

Multi-Agent AI System for Performance Marketing Teams

This document provides a comprehensive overview of a multi-agent artificial intelligence (AI) system designed for performance marketing teams. The system combines data ingestion from YouTube, LLM-driven research, persona development and creative hook generation to empower marketers with data-driven insights and reusable assets.

Overview

The AI Ad Research & Persona System harnesses multiple agents and tools to automate the discovery of niche-specific video content, analyse advertising techniques and generate buyer personas with tailored ad hooks. Data is stored in a PostgreSQL database (Neon) for easy reuse. The app operates in two main steps: ingesting YouTube data and generating personas/hooks from the database.

Core Capabilities

- **YouTube niche research:** constructs multiple search queries and fetches top videos for the selected niche.
- **Ad analysis:** examines winning ads, recurring hooks and main angles using a large language model (LLM).
- **Persona & hook generation:** synthesises buyer personas and creative hooks from the research results.
- **Persistent storage:** saves research data, personas and hooks in a serverless Postgres database for later reuse.

Workflow

1. YouTube Research

This flow is triggered when a user clicks the “Search YouTube & Save” button. The system performs several steps to collect and analyse relevant content:

1. **User input:** the user enters the niche, language and country in the global input section.
2. **Query generation:** the **Query Builder Agent** generates multiple YouTube search queries tailored to the niche and locale.
3. **Video retrieval:** the **YouTube Search Tool** uses the YouTube Data API v3 to return the top videos for each query. The API’s `search.list` endpoint returns a collection of videos, channels and playlists matching the query
4. **Research:** the **Research Agent** (using an LLM) examines video metadata to identify competitors, example advertisements, recurring hooks and primary angles. Research

results are stored in a `videos_json` and `research_json` column in the `niche_runs` table.

The following image illustrates the user interface for global inputs and the YouTube search trigger:

The screenshot shows a dark-themed web interface titled "AI Ad Research & Persona System". Below the title, there is a two-step instruction: "Step 1 (optional): run YouTube research and save it to the database. Step 2: generate personas and hooks from saved data for any niche." The interface is divided into two main sections. The first section, "Global inputs", contains a text input for "Niche for YouTube search" with the value "camping with campervan", a dropdown for "Language" set to "EN", and a dropdown for "Country" set to "US". The second section, "1) Search YouTube & save to Neon", includes a small blue button labeled "calls YouTube API", a text description of the process, and a text input for "Videos per query (optional)" with the value "3". At the bottom of this section is a large blue button with a play icon and the text "Search YouTube & Save".

User interface – global inputs and YouTube ingest

2. Personas & Hooks

This flow is activated when a user selects a niche and clicks the “Run analysis from DB” button. It generates personas and hooks based on previously saved YouTube research:

1. **Niche selection:** the frontend calls the backend to retrieve distinct niches from the database, populating a dropdown menu.
2. **Cache retrieval:** when a niche is selected, the backend retrieves the latest research row for that niche. If personas and hooks have already been generated (`personas_json` and `creatives_json` are present), the cached results are returned.
3. **Persona creation:** if no cached personas/hooks exist, the Persona & Targeting Agent uses the research data to create between three and five buyer personas. Each persona includes pains, desires and tones tailored to the niche.
4. **Hook generation:** the **Creative Agent** generates several hooks per persona, crafting persuasive marketing angles based on the identified pain points and desires.
5. **Persistence and return:** the new personas and hooks are saved back to the database and returned to the frontend for display.

The following images show the persona cards and hooks generated by the application:

Result (personas & hooks)

Run ID: 3 · generated now

Personas (5)

Alex, 29, solo outdoor adventurer

29 ·

Alex is an urban professional who loves weekend trips alone in nature. He values freedom and exploration, and wants a campervan to escape city life easily.

Pains: Finding a compact and versatile campervan that is easy to drive and park, Lack of clear information on essential gear for solo camping, Overwhelmed by campervan conversion options and technical details

Desires: A simple, stealthy campervan that blends in, Freedom to explore national parks and scenic routes at his own pace, Practical tips and hacks to optimize van life solo

Jessica, 38, family trip planner

38 ·

Jessica is a busy mom of 3 who organizes family vacations. She wants to find a campervan setup that suits a family and keeps everyone comfortable and entertained.

Pains: Limited campervans that fit 4+ people comfortably, Safety and ergonomic concerns for kids during travel, Balancing space for sleeping, eating, and playing

Persona cards

Hooks (10)

Alex, 29, solo outdoor adventurer: Ready to roll with a stealthy van that's made for solo freedom?

Alex, 29, solo outdoor adventurer: Discover must-have solo campervan gear for your next trip.

Jessica, 38, family trip planner: See how a family campervan fits everyone comfortably and safely.

Jessica, 38, family trip planner: Find the best family-friendly campervan rentals without hassle.

Chris, 45, aspiring campervan buyer: Get the real costs behind luxury campervan ownership.

Chris, 45, aspiring campervan buyer: Explore innovative two-story campervan designs and their value.

Morgan, 32, van life gear enthusiast: Check out game-changing campervan hacks to save space and time!

Morgan, 32, van life gear enthusiast: Must-have, affordable gear for every van lifer's toolkit.

Taylor, 26, first-time campervan renter: Rent a luxury van that feels like a hotel suite on wheels.

Taylor, 26, first-time campervan renter: Easy campervan rentals for your first unforgettable trip.

Hook list

Architecture

Frontend

The client side is built with **HTML**, **CSS** and **vanilla JavaScript**. The UI is simple and intuitive: users specify a niche, language and country, trigger YouTube ingestion and later select a niche from a dropdown to generate personas and hooks. The interface displays the resulting personas in cards and lists the hooks underneath.

Backend

The backend is implemented in Node.js using the Express framework. Node.js provides an *event-driven, asynchronous* architecture capable of handling many input/output operations and real-time web applications. Express adds a minimal and flexible layer, offering robust

routing, HTTP helpers and middleware for building web applications and APIs. The backend exposes four endpoints:

- GET /api/health – simple health check.
- GET /api/niches – returns distinct niche values for the dropdown.
- POST /api/ingest-youtube – triggers the YouTube research flow by calling the query builder, YouTube search and research agents.
- POST /api/analyze – triggers the persona and hooks generation flow.

Agents & Tools

The system leverages specialised agents and tools to modularise tasks:

- **Query Builder Agent** – constructs search queries from the user's niche input.
- **YouTube Search Tool** – wraps the YouTube Data API v3, which returns collections of videos, channels or playlists matching the search parameters.
- **Research Agent** – uses a large language model (LLM) to extract competitor names, example ads, hooks and angles from the video metadata.
- **Persona & Targeting Agent** – synthesises buyer personas by analysing the research data for pains, desires and tone of voice.
- **Creative Agent** – crafts compelling hooks for each persona, focusing on problems and aspirations.

Data Storage (Neon)

Research outputs, personas and hooks are stored in a PostgreSQL database hosted on Neon. Neon is a managed, serverless Postgres platform designed to accelerate development. It allows instant provisioning, autoscaling according to load and the ability to scale down to zero. Neon also provides instant read replicas that scale down when idle and a powerful API and CLI for managing thousands of databases programmatically. By using Neon, the application achieves efficient scaling and cost-effective persistence without managing servers.

Tech Stack

The system employs modern web technologies and cloud services to deliver a cohesive solution:

- **JavaScript** (Node.js and ES modules) for server logic and agents. Node.js's event-driven architecture optimises throughput and scalability.
- **Express** – a fast, minimalist web framework that offers robust routing and middleware without hiding Node.js features.

- **PostgreSQL (Neon)** – serverless database with autoscaling and branching capabilities.
- **OpenAI Chat Completions** – LLM API that provides a universal text input/output interface for a variety of tasks, returning a completion for each text prompt.
- **YouTube Data API v3** – used by the YouTube search tool to fetch videos matching query parameters.
- **HTML/CSS** – static structure and styling for the user interface.
- **Vanilla JavaScript** – client-side scripts for handling user events and fetching data from the backend.

Conclusion

This multi-agent AI system streamlines performance marketing by automating research, persona development and creative ideation.

By combining the flexibility of Node.js and Express with the scalability of Neon and the intelligence of OpenAI models, the application empowers marketers to base their campaigns on data-driven insights.

The modular architecture ensures each component can evolve independently, enabling future improvements such as new data sources or refined persona models.