

Two Agents Need to Build

Agent Name	Purpose
Agent 1: Content Generation Agent	Takes a high-level brief and generates ad copy, captions, emails, visual prompts , etc.
Agent 2: Marketing Automation Agent	Builds complete automation campaigns (e.g., welcome flows, cart abandonment, post-purchase) with copy, visuals, logic, and integration-ready assets (e.g., Klaviyo flow)

Agent 1: Content Generation Agent

Input:

"Launching an eco-friendly shampoo. Target: Gen Z moms. Generate a launch kit."

Output:

- Ad copy (headlines, primary text, CTAs)
- Social captions (Instagram, TikTok)
- Email creative (hero, CTA, body)
- Image prompts (for text-to-image)
- UGC video script (optional)
- Short-form video plan (optional)

Tasks to Plan & Execute:

1. Classify content needs from prompt
2. Generate ad copy
3. Generate captions
4. Generate email creative
5. Generate image prompts
6. Generate video script
7. Export all in a package

Tools to Use:

- HuggingFace LLM for text
 - Text-to-Image generation (via stable-diffusion, dalle-mini, etc.)
 - Your own modules from content_generator.py, prompt_templates.py, etc.
-

Agent 2: Marketing Automation Agent

Input:

"Here's the website: www.brand.com. Create a 7-email + 2-SMS cart abandonment flow."

Output:

- Email & SMS copy (subject, body, CTA)
- Campaign logic (timing, triggers, segments)
- Visuals (headers, banners)
- Structured flow in JSON or ready-to-upload Klaviyo format
- Editable with feedback

Tasks to Plan & Execute:

1. Understand brand from website or prompt

2. Select best-practice flow (e.g., cart abandonment, welcome, post-purchase)
3. Generate sequence of messages (tone-matched)
4. Add visuals (prompt + generate)
5. Structure campaign for Klaviyo/Hubspot
6. Package output for preview or upload

Tools to Use:

- HuggingFace LLM or LLaMA (copywriting)
- Image generation (AI banner/header)
- Your export_manager.py for formatting
- Optional: Klaviyo API or mock export

How to Build Both with LangGraph

Each agent should be a **LangGraph workflow** with:

- Input Node
- Planner Node
- Tool Execution Nodes (one per task)
- Memory (optional per brand)
- Feedback Node (optional)
- Final Export Node

Folder structure

```

agentic_ai_project/
|
|—— app.py                # Main entrypoint (e.g., Streamlit or CLI app)
|—— config/               # Configurations (API keys, constants)
|   |—— settings.py
|   |—— env.example
|
|—— agents/               # Agent logic (LangGraph or CrewAI)
|   |—— __init__.py
|   |—— marketing_automation_agent.py
|   |—— content_generation_agent.py
|
|—— workflows/           # Agent workflows / LangGraph nodes
|   |—— marketing_automation/  # Nodes specific to marketing flow
|       |—— planner.py
|       |—— email_generator.py
|       |—— sms_generator.py
|       |—— visual_generator.py
|       |—— flow_builder.py
|   |—— content_generation/    # Nodes for content generation

```

```
|   |—— ad_writer.py
|   |—— video_editor.py
|   |—— static_image_gen.py
|   |—— ugc_script_gen.py
|   |—— asset_packager.py
|
|—— tools/                # Tools/utilities like LLMs, APIs, Translators
|   |—— llm_manager.py      # Handles HuggingFace/OpenAI/Groq LLMs
|   |—— image_gen.py        # Image generation (e.g., stable diffusion)
|   |—— video_editor_api.py # Interface with Opus Clip, Captions
|   |—— klaviyo_exporter.py # Export campaign to Klaviyo format
|   |—— browser_utils.py    # Brand scraping, info extraction
|
|—— prompts/              # All prompt templates
|   |—— __init__.py
|   |—— email_prompts.py
|   |—— ad_prompts.py
|   |—— brand_analysis_prompts.py
|
|—— export/               # Exported flows (json, csv, zip)
|   |—— sample_campaigns/
|       |—— demo_flow_1.json
|       |—— demo_emails.csv
|       |—— demo_images.zip
|
|—— memory/               # Saved history (ChromaDB or simple JSON)
|   |—— brand_1_memory.json
|   |—— brand_2_memory.json
|
|—— tests/                # Unit + integration tests
|   |—— test_agents.py
|   |—— test_email_gen.py
|   |—— test_flow_builder.py
|
|—— docker/               # Docker + container setup
|   |—— Dockerfile
|   |—— requirements.txt
|
```

Development + Skills

Task	Skill Required
Project setup & LangGraph agent wiring	Intermediate Python
Email/SMS copywriter module	Prompt Engineering, NLP
Brand extractor / tone analyzer	LangChain + scraping
Image generator module	Diffusion API + frontend
Export builder (e.g., Klaviyo format)	Python logic + JSON mapping
Testing + Revisions	QA + debugging
Streamlit/UX wrapper	Streamlit, UI/UX
Content agent modules (ad gen, UGC script, caption, etc.)	NLP + multimodal workflow
LangGraph retry/edit loop logic	LangGraph or CrewAI design

Cost

1. Infrastructure & APIs

Item	Free Option	Paid Option (if needed)	Approx. Monthly
LLMs	HuggingFace free-tier, Groq free-tier, OpenRouter	OpenAI (GPT-4), Claude	\$0–\$50
Image Gen	stabilityai/stable-diffusion, deepfloyd/IF, or Replicate API (limited free)	Midjourney or RunwayML	\$0–\$25
Video Editing	Opus Clip free plan, Captions app APIs	Opus Clip Pro, Captions Pro	\$0–\$20
Vector DB	ChromaDB (local) or FAISS	Pinecone (if scaling)	\$0
Automation Platform	Zapier/Make.com free tier	Zapier Pro	\$0–\$20
Email Platform	Klaviyo dev/test account (free up to 250 contacts)	Klaviyo paid	\$0–\$30
Cloud Hosting (optional)	Localhost / Streamlit Cloud / Hugging Face Spaces	AWS / GCP / Railway	\$0–\$30
Browser Tools	Playwright, Selenium, or unstructured.io	None needed	\$0

Total Infra/Tool Cost Estimate:

- Minimum (Free): \$0
- Realistic MVP Range: \$20 – \$75/month (based on API usage)