

Marketing Copy Generator – Technical Documentation

Created using VS Code | Flask | HTML/CSS | OpenAI/Hugging Face API (or mock)

1. Project Overview

The Marketing Copy Generator is a lightweight web application built using Python and Flask in VS Code. Its purpose is to allow users to generate customized marketing copy based on three simple inputs:

- **Product**
- **Target Audience**
- **Tone**

Once these fields are submitted, a prompt is generated and processed through a generative AI API (e.g., OpenAI or Hugging Face) or a mock model. The returned marketing content is displayed on the same page.

The tool is useful for businesses, marketers, and entrepreneurs who want to quickly create ad copy, taglines, social posts, or introductory content.

2. Technologies Used

Tool	Purpose
Python 3.11	Backend development
Flask	Lightweight web framework
HTML + Jinja	Template rendering
CSS	Styling and UI design
OpenAI / Hugging Face API (optional)	Content generation
VS Code	Code development environment
JSON	Data handling for prompt templates

3. Folder Structure

```
csharpCopy codeproject-folder/  
|  
|—— app.py          # Main Flask backend |—— templates/
```

| |—— index.html # Frontend HTML |—— static/
| |—— style.css # Custom CSS and background image |—— prompt_templates.json #
(Optional) Structured prompt templates

4. Prompt Engineering Methodology

To ensure varied and rich outputs, we designed **five prompt templates** with different copywriting goals:

1. Product Launch Intro

“Introducing {product} – the perfect solution for {audience}. Enjoy a {tone} experience.”

2. Social Media Post

“Say hello to {product}, built for {audience}. Time to make a {tone} impact!”

3. Ad Headline/Tagline

“{product}: Because {audience} deserves a {tone} choice.”

4. Short Promotional Copy

“Looking for something {tone}? {product} is here for {audience} like you.”

5. Email Snippet

“Hey {audience}, experience the {tone} benefits of {product} now!”

Each prompt is fed with the 3 user inputs and optionally sent to a generative model API for completion.

5. Input Validation and Output Filtering

Input Validation:

- All form fields (product, audience, tone) are required.
- Flask checks for missing inputs and returns an error message if any are blank.
- Optional JS validation can be added to block empty form submissions.

Output Filtering:

- Any profanity or harmful content from generated text can be sanitized using a Python library (e.g., better-profanity).
- Length restrictions are applied to prevent overly long or nonsensical outputs.
- Only sanitized text is shown on the page.

6. Sample Outputs

Input	Tone	Generated Output
EcoSmart Bottles	Eco-conscious adults	Friendly

| QuickChef Meals | Busy parents | Persuasive | “Busy parents, save time and eat better! QuickChef Meals brings healthy dinner to your doorstep in minutes.” |

7. Future Improvements

- Add login functionality and user history
- Let users select a format (email, tweet, post, tagline)
- Add more tones (funny, urgent, serious)
- Connect to OpenAI or Gemini API for real-time generation
- Add multi-language support for marketing in different regions

8. Deployment Note

The project is currently run locally via:

```
bashCopy codeflask run
```

To deploy it online:

- Use Render, Heroku, or Railway
- Set environment variables (API key, Flask secret)
- Move templates and static folders as-is