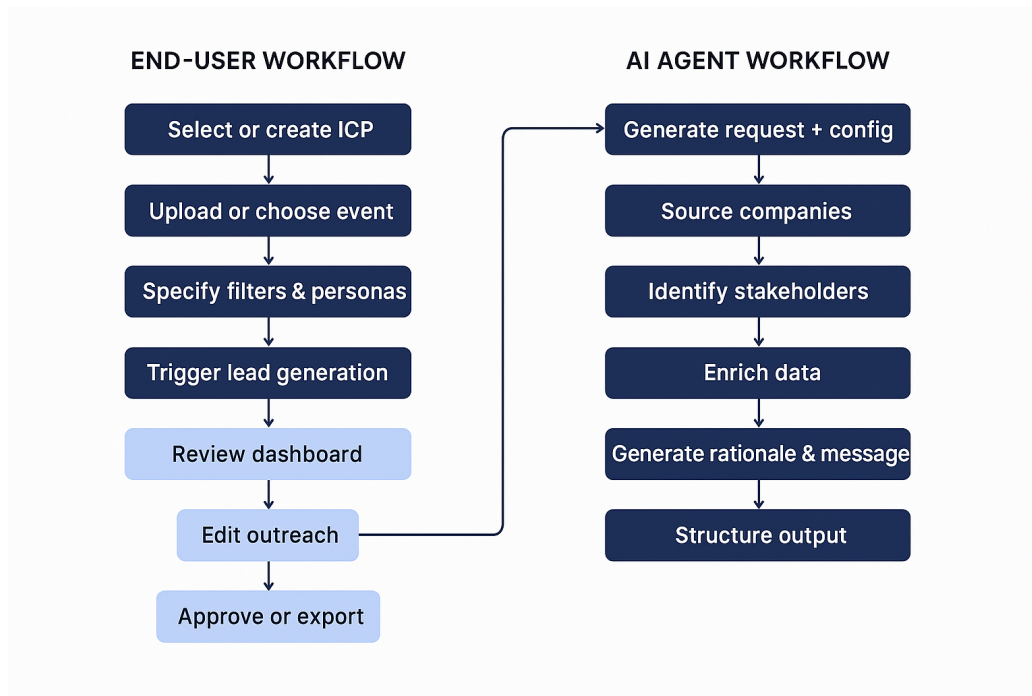
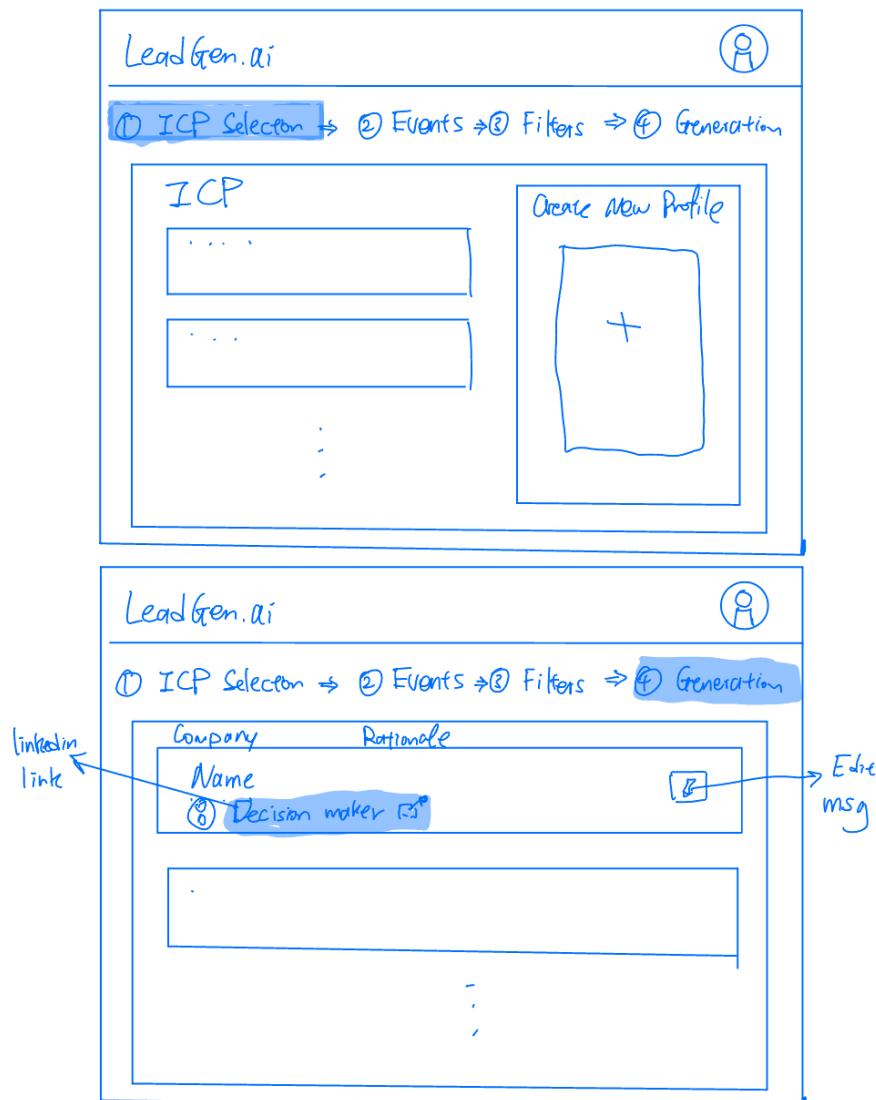


Figure 1: End User & AI Agent Workflows

## 2.2 AI Agent Workflow

1. **Receive Request + ICP Config:** Accepts structured input: event name, ICP filters (industry, size, geography), and target personas.
2. **Company Sourcing:** Scrapes company lists from event websites, trade associations, or uploaded files. Filters companies based on ICP using public data (Crunchbase, LinkedIn, Clearbit, etc.).
3. **Stakeholder/Decision maker Identification:** For each company, uses LinkedIn Sales Navigator or Clay API to find decision-makers matching the user's persona filters.
4. **Lead Enrichment:** Gathers info such as company size, market activity, recent news, or tech stack.
5. **Rationale Generation:** Uses GPT-4o to explain why the company is a match based on ICP + enriched data.
6. **Outreach Message Drafting:** Prompts LLM to generate a brief, tailored outreach message for each stakeholder, referencing event, company, and rationale.
7. **Data Structuring + Error Handling:** Ensures clean JSON/table format, with null checks, deduplication, and scoring.
8. **Serve Output:** Returns structured output to the frontend dashboard API for display and export.

## 2.3 Initial UX Design



Here's a sketch of my UX design ( Normally if I had more time, I'd do it in figma just so that it looks nicer, but a sketch conveys the idea as well!)

## 3 Data Processing Steps

For more details and code examples, see DOCUMENTATION.md in the github repo. Firstly, to use **real data**, the system collects data from multiple sources as shown in server/services/eventScraper.ts. It **scrapes event information** from industry websites like FESPA, ISA, and other sources. The scraper extracts event details including names, dates, locations, and exhibitor counts. This data is stored in the database using the schema defined in shared/schema.ts.

As seen in server/services/openai.ts, the system then processes lead generation requests through several steps:

- Takes input parameters including ICP profile, event details, and personas
- Validates the OpenAI API key
- Creates a detailed prompt for the AI model & Creates a detailed prompt for the AI model
- Calls the OpenAI API with the GPT-4o model
- Processes the response to generate structured lead data including company info, stakeholder details, and match scores

If you turn on the data enrichment option, then the getCompanyEnrichment function in openai.ts will gather technology stack information from the internet, then retrieve funding information, as well as collecting recent news and identifying competitors. All enrichment data is gathered using validated and formatted according to the

The system also calculates match scores based on multiple criteria as defined in the MatchDetails interface. To better align with the example of a good lead in the case study description, I added the following components that contribute to the match score:

- Industry relevance
- Product fit
- Decision-making authority
- Budget alignment
- Geographic match
- Company size validation (e.g. num of employees)

## 4 Implementation Results

As mentioned, the platform integrates React with TypeScript for the frontend and Node.js/Express for the backend, creating a seamless workflow for lead generation. The final implemented system successfully includes comprehensive ICP targeting, event research capabilities, and strategic filtering through OpenAI integration.

Key achievements include

- the implementation of personalized outreach generation, optional lead enrichment with company data, and a review system
- scrapes event data from industry sources like FESPA and ISA, processes it through AI qualification using GPT-4o, and generates detailed lead profiles with match scores and personalized outreach messages
- error handling for API issues, data validation through Zod schemas
- a responsive UI built with shadcn/ui components and TailwindCSS
- data integrity through proper source citations and implements comprehensive security measures for API access and data protection

I believe the best way to showcase the implementation result is through a video. So I made a loom walk-through. Refer to the github repo for the demo with my verbal explanation.

**Thank you for taking the time to read this report!**