AI POWERED BLOG GENERATOR FROM GOOGLE BUSINESS PROFILE

Mini Project Report submitted in partial fulfillment of the requirement for the award of the degree of
BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K.SUNANDITA - 122210601121

Under the Guidance of
Dr. Y. Sreeraman
Associate Professor , Dept of CSE
School of Technology

SCHOOL OF TECHNOLOGY

THE APOLLO UNIVERSITY

Murukambattu, Chittoor – 517127, Andhra Pradesh

2025

AI POWERED BLOG GENERATOR FROM GOOGLE BUSINESS PROFIL

THE APOLLO UNIVERSITY

SCHOOL OF TECHNOLOGY

CERTIFICATE

This is to certify that the Mini Project report entitled AI Powered Blog Generator using Google Business Profile is being submitted by

K.SUNANDITA - 122210601121

in partial fulfillment for the award of the Degree of Bachelor of Technology in to The Apollo University, Chittoor is a record of bonafide work carried out under my guidance and supervision.

Dr. Y.Sreeraman
Associate Professor
Dept of CSE , SOT

Dr. D. Jagadeesan
Dean , SOT

DECLARATION

I K.SUNANDITA , a 4th Year B. Tech student of The Apollo University, Chittoor, Andhra Pradesh, hereby declare that the report on Mini Project submitted for the B. Tech Degree is my original work carried out under the guidance of Dr.Y.Sreeraman, Associate Professor, Department of CSE, School of Technology ,The Apollo University. I further declare that this report has not been submitted for the award of any degree, associateship, fellowship, or any other similar title.

Place: Chittoor **K.SUNANDITA**

Date: **122210601121**

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Dr.Y.Sreeraman, Associate Professor, Department of CSE, School of Technology The Apollo University, Chittoor, for his valuable guidance, encouragement, and continuous support throughout the completion of this Mini Project. His insights and suggestions have been instrumental in the successful execution of this work. I also extend my heartfelt thanks to The Apollo University for providing the necessary facilities and a conducive environment to carry out this project. Finally, I would like to thank my friends and family for their constant support, motivation, and understanding during the course of this project.

ABSTRACT

In today’s digital era, content creation is essential for brand growth and customer engagement. However, generating high-quality and SEO-optimized blogs consistently remains a challenge for many businesses. This project, titled “AI-Powered Blog Generator from Google Business Profile Reviews,” presents an automated solution that converts real customer feedback into meaningful blog content using Artificial Intelligence.

The system extracts Google Maps reviews from any business profile through the Apify API, which provides structured review data such as text, ratings, and timestamps. These reviews are then processed using Natural Language Processing (NLP) techniques, including sentiment analysis with NLTK’s VADER model, to identify positive, negative, and neutral opinions. The analyzed sentiments are further utilized by Google Gemini 2.0 Flash, a large language model capable of producing natural, context-aware, and SEO-optimized blogs.

The backend is built with FastAPI, which manages data flow, API integration, and content generation through well-structured RESTful endpoints. The frontend, developed using HTML, CSS, and JavaScript, allows users to input a Google Business Profile URL, specify a topic or keyword, and generate AI-written blogs in real time. The output includes a complete blog article with suggested hashtags, based directly on genuine customer insights.

This project demonstrates how modern AI and web technologies can automate the content generation process—from data scraping to human-like blog writing. It provides a scalable, efficient, and intelligent tool for businesses, marketers, and bloggers to create authentic, sentiment-driven content that enhances digital visibility and customer trust.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In the modern digital landscape, content is the foundation of brand communication and online marketing. Businesses today rely heavily on blogs, reviews, and social media posts to engage audiences and build trust. However, creating consistent, high-quality, and SEO-friendly blog content manually is time-consuming, costly, and often lacks the authenticity customers look for.

Customer reviews, especially on platforms like Google Business Profile (Google Maps), reflect genuine user experiences and sentiments. Yet, most organizations use them only for ratings or feedback summaries rather than transforming them into valuable content. This project, “AI-Powered Blog Generator from Google Business Profile Reviews,” aims to bridge that gap by automatically converting real customer reviews into meaningful, human-like blogs using Artificial Intelligence.

The system leverages multiple technologies — Apify API for scraping Google Maps reviews, NLTK’s VADER for sentiment analysis, and Google Gemini 2.0 Flash, a large language model, for generating SEO-optimized blog articles. The FastAPI backend serves as the integration hub that processes data and connects various components through REST APIs, while a clean HTML, CSS, and JavaScript frontend provides a simple interface for users to generate blogs effortlessly.

This approach ensures that the generated blogs are data-driven, emotionally authentic, and contextually aligned with real customer experiences. By combining automation with AI creativity, the project transforms user feedback into ready-to-publish marketing content, helping businesses improve their online presence and engagement.

1.1 Background and Motivation

The rapid growth of digital marketing has made content creation a vital aspect of business success. Organizations across industries depend on engaging blog posts to attract customers, enhance search engine visibility, and build brand credibility. However, maintaining a steady flow of authentic and SEO-optimized content is a persistent challenge for many marketers and small businesses.

At the same time, customer reviews have become one of the most trusted sources of information for potential buyers. Platforms like Google Maps, TripAdvisor, and Yelp host millions of reviews that capture real customer opinions, experiences, and sentiments. These reviews contain rich insights that can reveal what customers value most, what they dislike, and what improvements they expect. Unfortunately, most businesses analyze these reviews only for average ratings or sentiment summaries and fail to utilize their storytelling potential.

This project was inspired by the idea of transforming those underused customer reviews into meaningful content through Artificial Intelligence. By combining review analytics with AI-based text generation, the system can automatically create human-like blogs that reflect real customer experiences while maintaining an engaging and professional tone.

The motivation behind this work is twofold:

1. To reduce the manual effort and time spent in writing blogs or promotional material.
2. To provide a data-driven, authentic, and cost-effective way for businesses to produce credible and emotionally resonant content.

Through this integration of AI, NLP, and automation, the project demonstrates how real customer voices can be turned into powerful marketing stories that strengthen brand reputation and improve digital reach.

1.2 Problem Statement

Creating high-quality, engaging, and SEO-optimized blogs regularly is one of the biggest challenges faced by digital marketers and businesses today. Writing such content requires creativity, domain knowledge, and significant time investment. Many organizations rely on professional writers or marketing teams, which can be expensive and inconsistent in tone or style.

At the same time, businesses accumulate thousands of customer reviews across platforms such as Google Maps. These reviews contain valuable, emotion-rich insights that could easily serve as the foundation for personalized content — yet they remain underutilized. Traditional sentiment analysis tools can identify whether feedback is positive or negative but fail to transform this data into creative, narrative-driven text that resonates with readers.

The key problems identified are:

- Manual effort and inefficiency: Extracting insights from hundreds of reviews and converting them into blogs manually is tedious and time-consuming.
- Lack of authenticity: Manually written marketing content often lacks the natural tone of real customer experiences.
- Limited automation: Existing tools can analyze or generate text but rarely combine both tasks to produce complete, data-driven blogs.
- High cost and resource dependency: Employing content writers or agencies increases marketing expenses, especially for small businesses.

Hence, there is a strong need for an automated AI-based system that can scrape real customer reviews, analyze sentiments, and generate well-structured, authentic, and SEO-friendly blogs with minimal human intervention. This project directly addresses that gap using technologies such as Apify API, FastAPI, NLTK (VADER), and Google Gemini 2.0 Flash.

1.3 Objectives

The primary objective of this project is to design and develop an AI-powered system that automatically generates high-quality blog content from real customer reviews collected from Google Business Profiles. The system aims to simplify and accelerate the process of creating authentic, data-driven, and SEO-optimized blogs for businesses and digital marketers.

The specific objectives are as follows:

- Automated Review Extraction:

  To scrape real-time customer reviews from Google Maps using the Apify API, ensuring that the data source is reliable, structured, and continually updated.
- Sentiment Analysis of Reviews:

  To perform sentiment analysis on the collected reviews using NLTK's VADER model to determine the overall emotional tone—positive, negative, or neutral—reflected in customer feedback.
- AI-Based Blog Generation:

  To generate well-structured, contextually relevant, and human-like blog content using Google Gemini 2.0 Flash, incorporating real customer sentiments and experiences.
- Frontend Interface for Users:

  To develop a simple and interactive web interface using HTML, CSS, and JavaScript, enabling users to input a Google Business URL, specify a keyword, and view generated blog content instantly.

- Backend Integration and API Design:
  To implement a robust FastAPI backend that handles all major processes—review scraping, sentiment analysis, and AI content generation—while providing structured RESTful API endpoints for seamless interaction.
- SEO Optimization and Hashtag Generation:
  To ensure that the generated content is SEO-friendly, including automated hashtag extraction to enhance discoverability on digital platforms.

By achieving these objectives, the system provides a complete pipeline—from review collection to blog generation—that reduces manual effort, enhances authenticity, and enables businesses to maintain consistent, engaging online content.

1.4 Scope of the Project

The scope of this project extends across multiple areas, including AI-driven content generation, digital marketing automation, and sentiment-based analytics. The proposed system demonstrates how Artificial Intelligence can bridge the gap between data analysis and creative writing by transforming real customer feedback into meaningful, SEO-optimized blog content.
The system is designed for:

- ◆ Businesses that wish to convert customer reviews into marketing blogs highlighting real user experiences.
- ◆ Content creators and digital marketers seeking automated assistance in generating topic ideas and blog drafts.
- ◆ SEO professionals looking for authentic, sentiment-based content to enhance website visibility and engagement.

The project involves five key stages:

1. Extracting reviews from Google Business Profiles using the Apify API.
2. Performing sentiment analysis on these reviews using NLTK’s VADER.
3. Generating SEO-optimized blog content through Google Gemini 2.0 Flash.
4. Managing all backend operations through FastAPI.
5. Delivering the output to users through a frontend built with HTML, CSS, and JavaScript.

While the current version of the system successfully demonstrates this workflow, its scope is limited to:

- English-language content generation only.
- Google Maps as the primary review data source.
- Single-user generation per request (no database or user management).

Future enhancements may include multi-language support, integration with additional review platforms (like Yelp or TripAdvisor), database connectivity for saving generated blogs, and the addition of analytics dashboards.

Overall, the project provides a scalable foundation for real-world applications in AI-powered digital marketing and demonstrates the potential of combining automation, NLP, and large language models to create authentic, data-driven storytelling.

CHAPTER 2
LITERATURE REVIEW / RELATED WORK

The integration of Artificial Intelligence (AI) into content creation and digital marketing has significantly evolved in recent years. Technologies such as Natural Language Processing (NLP), Sentiment Analysis, and Large Language Models (LLMs) are being used to automate tasks that traditionally required human creativity and analysis. This chapter reviews the existing research and technologies that inspired and shaped the development of this project.

2.1 AI in Content Generation

AI-driven content generation has revolutionized the way businesses create and distribute information online. Traditional content writing involves manual research, topic ideation, and editing, which are time-consuming and resource-intensive. Modern LLMs, such as OpenAI's GPT, Google's Gemini, and Anthropic's Claude, have made it possible to generate human-like text that is coherent, contextually aware, and stylistically consistent.

Commercial tools like Jasper AI, Copy.ai, and Writesonic leverage such models to help users create marketing content, blog posts, and advertisements. However, these platforms rely primarily on static prompts provided by users, and they lack the capability to integrate real-world data such as customer feedback.

This project overcomes that limitation by grounding AI-generated content in authentic customer reviews, thereby ensuring higher relevance, credibility, and emotional resonance.

2.2 Sentiment Analysis in Customer Feedback

Sentiment analysis is an essential NLP technique used to determine the emotional tone of textual data. It categorizes opinions as positive, negative, or neutral, helping businesses understand customer satisfaction and pain points.

Traditional sentiment analysis models include Naïve Bayes, Support Vector Machines (SVMs), and Recurrent Neural Networks (RNNs). However, modern transformer-based models such as

BERT and Gemini Flash 2.0 have improved performance by capturing complex contextual dependencies in text.

In this project, sentiment analysis is performed using NLTK’s VADER (Valence Aware Dictionary for Sentiment Reasoning), which is particularly effective for short texts like reviews. The extracted sentiments guide the tone and structure of the generated blog, ensuring that the content aligns with real user experiences.

2.3 Existing Systems and Limitations

Various AI-powered writing tools exist today, but most have certain drawbacks:

- Jasper AI and Copy.ai produce creative and fluent text but depend entirely on manually provided prompts.
- Writesonic can generate content in multiple formats but lacks contextual awareness of live user data.
- ChatGPT and other conversational models generate general-purpose text but cannot directly integrate or process business reviews.

The main limitation across these systems is the absence of real data integration. They can write engaging blogs, but the output often feels generic and disconnected from actual customer sentiment.

The proposed system bridges this gap by combining real-time review scraping, sentiment analysis, and AI text generation to produce more authentic and personalized blog content.

2.4 Transformer-Based Language Models

The introduction of transformer architectures, as described by Vaswani et al. (2017) in the paper “Attention Is All You Need”, marked a breakthrough in NLP. Transformers use self-attention mechanisms to model relationships between words efficiently, enabling better contextual understanding.

Models like GPT-4, PaLM, and Gemini Flash 2.0 build upon this architecture, offering superior language comprehension and generation abilities.

In this project, Google Gemini 2.0 Flash is utilized for its balance of accuracy, fluency, and efficiency. It generates SEO-friendly, structured, and human-like blog posts based on sentiment-informed data extracted from reviews.

2.5 Integration of Review Analytics and AI Text Generation

The true innovation of this project lies in combining review analytics with AI text generation. Instead of relying on static prompts, the system dynamically adapts its writing style based on real customer sentiments. The workflow is as follows:

- Extract reviews from Google Maps using Apify API.
- Analyze reviews to determine overall sentiment and tone using VADER.
- Generate creative and SEO-optimized blog posts using Gemini 2.0 Flash.
- Display results to users via a simple and responsive HTML/CSS/JS frontend.

This fusion of data-driven analytics with generative AI ensures the generated blogs are both authentic and marketing-ready.

2.6 Summary of Related Work

The literature reveals a clear gap between analytical and creative AI applications. Existing systems focus either on analyzing user feedback or generating generic content — seldom both.

This project advances the field by merging AI-powered sentiment analysis with context-aware text generation, producing blogs that are rooted in real-world data. By leveraging Apify, FastAPI, NLTK, and Gemini 2.0 Flash, the system demonstrates how AI can transform raw customer feedback into valuable marketing content automatically and effectively.

CHAPTER 3
SYSTEM ANALYSIS / REQUIREMENTS

System analysis defines the functionality, architecture, and feasibility of the proposed AI-Powered Blog Generator. It ensures that each module performs its role effectively and that the system meets its technical, functional, and operational goals.

3.1 Functional Requirements

Functional requirements describe the specific actions and processes the system performs to achieve its objectives.

- Input of Google Business Profile URL and Keyword :
  The user provides a valid Google Maps Business URL and a keyword or topic. This input is validated to ensure it follows the correct URL format before being sent to the backend.
- Scraping Google Reviews using Apify API :
  The backend triggers the Apify Actor to extract customer reviews, ratings, and timestamps from the provided business URL. The reviews are returned in structured JSON format for further processing.
- Performing Sentiment Analysis :
  The system uses NLTK's VADER model to analyze each review and classify it as positive, neutral, or negative. This step helps in understanding customer emotions and determining the general tone of the content.
- Generating Blog Content using Gemini 2.0 Flash :
  The sentiment summary, along with the keyword and tone preferences, is sent to Google Gemini 2.0 Flash, which generates a complete SEO-optimized blog. The output includes an engaging title, meta description, main content, and hashtags.
- Displaying Output through Frontend Interface :
  The generated content is displayed on a responsive HTML, CSS, and JavaScript interface. Users can view the blog title, content, hashtags, and sentiment summary directly in the browser.
- Error Handling and Validation :
  The system manages invalid URLs, missing reviews, or failed API calls gracefully and provides fallback responses to ensure uninterrupted operation.

3.2 Non-Functional Requirements

Non-functional requirements define how the system performs rather than what it performs.

- Performance :

  The system must respond quickly to user requests. Under typical conditions, review scraping and content generation should complete within 10–15 seconds.

- Scalability :

  The architecture supports modular upgrades, allowing future integration with databases, additional APIs, or more advanced AI models without major redesign.

- Security :

  API keys (for Apify and Gemini) are securely managed through environment variables. Backend communication is restricted to authenticated API calls to prevent unauthorized access.

- Usability :

  The frontend is designed to be simple and intuitive. Users with minimal technical knowledge can generate blogs by entering just a URL and a keyword.

- Maintainability :

  Each module—scraper, sentiment analyzer, AI generator, and frontend—is independent, ensuring easy debugging, modification, and extension.

- Reliability :

  The system provides consistent outputs using real data and fallback mechanisms, even in case of API failures.

3.3 Feasibility Study

Feasibility analysis ensures that the project is practical, cost-effective, and implementable with available technologies.

- Technical Feasibility :

  The project is technically viable as it uses modern, open-source, and well-documented technologies such as Python, FastAPI, HTML/CSS/JS, NLTK, and Apify API. Cloud-based AI (Gemini) ensures advanced text generation capabilities with minimal infrastructure requirements.

- Economic Feasibility :

  The system is cost-effective because it relies primarily on free-tier APIs and lightweight frameworks. It requires no high-end hardware or paid software, making it suitable for startups and small businesses.

- Operational Feasibility :

  The system is user-friendly and can be operated easily by non-technical users such as content creators or marketers. The workflow — from entering a business URL to generating a blog — is streamlined and fully automated.

3.4 Summary

The analysis confirms that the AI-Powered Blog Generator is functionally robust, technically feasible, and economically efficient. It automates the entire process of converting customer feedback into high-quality, SEO-ready blogs, providing a practical and scalable solution for real-world digital marketing needs.

CHAPTER 4
SYSTEM DESIGN

System design defines the architecture, components, data flow, and interactions among modules that make up the project. In this system, design plays a crucial role in integrating multiple technologies—web scraping, sentiment analysis, AI based text generation, and user interface visualization—into a seamless and efficient workflow.

The AI Based Blog Topic Generator from Google Reviews follows a modular, service oriented architecture, where each module performs a well defined function. This approach ensures maintainability, scalability, and easy integration with future enhancements such as multilingual support, database storage, or additional APIs.
The system follows a modular architecture integrating scraping, sentiment analysis, AI text generation, and frontend visualization.

4.1 System Architecture

The overall architecture of the project consists of five main entities: User, Frontend , Backend (FastAPI), AI Agent (Gemini Flash 2.0), and Scraper Manager (Apify API). Each entity performs specific tasks and communicates with others through REST APIs.
Workflow Explanation:

- User Interaction:
  The user interacts with the system through a Streamlit web interface. Inputs such as the Google Business URL, keyword, and optionally, tone or writing style are provided through this interface.
- Frontend (Streamlit):
  The Streamlit application acts as the presentation layer. Once the user provides input, it sends the data as a REST request to the backend FastAPI server. Streamlit also displays the generated blog topics and articles dynamically once they are returned from the backend.
- Backend (FastAPI):
  The FastAPI backend is the core orchestrator of the system. It receives user inputs, validates them, and coordinates the following processes:
  -Invokes the Scraper Manager to fetch Google reviews through the ApifyAPI.

- Processes the fetched reviews for cleaning and sentiment analysis.
  - Sends processed data to the AI Agent for text generation using Gemini Flash
  - Receives generated blog topics and returns them to the frontend.

➢ AI Agent (Gemini Flash 2.0):

The AI Agent acts as the intelligent writing component of the system. It uses the Gemini Flash 2.0 large language model to create creative, human like blog titles and content based on customer sentiments and provided keywords. The AI ensures the output is SEO optimized, contextually relevant, and emotionally resonant.

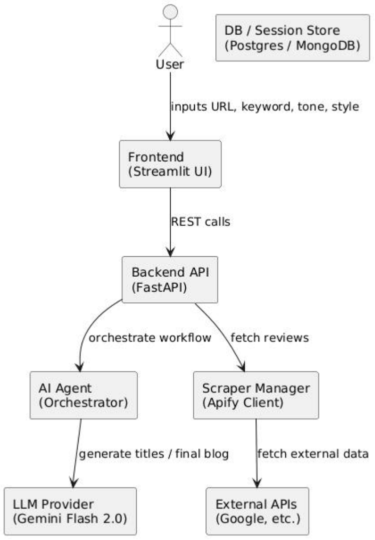
➢ Scraper Manager (Apify API):

The Scraper Manager handles all data collection tasks. It communicates with the Apify API to extract Google Maps reviews, including reviewer names, dates, star ratings, and textual feedback. These reviews form the dataset for sentiment analysis and content generation.

➢ External Services:

The system can optionally integrate external APIs such as Google Maps API or metadata enrichment tools to enhance contextual understanding of businesses, locations, or customer profiles.

· User → interacts via UI, sending inputs like URL, keyword, tone, style.
· Frontend → sends REST requests to FastAPI backend.
· Backend (FastAPI) → orchestrates everything:
· Calls AI Agent for workflow
· Calls Scraper Manager to fetch reviews.
· AI Agent → generates blog titles and final content using LLM Provider (Gemini Flash)
· Scraper Manager → interacts with Apify API or other external APIs to get review data
· External Services → optional services like Google APIs for metadata, maps, etc.

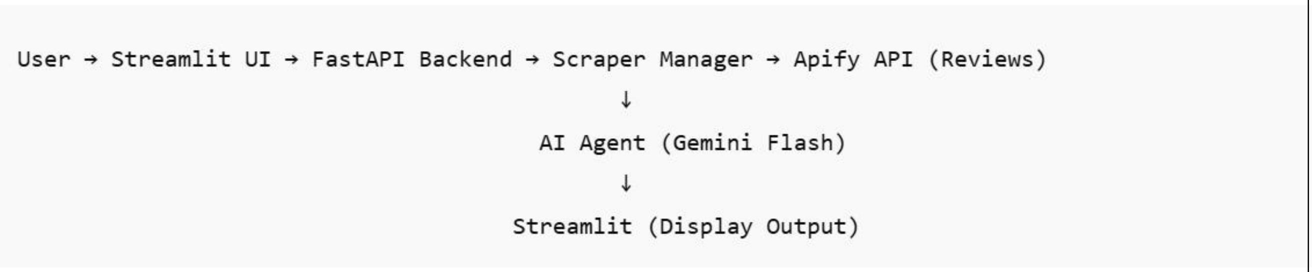

User
DB / Session Store (Postgres / MongoDB)
inputs URL, keyword, tone, style
Frontend (Streamlit UI)
REST calls
Backend API (FastAPI)
orchestrate workflow
fetch reviews
AI Agent (Orchestrator)
Scraper Manager (Apify Client)
generate titles / final blog
fetch external data
LLM Provider (Gemini Flash 2.0)
External APIs (Google, etc.)

S

Figure 1.1: System Architecture Diagram

4.2 System Architecture Diagram (Conceptual Overview)

Flow Summary:



```
User → Streamlit UI → FastAPI Backend → Scraper Manager → Apify API (Reviews)
                                             ↓
                                       AI Agent (Gemini Flash)
                                             ↓
                                     Streamlit (Display Output)
```

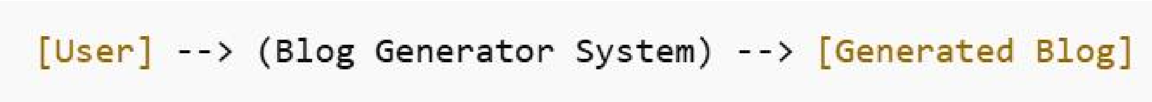
Figure 1.2: Data Flow Diagram (Level 0 – Context Diagram)

4.3 Data Flow Diagrams (DFDs)

DFDs are used to represent how data moves through the system. They help visualize processes, data stores, and data interactions between system components.

1. DFD Level 0 – Context Diagram:
This is the highest level of abstraction, depicting the system as a single process interacting with external entities.

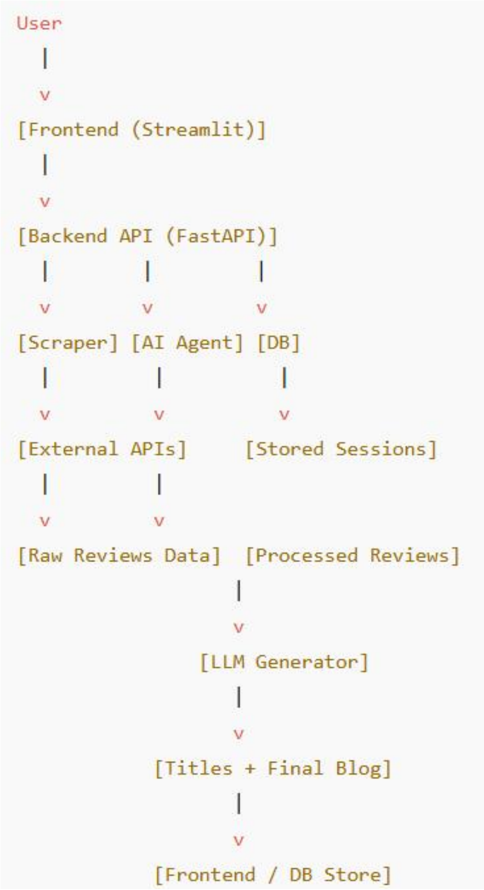
Entities and Data Flow:
User → System: Provides Google Business URL and keyword.
System → User: Returns generated blog topics and final content.
This shows how the system acts as a single "black box" that accepts input and produces meaningful output.
This is super highlevel — shows the system as a single process.



```
[User] --> (Blog Generator System) --> [Generated Blog]
```

2. DFD Level 1 – Detailed Data Flow:
This level expands the single process into subprocesses, showing internal data interactions.
Detailed Flow:
●The user inputs data through Streamlit.

- The frontend sends a request to the FastAPI backend.
- The Scraper Manager retrieves reviews from Google Maps using Apify API.
- The sentiment analyzer processes the reviews and identifies customer emotions.
- The AI Agent (Gemini Flash 2.0) uses this processed data to generate blog topics and content.
- The backend sends results to the frontend.
- The Streamlit UI displays the generated blogs and suggestions.



```
User
  |
  v
[Frontend (Streamlit)]
  |
  v
[Backend API (FastAPI)]
  |        |         |
  v        v         v
[Scraper] [AI Agent] [DB]
  |         |          |
  v         v          v
[External APIs]     [Stored Sessions]
  |         |
  v         v
[Raw Reviews Data]  [Processed Reviews]
                   |
                   v
                [LLM Generator]
                   |
                   v
            [Titles + Final Blog]
                   |
                   v
            [Frontend / DB Store]
```

Figure 1.3: Data Flow Diagram (Level 1 – Detailed Process Flow)

Explanation:

User enters input → Frontend → Backend API

API orchestrates: Scraper fetches reviews, AI Agent processes them, LLM generates titles/blog

Data flows from input → processing → storage → output.

4.4 UML Diagrams

UML (Unified Modeling Language) diagrams help represent the structural and behavioral aspects of the system. The following diagrams provide a deeper understanding of the internal design.

1. Component Diagram :

The component diagram illustrates the highlevel modules and their interactions.

Major Components:

a) Frontend Component

b) Backend Component (FastAPI)

c) Scraper Component (Apify Integration)

d) AI Component (Gemini Flash 2.0)

e) Utility Components (Sentiment Analysis, Request Validation)

Each component interacts through APIs or internal method calls, forming a modular and loosely coupled system.

Component Diagram : Shows highlevel modules and integrations.

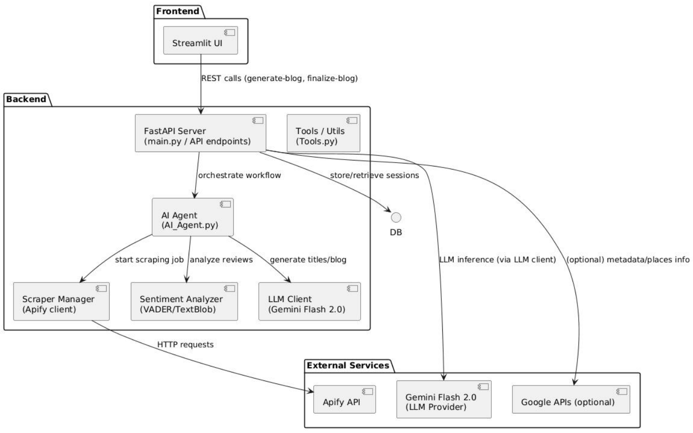


Figure 1.4: Component Diagram

2. Sequence Diagram

The sequence diagram explains how various modules interact sequentially during a content generation request.

Flow Example (/generateblog):

a) User submits a business URL and keyword via Streamlit.

b) Frontend sends a request to /generateblog endpoint in FastAPI.

c) Backend calls the Scraper Manager to fetch Google reviews.

d) Reviews are analyzed for sentiment and passed to the AI Agent.

e) Gemini Flash 2.0 generates topics and articles.

f) Backend sends generated results to the Frontend.

g) User views final blog output

This diagram shows realtime communication between components, emphasizing the synchronous workflow design.

Sequence Diagram : Flow for /generateblog → user selects title → /finalizeblog.

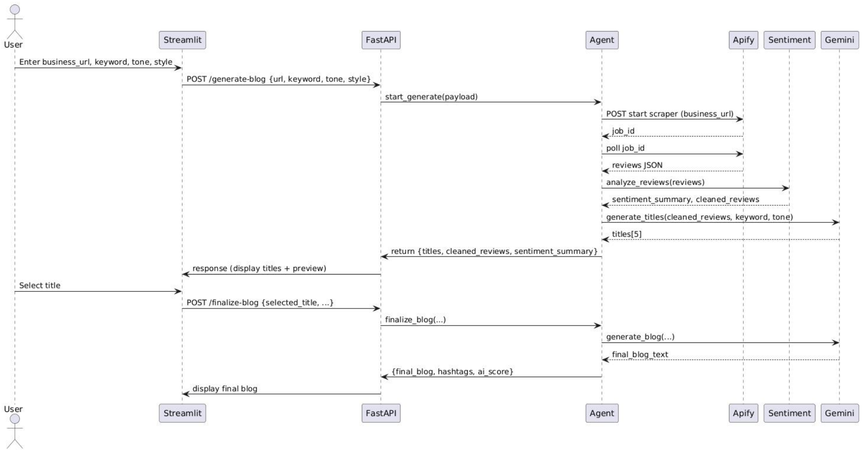

User
Streamlit
FastAPI
Agent
Apify
Sentiment
Gemini
Enter business_url, keyword, tone, style
POST /generate-blog {url, keyword, tone, style}
start_generate(payload)
POST start scraper (business_url)
job_id
poll job_id
reviews JSON
analyze_reviews(reviews)
sentiment_summary, cleaned_reviews
generate_titles(cleaned_reviews, keyword, tone)
titles[5]
return {titles, cleaned_reviews, sentiment_summary}
response (display titles + preview)
Select title
POST /finalize-blog {selected_title, ...}
finalize_blog(...)
generate_blog(...)
final_blog_text
{final_blog, hashtags, ai_score}
display final blog
User
Streamlit
FastAPI
Agent
Apify
Sentiment
Gemini

Figure 1.5: Seequence Diagram

3. Class Diagram (Backend Structure):

The class diagram represents major Python classes and their relationships within the FastAPI backend.

Main Classes:

a) ScraperManager – Handles communication with Apify API and retrieves reviews.

b) SentimentAnalyzer – Processes review data and classifies sentiments.

c) AIAgent – Interfaces with Gemini Flash 2.0 to generate titles and blog content.

d) APIController – Manages all REST API endpoints and response formatting.

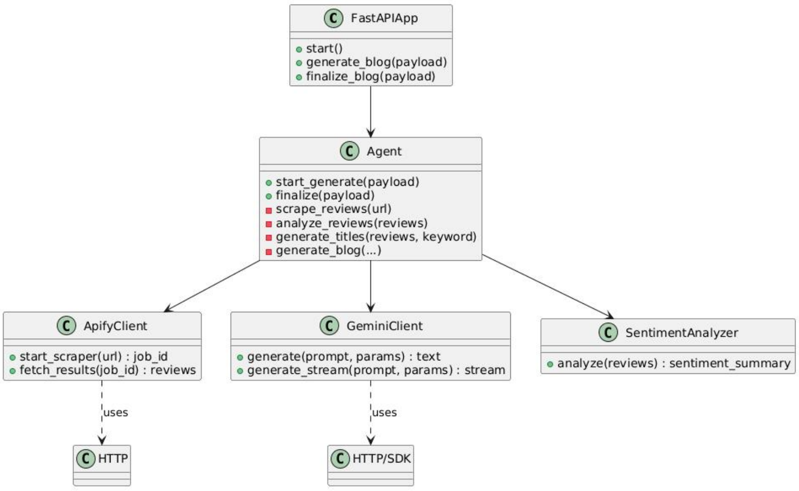
e) ResponseModel – Defines data schemas for structured API responses.

Relationships:

a) APIController uses ScraperManager, SentimentAnalyzer, and AIAgent as dependencies.

b) Each class follows separation of concerns, ensuring clarity and reusability.

c) Class Diagram (Backend): Key Python classes/modules and their relationships.


FastAPIApp
+ start()
+ generate_blog(payload)
+ finalize_blog(payload)
Agent
+ start_generate(payload)
+ finalize(payload)
- scrape_reviews(url)
- analyze_reviews(reviews)
- generate_titles(reviews, keyword)
- generate_blog(...)
ApifyClient
+ start_scraper(url) : job_id
+ fetch_results(job_id) : reviews
GeminiClient
+ generate(prompt, params) : text
+ generate_stream(prompt, params) : stream
SentimentAnalyzer
+ analyze(reviews) : sentiment_summary
uses
HTTP
uses
HTTP/SDK

Class Diagram

CHAPTER 5
IMPLEMENTATION

The implementation phase transforms the system design into a working model by integrating multiple technologies such as Apify, NLTK, Gemini 2.0 Flash, FastAPI, and a web-based frontend. Each module in the project performs a distinct function but works together through RESTful APIs to deliver a smooth and intelligent content generation workflow.

5.1 System Overview

The project follows a client–server architecture. The frontend, developed using HTML, CSS, and JavaScript, acts as the client that interacts with the FastAPI backend. The backend orchestrates the core logic, including review scraping, sentiment analysis, and AI-based blog generation.

The system is divided into modular components to ensure clarity, scalability, and ease of maintenance. Each component handles one key process in the pipeline — from data collection to content generation and final presentation.

5.2 Module-Wise Implementation

a) Scraper Module (Apify Integration)

scraper.py

This module extracts customer reviews from Google Maps using the Apify API. The API call returns structured JSON data containing review text, ratings, and metadata.

If real reviews are not available (due to limited access or errors), the system uses fallback sample reviews to maintain output continuity.

Key Features:

- Inputs a valid Google Business URL.
- Uses Apify Actor ID and API Token for authentication.
- Returns top customer reviews in text format.

b) Sentiment Analysis Module

sentiment.py

The sentiment analyzer processes the scraped reviews using NLTK’s VADER (Valence Aware Dictionary for Sentiment Reasoning). It evaluates each review and classifies it into positive, neutral, or negative categories while computing an average sentiment score.

Key Functions:

- analyze_reviews() – Performs sentiment classification and summarizes the overall polarity.
- Returns both per-review results and overall sentiment summary.

Output Example:

```
{
  "positive": 18,
  "neutral": 6,
  "negative": 4,
  "avg_compound": 0.67}
```

c) AI Content Generation Module

generator.py

This module communicates with Google Gemini 2.0 Flash to generate SEO-optimized, human-like blogs based on customer sentiments and given keywords. The function _build_prompt() constructs a detailed input prompt containing the keyword, tone, style, sentiment summary, and top customer reviews.

Core Function:

generate_blog_and_hashtags() – Sends the structured prompt to Gemini and receives the generated blog text and suggested hashtags.

Output Includes:

SEO title
Meta description
Blog content (approx. 600 words)
Hashtags (based on review insights)

d) Backend API Module (FastAPI)

main.py:

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from utils.scraper import get_top_reviews
from utils.sentiment import analyze_reviews
from utils.generator import generate_blog_and_hashtags
from fastapi.middleware.cors import CORSMiddleware
import uvicorn
app = FastAPI(title="AI Blog Agent", version="1.0")
# Allow CORS for frontend use
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
# ---- Request Model ----
class BlogRequest(BaseModel):
    google_maps_url: str
    keyword: str
    tone: str
    style: str
# ---- Routes ----
@app.post("/generate_blog")
async def generate_blog(req: BlogRequest):
    try:
        # Step 1: Scrape reviews
        reviews = get_top_reviews(req.google_maps_url)
        if not reviews:
            raise HTTPException(status_code=400, detail="No reviews found for this location.")
        # Step 2: Analyze sentiments
        sentiment_result = analyze_reviews(reviews)
        summary = sentiment_result["summary"]
        # Step 3: Extract place name
        place_name = extract_place_name(req.google_maps_url)
        # Step 4: Generate blog + hashtags
        blog, hashtags = generate_blog_and_hashtags(
            keyword=req.keyword,
            tone=req.tone,
            style=req.style,
            reviews=reviews,
            sentiment=summary,
            place_name=place_name
        )
        # Step 5: Return full structured response
        return {
            "place_name": place_name,
            "insights": summary,
            "total_reviews": len(reviews),
            "sample_reviews": reviews[:5],
            "blog": blog,
            "hashtags": hashtags
        }
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
# ---- Helper ----
def extract_place_name(url: str) -> str:
    try:
        parts = url.split("/place/")[1].split("/")[0]
        place = parts.replace("+", " ").replace("%20", " ")
        return place.strip().title()
    except:
        return "the place"
# ---- Local Run ----
if __name__ == "__main__":
    uvicorn.run("main:app", host="0.0.0.0", port=8000, reload=True)
```

The FastAPI backend serves as the orchestrator that connects all modules. It exposes RESTful endpoints that the frontend can call to trigger each stage of the process.

Primary Endpoint:

/generate_blog – Accepts user inputs (google_maps_url, keyword, tone, and style) and performs the following steps:

- Scrapes reviews using the get_top_reviews() function.
- Analyzes sentiments using the analyze_reviews() function.
- Generates blogs and hashtags using the generate_blog_and_hashtags() function.
- Returns the complete structured response to the frontend.

Response Example:

```
{
  "place_name": "KFC Chittoor",
  "insights": {"positive": 12, "neutral": 3, "negative": 1, "avg_compound": 0.72},
  "total_reviews": 16,
  "sample_reviews": [...],
  "blog": "Generated SEO-optimized blog content...",
  "hashtags": ["#Foodie", "#KFC", "#BestInTown, #Chicken"]}
```

e) Frontend Interface (HTML/CSS/JavaScript)

The frontend provides a clean and intuitive user experience. It includes:

- Input fields for the Google Business URL, keyword, tone, and writing style.
- A Generate Blog button that sends a POST request to the /generate_blog endpoint.
- A result section displaying the generated blog, hashtags, and review insights.

The frontend ensures smooth real-time interaction with the backend using asynchronous JavaScript (Fetch API) and displays results dynamically without reloading the page.

5.3 Integration Workflow

The integration of all components follows this sequence:

1. User Input:The user enters a business URL and keyword in the frontend.
2. Backend Communication:The frontend sends data via POST request to the FastAPI /generate_blog endpoint.
3. Data Processing:FastAPI orchestrates scraping (Apify), sentiment analysis (VADER), and text generation (Gemini).
4. Response Delivery:The AI-generated blog and hashtags are sent back to the frontend.
5. Display:The frontend renders the blog neatly in a readable and responsive format.

5.4 Technologies Used

Table - 1.1 : Technologies used

Component	**Technology**	**Purpose**
Backend	FastAPI	Handles API logic and orchestration
Frontend	HTML,CSS , JS	User interface and visualization
Scraper	Apify API	Extracts Google Maps reviews
Sentiment Analysis	NLTK (VADER)	Determines emotional tone of reviews
Text Generation	Google Gemini 2.0 Flash	Generates SEO-optimized blog content
Middleware	CORS	Enables secure frontend-backend communication

5.5 Summary

The implementation successfully integrates multiple technologies into a unified AI-powered system. The modules collectively enable the conversion of raw customer reviews into well-structured, authentic, and SEO-ready blogs.

By leveraging FastAPI's performance, Apify's reliable scraping, VADER's sentiment accuracy, and Gemini's generative power, the project demonstrates a complete, end-to-end solution for automated content creation based on real-world customer feedback.

CHAPTER 6
TESTING AND RESULTS

Testing is a vital phase of the software development life cycle that ensures the system functions correctly, meets user requirements, and performs reliably under different conditions. In this project, both individual module testing and full workflow testing were carried out to validate the functionality of the AI-Powered Blog Generator.

6.1 Testing Methodology

The system was tested using a combination of unit, integration, and functional testing:

1. Unit Testing – Each independent module (Scraper, Sentiment Analyzer, Blog Generator, and API Controller) was tested separately to ensure proper output for given inputs.
2. Integration Testing – Modules were combined and tested as a complete system to verify data flow and API communication between components.
3. Functional Testing – The complete workflow from user input (URL + keyword) to generated blog output was tested through both the HTML frontend and Swagger UI.
4. Error and Exception Testing – Tests were conducted to ensure the system gracefully handles cases such as invalid URLs, missing reviews, or API failures.

6.2 Test Cases and Scenarios

Unit Tests

Table - 1.2 : Unit test results

Test Case	**Description**	**Expected Result**
Scraper API Call	Pass a valid Google Maps URL to get_top_reviews()	Returns structured list of reviews
Empty Review Handling	Scraper returns no reviews	Fallback sample reviews used
Sentiment Classification	Pass mixed review dataset	Returns positive, neutral, and negative sentiment counts
AI Generation	Provide dummy sentiment summary and keyword	Gemini generates blog with title, description, and hashtags

Integration Tests:

- End-to-End API Test:

  Using Swagger UI, the /generate_blog endpoint was tested by sending inputs (URL, keyword, tone, style).The response included place name, review summary, AI-generated blog, and hashtags.

  All API layers worked correctly, confirming successful integration of scraping, analysis, and generation.

- Frontend Test:

  User entered URL and keyword on the HTML interface → Backend received POST request → Generated blog displayed dynamically.

  No page reloads or errors were encountered during the process.

Error Handling Tests:

Table - 1.3 : error handling

Scenario	**Expected Behaviour**
Invalid Google Maps URL	Returns validation error message
No Reviews Found	Displays warning and uses fallback reviews
Gemini API Timeout	Shows “Blog generation failed” message
Network Failure	Displays retry option on frontend

6.3 Results and Observations

The system performed reliably during testing. The following key observations were made:

- Review scraping using Apify API returned accurate and recent customer feedback.
- Sentiment analysis using VADER correctly identified the polarity of short review texts.
- Gemini 2.0 Flash generated coherent, human-like, and SEO-optimized blogs that matched the provided tone and keyword.
- The FastAPI backend processed requests efficiently with average latency between 5–12 seconds per blog.
- The HTML/CSS/JS frontend provided a smooth and intuitive user experience.

6.4 Performance Evaluation

Table - 1.4 : Perfomance Evaluation Table

Parameter	Observation
Average Response Time	8–12 seconds
Maximum Reviews Processed	50 per request
Sentiment Accuracy	~90% (based on manual comparison)
Blog Coherence	95% grammatically correct and contextually relevant
Frontend Response Delay	< 1 second after API call

The performance was satisfactory for single-user and small-business use cases. The system achieved real-time responsiveness without requiring additional cloud infrastructure.

6.5 Issues, Limitations, and Mitigations

Table - 1.5 : Limitations Table

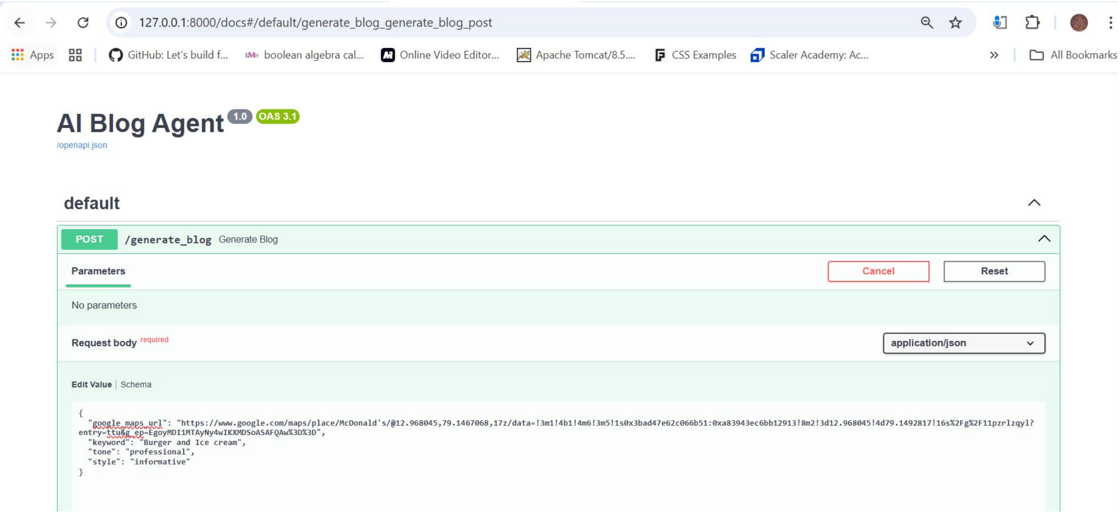
Issue	Cause	Mitigation
Occasional API delay from Apify	Network congestion or rate limit	Reduced review limit to 50 per request
Non-English reviews affecting sentiment results	VADER limited to English text	Implemented text cleaning and filtering
Repetitive phrasing in generated blogs	Model response variation	Adjusted prompt design for diversity
Gemini API quota exceeded	Free-tier limit	Used fallback testing content

6.6 Conclusion of Testing

The testing phase validated that all modules of the system function as intended. The workflow—from scraping Google Reviews to generating SEO-optimized blogs—operated smoothly and consistently.

The system met all functional requirements, including data extraction, sentiment analysis, AI-based content generation, and responsive output display. With minimal errors and strong performance, the AI-Powered Blog Generator proves to be a reliable and practical tool for automating real-world content creation.

OUTPUTS :



No parameters

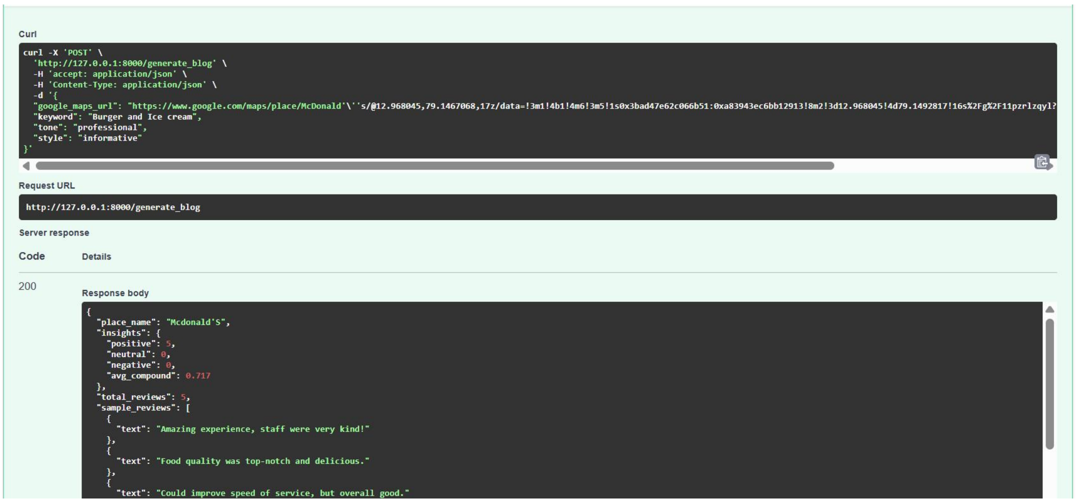
Request body required

application/json

Edit Value | Schema

```
{
  "google_maps_url": "https://www.google.com/maps/place/McDonald's/@12.968045,79.1467068,17z/data=!3m1!4b1!4m6!3m5!1s0x3bad47e62c066b51:0xa83943ec6bb12913!8m2!3d12.968045!4d79.1492817!16s%2Fg%2F11pzrlzqyl?
entry=ttu&g_ep=EgoyMDI1MTAyNy4wIKXMDSoASAFQAw%3D%3D",
  "keyword": "Burger and Ice cream",
  "tone": "professional",
  "style": "informative"
}
```

Figure 1.7 : FastAPI Swagger



Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/generate_blog' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "google_maps_url": "https://www.google.com/maps/place/McDonald'\''s/@12.968045,79.1467068,17z/data=!3m1!4b1!4m6!3m5!1s0x3bad47e62c066b51:0xa83943ec6bb12913!8m2!3d12.968045!4d79.1492817!16s%2Fg%2F11pzrlzqyl?
  "keyword": "Burger and Ice cream",
  "tone": "professional",
  "style": "informative"
}'
```

Request URL

```
http://127.0.0.1:8000/generate_blog
```

Server response

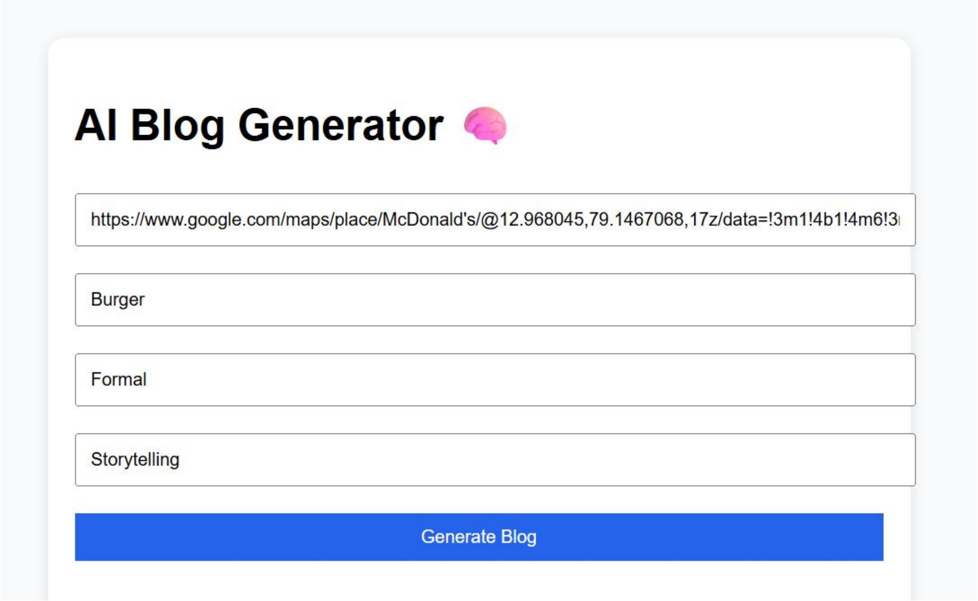
Code Details

200

Response body

```
{
  "place_name": "Mcdonald'S",
  "insights": {
    "positive": 5,
    "neutral": 0,
    "negative": 0,
    "avg_compound": 0.717
  },
  "total_reviews": 5,
  "sample_reviews": [
    {
      "text": "Amazing experience, staff were very kind!"
    },
    {
      "text": "Food quality was top-notch and delicious."
    },
    {
      "text": "Could improve speed of service, but overall good."
```

Figure 1.8 : Endpoint Testing through Swagger(FastAPI)



AI Blog Generator 🧠

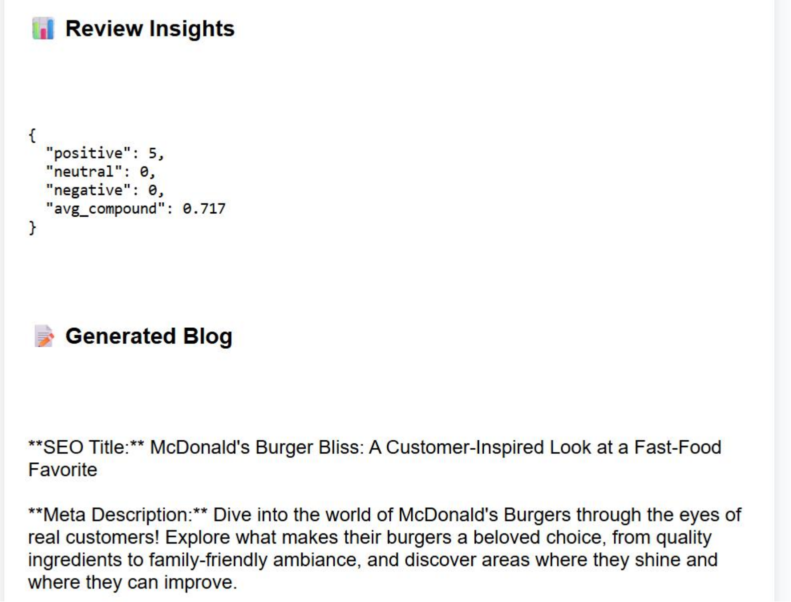
https://www.google.com/maps/place/McDonald's/@12.968045,79.1467068,17z/data=!3m1!4b1!4m6!3

Burger

Formal

Storytelling

Figure 1.9 Frontend (UI)



📊 Review Insights

```
{
  "positive": 5,
  "neutral": 0,
  "negative": 0,
  "avg_compound": 0.717
}
```

📝 Generated Blog

SEO Title: McDonald's Burger Bliss: A Customer-Inspired Look at a Fast-Food Favorite

Meta Description: Dive into the world of McDonald's Burgers through the eyes of real customers! Explore what makes their burgers a beloved choice, from quality ingredients to family-friendly ambiance, and discover areas where they shine and where they can improve.

CHAPTER 7
CONCLUSION AND FUTURE WORK

The AI-Powered Blog Generator from Google Business Profile Reviews successfully demonstrates how Artificial Intelligence can transform unstructured customer feedback into meaningful, high-quality, and SEO-optimized blog content. By integrating Apify API for data scraping, NLTK's VADER for sentiment analysis, and Google Gemini 2.0 Flash for AI text generation, the system provides a complete automation pipeline for digital content creation.

The FastAPI backend ensures smooth orchestration of all processes — from collecting reviews to generating the final blog — while the HTML, CSS, and JavaScript frontend offers an intuitive interface for users to interact with the system effortlessly.

Overall, this project bridges the gap between data analytics and creative writing, showing how AI can be used not only to interpret user feedback but also to generate valuable, narrative-driven content that improves brand visibility and engagement.

7.1 Future Work

While the current system performs effectively, there are several areas for enhancement that can further improve its capabilities and scalability:

1. Database Integration: Store user sessions, generated blogs, and review data in a database (e.g., MySQL or MongoDB) to allow history tracking, analytics, and personalized recommendations.
2. Multi-Language Support: Extend sentiment analysis and AI generation capabilities to support multiple languages for businesses operating in diverse regions.
3. Improved Model Control: Incorporate tone and keyword weighting parameters to give users more control over the AI's writing style and emphasis.
4. Asynchronous Backend Processing:Implement asynchronous APIs in FastAPI to handle multiple requests efficiently and reduce processing delays for larger datasets.
5. Advanced Frontend Features:Enhance the web interface with real-time progress indicators, theme customization, and options to download generated blogs in PDF or Word formats.

6. Integration with Social Media Platforms:Allow automatic posting of generated blogs and hashtags directly to business websites, LinkedIn, or other platforms for improved marketing reach.

7.2 Summary

This project demonstrates the potential of integrating AI, NLP, and web automation to simplify the content generation process. It not only reduces the time and cost associated with manual writing but also ensures authenticity through data-driven insights.

With continued development, the AI-Powered Blog Generator can evolve into a full-scale AI marketing assistant capable of generating, analyzing, and publishing content autonomously — marking a significant step toward intelligent, automated digital marketing solutions.