Approved Content Variant Generator – Team Introduction Brief

Welcome

Welcome to the Approved Content Variant Generator experiment. This brief provides everything you need to get started on this focused development spike. We're excited to work with you to explore how AI-assisted tooling can help marketing teams efficiently generate channel-appropriate content variants while maintaining consistency, control, and reviewability.

Project Overview

Background

Within large pharmaceutical organizations, marketing teams are required to produce content for multiple channels and audiences at a rapid pace. In most cases, this work does not involve creating new messaging, but rather adapting already approved content into formats suitable for different channels. Today, this adaptation process is manual, repetitive, and time-consuming, increasing operational overhead and slowing content delivery.

The Opportunity

This experiment explores whether an internal AI-assisted tool can streamline the creation of channel-specific content variants from approved source material. The goal is to accelerate content production while ensuring messaging remains consistent, traceable, and reviewable, without introducing uncontrolled or non-compliant changes.

What We're Building

An internal **Approved Content Variant Generator** that can:

- Store and manage approved source content
- Generate multiple channel-appropriate variants from approved messaging
- Apply basic guardrails to prevent unsafe or non-compliant outputs
- Support a lightweight internal review workflow
- Enable simple export of finalized variants

Technical Context

Proposed Technology Stack

- **React and/or Next.js** for the web application frontend
- **Supabase** for authentication, data storage, and asset management
- **n8n** for workflow orchestration and LLM interaction
- **LLM access via approved client tooling**, such as Requesty

Sample Use Cases

The tool will support common internal marketing workflows, including:

1. Generating email, web, or SMS variants from approved brand content
2. Adapting messaging for different audiences such as HCPs, patients, or internal teams
3. Preparing draft content for lightweight internal review prior to formal MLR processes

---

Experiment Scope & Timeline

Duration

This experiment is designed to be completed within a short, focused timeframe and can be worked on incrementally during spare time. The suggested timeline spans multiple weeks but can be adapted based on availability.

This is an exploratory spike intended to validate feasibility and usability rather than deliver a finished product.

What This Is

- A proof of concept for an internal marketing enablement tool
- A learning exercise to test AI-assisted content adaptation workflows
- An opportunity to explore guardrails, traceability, and usability considerations

What This Is Not

- A production-ready marketing system
- A replacement for formal MLR or approval workflows
- A fully hardened enterprise platform

Success Criteria

At the conclusion of the experiment, we aim to understand:

1. **Technical Feasibility**
   Can React/Next.js, Supabase, n8n, and approved LLM tooling be effectively combined to support this workflow?
2. **Quality of Output**
   Do the generated variants remain faithful to approved content while being appropriately adapted for each channel?
3. **Usability**
   Does the tool feel like a realistic internal application that marketing teams could use?
4. **Safety and Guardrails**
   Are basic safeguards effective at preventing unsafe or non-compliant language?

Deliverables

By the end of the experiment, we expect the following:

1. Working Prototype

A functional prototype demonstrating:

- An approved content library
- Variant generation by channel, audience, and tone
- A basic review workflow
- Export of generated content

The prototype does not need to be polished, but should clearly illustrate the end-to-end workflow.

2. Knowledge Sharing

A discussion or walkthrough covering:

- What worked well
- Challenges encountered
- Design and technical decisions made
- Any unexpected learnings

3. Findings Documentation

Lightweight documentation capturing:

- The technical approach taken
- Feasibility and limitations
- Quality of generated outputs
- Recommendations for future iterations

This documentation can be informal and will be stored for future reference.

4. Technical Assets

Any reusable code, configurations, schemas, or workflows created during the experiment, including:

- GitHub repository with setup instructions
- Seed data scripts for mock approved content
- Screenshots or a short demo recording

Team & Communication

Communication Approach

- Direct communication is encouraged
- Questions should be raised as they arise
- Formal status reporting is not required
- Informal syncs and check-ins are welcome

Access & Data

System Access

No real client or confidential data should be used. This experiment must rely entirely on mock or synthetic content.

Data Approach

Sample data will include:

- Mock approved content
- Simulated claims, references, and audiences
- Example channels and tones

Your Setup

You are expected to provision:

- LLM access via approved tooling
- Supabase instance
- n8n instance

Getting Started

Immediate Next Steps

1. Review this brief and project context
2. Set up your development environment and tooling
3. Seed mock approved content
4. Begin building and experimenting

Questions to Consider

- How do we balance flexibility with safety?
- What guardrails are essential versus optional?
- What makes a variant “review-ready”?
- How do we ensure traceability back to source content?
- What would be required to harden this for enterprise use?

Integration Vision (Future State)

If successful, this capability could evolve into an internal marketing enablement platform, supporting:

- Faster content adaptation across channels
- Improved consistency and traceability
- Reduced operational burden on marketing teams

This experiment is focused on validating the core concept. Integration and enterprise hardening would be considered in future phases.

Support & Resources

Available to You

- Sample data and use cases that you can source from publicly available content
- Flexibility in technical decision-making
- Freedom to experiment within defined guardrails

What We Need From You

- Proactive communication if blocked
- Honest assessment of what works and what doesn't
- An experimentation-focused mindset
- Clear documentation of learnings

Closing Thoughts

This is an experiment, and the goal is learning. We do not expect perfection, but we do expect insight. Whether the outcome validates the approach or highlights challenges, both are valuable as long as they inform next steps.

Thank you for contributing to this experiment and helping explore how AI can responsibly support marketing workflows.

Reference Documentation of Suggested Technologies

Suggested Tooling

All applications built as part of this program should try and use the following tools to begin with:

- **VS Code with Roo Code** for AI-assisted development
- **Requesty**, preferably with **Claude Sonnet or Claude Opus**, for LLM access
- **MCP Servers** for exposing structured tools and data to LLM workflows
- **n8n** for workflow orchestration and automation
- **Supabase** for authentication, database, storage, and access control
- **shadcn/ui** for building clean, consistent internal user interfaces
- **Coolify** is an open-source, self-hosted platform that lets you deploy and manage applications, databases, and services on your own infrastructure with a developer-friendly experience similar to Heroku or Vercel. (**OPTIONAL**)
- **Bolt or Lovable**

VS Code & Roo Code

- VS Code documentation: https://code.visualstudio.com/docs
- Extension marketplace overview: https://code.visualstudio.com/docs/editor/extension-marketplace
- Roo Code documentation: https://docs.roocode.com

Requesty

- Requesty documentation: https://docs.requesty.ai
- Core concepts (providers, routing, policies): https://docs.requesty.ai/concepts
- API reference: https://docs.requesty.ai/api

Claude (model behaviour reference)

- Claude models overview: https://platform.claude.com/docs/models
- Prompting guide: https://docs.anthropic.com/claude/docs/prompting

MCP (Model Context Protocol)

- MCP overview: https://modelcontextprotocol.io
- MCP specification: https://modelcontextprotocol.io/specification
- Examples and SDKs: https://modelcontextprotocol.io/examples

n8n

- n8n documentation: https://docs.n8n.io
- Core concepts: https://docs.n8n.io/getting-started/core-concepts/

- Built-in nodes and integrations: https://docs.n8n.io/integrations/builtin/core-nodes/

Supabase

- Supabase docs: https://supabase.com/docs
- Authentication: https://supabase.com/docs/guides/auth
- Database (Postgres): https://supabase.com/docs/guides/database
- Storage: https://supabase.com/docs/guides/storage
- Row Level Security: https://supabase.com/docs/guides/auth/row-level-security

shadcn/ui

- Documentation: https://ui.shadcn.com/docs
- Component library: https://ui.shadcn.com/docs/components