

# AI-Enhanced Websites with LLM APIs

---

Leveraging **LLM APIs** is an effective way to transform standard websites into interactive, AI-driven platforms. In this document, we demonstrate how LLM APIs can be seamlessly utilised to create powerful features like a **chatbot** and a **sentiment analysis system**, adding intelligence and engagement to the website with ease.

Through the LLM-powered chatbot, users enjoy real-time, responsive interactions, while the sentiment analysis system automatically processes feedback to uncover customer insights. This project showcases the versatility and efficiency of LLM APIs in building advanced, user-centered web features that enhance the overall digital experience.

## **Liverpool Larder - A Showcase of LLM API-Enhanced Food Ordering Platform**

**Liverpool Larder** is a sample food ordering website designed on **Bubble.io** to illustrate how **Large Language Model (LLM) APIs** can significantly enhance website interactions and functionality. This project utilizes the **Gemini API** to power **ChefBot**, an AI chatbot that assists users with personalized recommendations and answers to common queries, creating a seamless and interactive experience. Additionally, a **Sentiment Analysis System** analyzes user reviews to extract valuable feedback, helping the platform continuously improve its service.

While Gemini API is used here, similar results can be achieved with other LLM APIs, such as **ChatGPT**. These APIs offer versatile options for integrating intelligent features like chatbots, sentiment analysis, and personalized recommendations, making LLM technology an invaluable tool for enriching any website.

## Website Creation Using Bubble.io

The Liverpool Larder platform was developed entirely on Bubble.io, which allowed for rapid and adaptable web development without requiring extensive coding knowledge. The choice of Bubble.io as the development platform was made to enable quick prototyping, easy modifications, and efficient management of both front-end and back-end components.

*Link : <https://liverpool-larder.bubbleapps.io/version-test>*

## Sentiment Analysis

The primary goal of implementing sentiment analysis was to enable the team at Liverpool Larder to gain a deeper understanding of customer experiences. Customers often left feedback in the form of comments after placing orders, providing insights into what they liked and areas that needed improvement. However, manually analysing this feedback would be time-consuming and prone to inconsistencies. By incorporating sentiment analysis, administrators could assess the tone of each review: whether it was positive, negative, or neutral and then use this information to make data-driven decisions to enhance service quality.

## Implementation Approach Using Gemini API

The sentiment analysis system was implemented using the Gemini API, which was integrated into the Bubble platform to streamline feedback analysis. This integration allowed for the automatic categorisation of customer comments, providing actionable insights directly to administrators.

**Data Collection:** Feedback data was gathered whenever a customer left a comment on the platform. These comments were sent to the Gemini API via API workflows created in Bubble. This automated process ensured that feedback was collected and processed consistently without manual intervention.

**Sentiment Analysis via Gemini API:** The feedback data was passed through the Gemini API, which utilised Natural Language Processing (NLP) techniques to analyse the text and determine its sentiment. Gemini assessed key words, context, and phrases to classify each review into one of three categories: positive, negative, or neutral. The use of Gemini allowed Liverpool Larder to leverage sophisticated AI algorithms, making the sentiment analysis more accurate and reliable.

**Integration with the Admin Panel:** The results of the sentiment analysis were displayed in the Admin Panel, allowing administrators to view a summary of customer sentiment. The integration with Bubble enabled the sentiment analysis outcomes to be presented in an accessible manner. This made it easy for administrators to quickly gauge the overall sentiment of recent feedback and identify any emerging issues.

### **Bubble Workflow for Adding a Review and Analysing Sentiment**

When a user submits a review on Liverpool Larder, a specific workflow is triggered to create a new entry in the system and analyse the sentiment of the feedback. This workflow ensures that user reviews are automatically processed and the sentiment is categorised in real time. The following explains the details of how this workflow operates:

The workflow is initiated as soon as a user adds a review on the platform. It includes the following key steps:

#### **Step 1: User Login Requirement**

To ensure the integrity of the reviews and maintain a high level of accountability, only logged-in users are permitted to submit reviews. This step helps in reducing spam and fake reviews, ensuring that the feedback comes from genuine customers who have placed orders through Liverpool Larder.

## Step 2: Creating a New Element in Review Data Type

Once the user submits their review, the workflow creates a new element in the "Review" data type. This entry stores all relevant details, including the text of the review, the user's name, and a timestamp indicating when the review was submitted. This ensures that all reviews are properly organised and can be retrieved for further use, such as sentiment analysis or future reference.

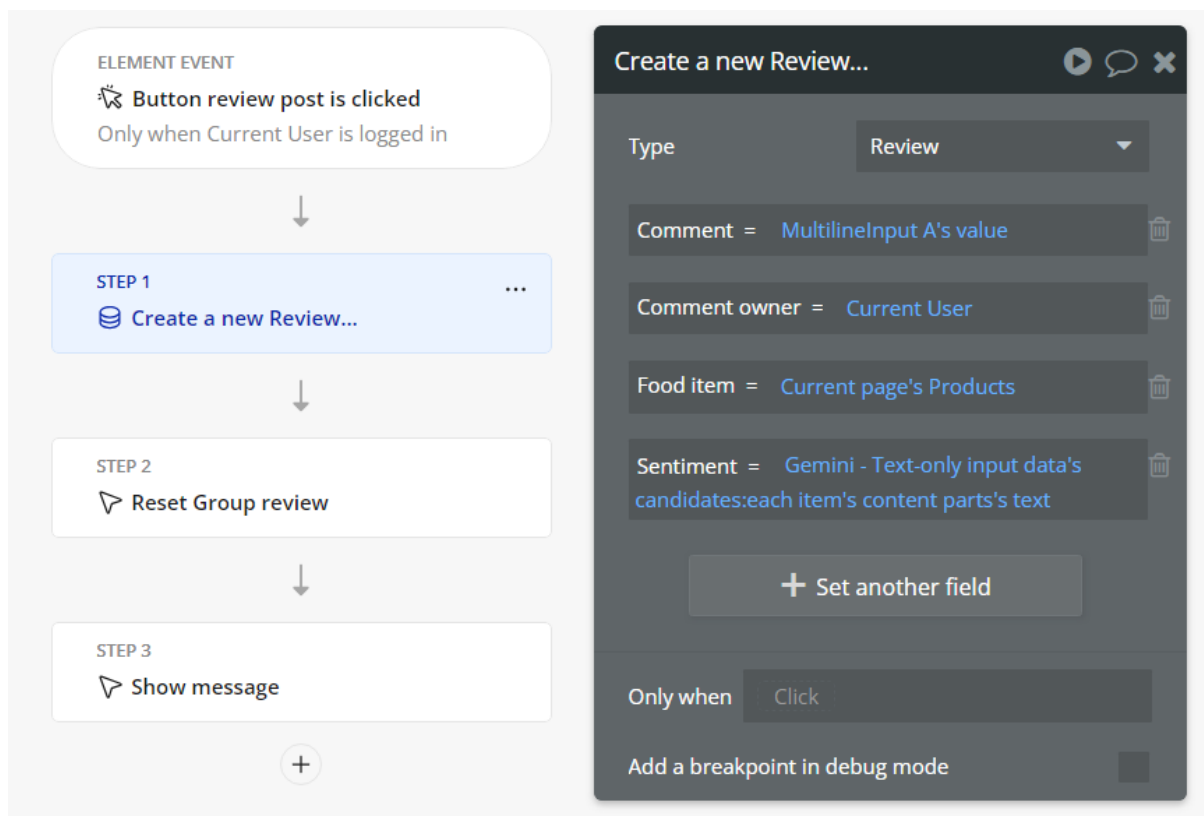


Figure: Workflow for creating a new element in Review Datatype

### Step 3: Calling the Gemini API for Sentiment Analysis

Immediately after the review is stored, the workflow calls the Gemini API to analyse the sentiment of the submitted review. This API call is automatically triggered within the same workflow to maintain real-time processing of feedback.

The API is called with a dynamic prompt that is structured as follows:

#### **Prompt:**

"You are a sentiment analysis tool. Analyze the sentiment of the following sentence and classify it as either 'positive,' 'negative,' or 'neutral.' Sentence: [User review taken dynamically]. Provide a one-word answer indicating the sentiment: positive, negative, or neutral."

In this prompt, the user review is dynamically inserted, allowing the system to adapt to whatever feedback the customer has provided. The Gemini API processes the review text and returns a one-word sentiment classification either positive, negative, or neutral.

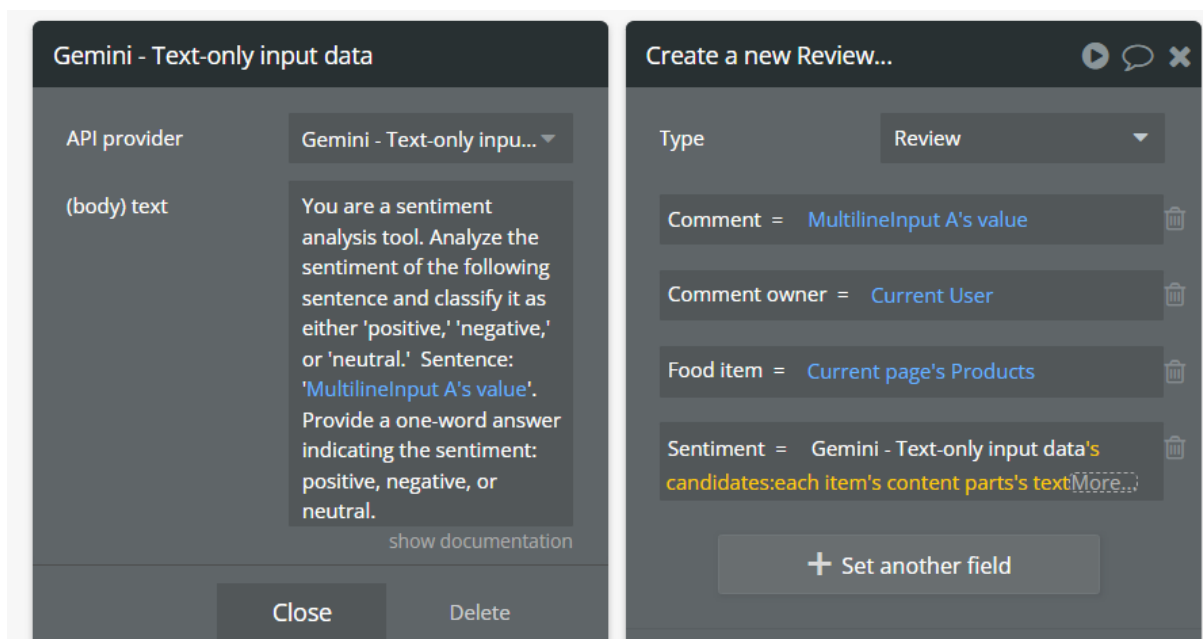


Figure: Calling Gemini API using Bubble workflow for Sentiment Analysis

#### Step 4: Storing the Sentiment Result

Once the Gemini API returns the sentiment classification, the workflow updates the corresponding review entry in the database. The sentiment field is set based on the API response, allowing administrators to view the sentiment of each review through the Admin Panel.

#### Step 5: Display in the Admin Panel

The Admin Panel is updated to reflect new reviews and their associated sentiments in real time. This ensures that administrators can quickly see how customers are responding to their experience. the system assigns a colour code based on the sentiment:

- Green if the sentiment is positive, indicating satisfaction and positive feedback.
- Red if the sentiment is negative, highlighting areas where there might be issues that need attention.

**Name=>Akbar Angelo** | *Comment=>Liverpool Larder's student meals are a lifesaver! As a student, finding affordable yet tasty food is always a challenge, but their meals have been amazing. The portions are generous, the flavors are spot on, and there's a good variety to choose from. It's so convenient for busy days when cooking isn't an option. I love that they cater to different dietary needs as well, making it easy to find something that suits me. Great value for money – highly recommend to any students looking for a quick and satisfying meal!* | **Sentiment=>positive**

**Name=>Vincent Gomez** | *Comment=>I had high hopes for the sausage rolls from Liverpool Larder, but I was quite disappointed. The pastry was too thick and dry, lacking the buttery flakiness I was expecting. The sausage filling itself was bland and under-seasoned, and it seemed to lack the savory punch that makes a sausage roll enjoyable. The overall texture felt heavy and unbalanced, and I found it hard to finish even one. Considering the quality of other items on their menu, I hope they revisit their sausage roll recipe to bring it up to the same standard.* | **Sentiment=>Negative**

Figure: The sentiment analysis page showing customer reviews. The top review was detected as positive, indicated by green, while the bottom review was classified as negative, indicated by red.

The **Sentiment Analysis** feature, implemented using the **Gemini API** within the **Bubble.io** platform, showcases how sentiment analysis can be seamlessly integrated into any website to enhance understanding of customer feedback and acted upon user feedback.

## Designing ChefBot – The Chatbot powered by LLM

The purpose of ChefBot was to serve as a virtual assistant that guided customers through their food ordering process. ChefBot provided real-time assistance, answered user queries, and supported customers throughout their ordering journey. The chatbot's design focused on making the experience as helpful, friendly, and efficient as possible.

### Implementation Approach Using Gemini API

ChefBot was also implemented using the Gemini API, integrated with the Bubble platform to provide intelligent conversational capabilities. This integration allowed ChefBot to deliver a more natural and engaging interaction for users.



Figure: The user asks a question about Liverpool Larder's menu.

## **Bubble Workflow for ChefBot**

The ChefBot chatbot was developed to provide real-time assistance to users by answering questions about Liverpool Larder's menu, including ingredients, pricing, and allergens.

### **Step 1: User Query Initiation**

Whenever a user types a question into the chat window, the workflow begins by preparing a prompt for the Gemini API.

ChefBot uses a predefined prompt that includes current menu data, which is fetched dynamically from the database. This prompt is used for all questions to ensure consistency and accuracy.

#### **Prompt Template:**

"You are an AI assistant named 'ChefBot' helping users with questions about Liverpool Larder cloud kitchen menu. Here is the current menu: **[dynamic data which includes food name, price, ingredients]**. Here is the user question: **[the data filled by user in the input column]**."

The prompt is dynamically updated with the latest data from the menu data type, ensuring all responses reflect the most recent information.

### **Step 2: Call the Gemini API**

Once the dynamic prompt is ready, ChefBot makes an API call to the Gemini API.



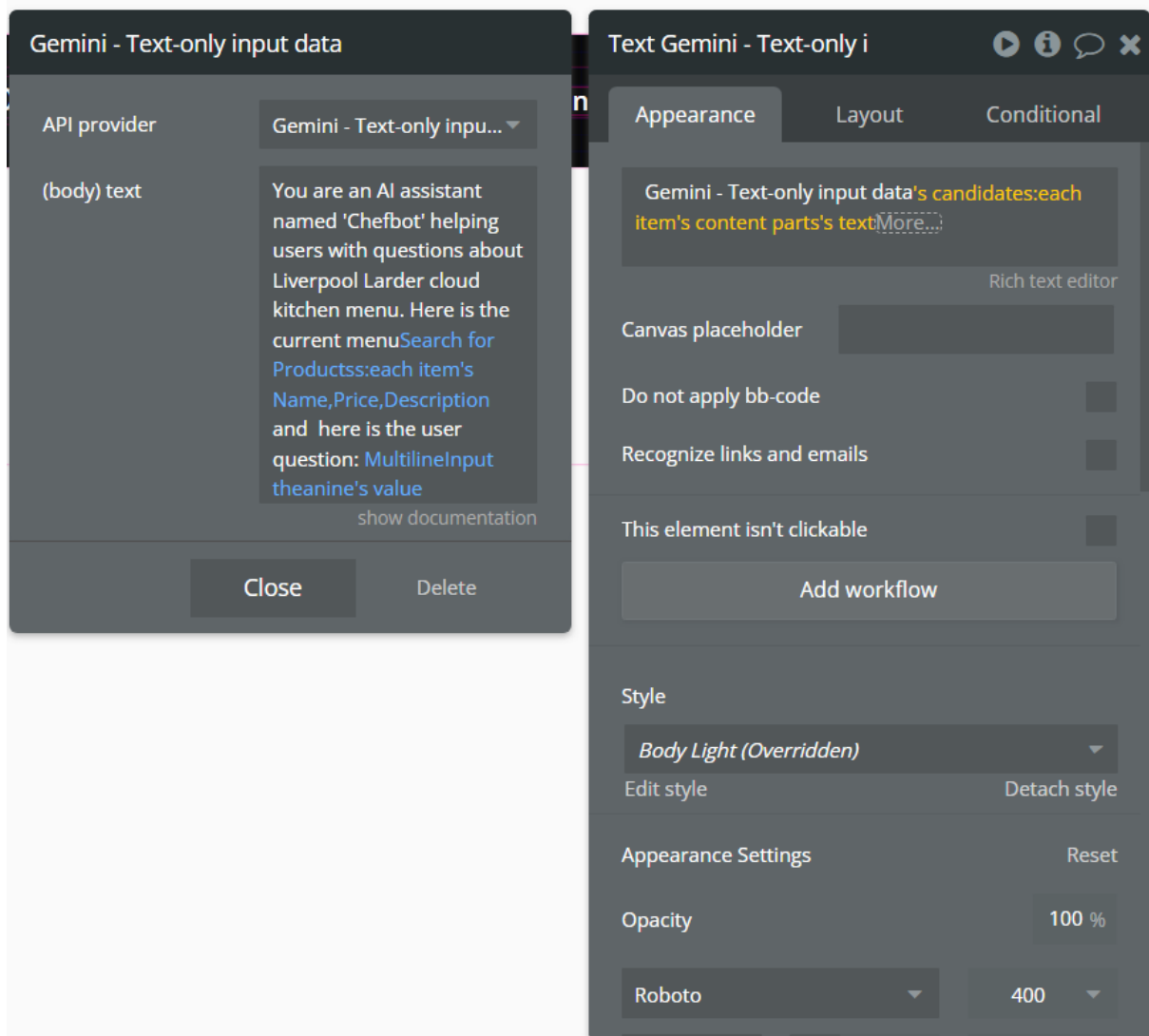


Figure: Calling Gemini API using Bubble workflow for ChefBot

API Workflow in Bubble: ChefBot is connected to the Gemini API through Bubble workflow. The prompt, containing both the user's question and current menu details, is sent to the Gemini API for processing.

The Gemini API uses Natural Language Processing (NLP) to analyse the prompt and generate an appropriate response based on the user's question.

### Step 3: Receive and Display the Response

After processing the prompt, the Gemini API returns a response, which ChefBot then displays to the user.

The generated response is delivered in real time, providing the user with information regarding their query—such as the price of a dish, its ingredients, or any allergens.

#### Step 4: Handling Follow-Up Queries

If the user asks a follow-up question or rephrases their initial query, ChefBot handles the new question in the same way, using the same workflow.

ChefBot uses the same prompt structure each time a new question is asked. The only changes are the specific question asked by the user and the latest menu data from the database.

The new question, along with the current menu data, is sent back to the Gemini API using the same workflow, ensuring consistency across all responses.

This loop continues as long as the user keeps asking questions. By using the same workflow, ChefBot ensures that the answers are consistent, comprehensive, and based on the most current menu information.

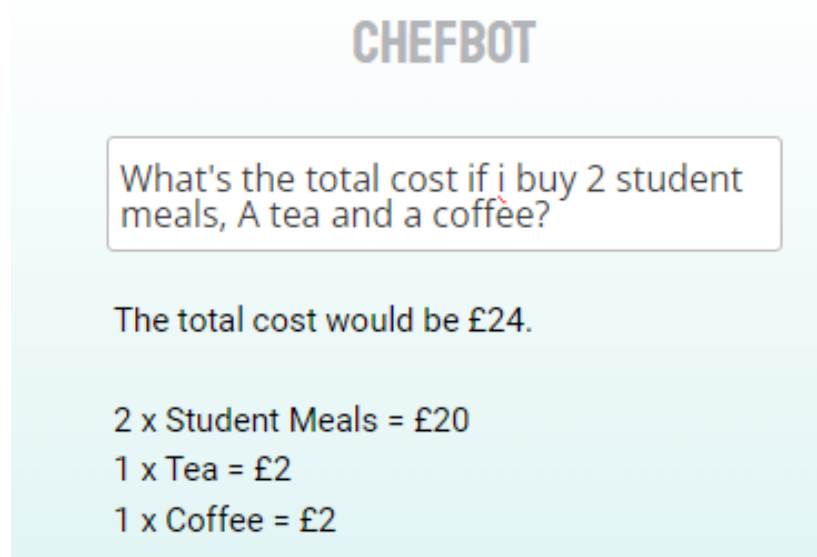


Figure: User interacting with ChefBot

### Step 5: Ending the Conversation

Once the user is satisfied and stops interacting with ChefBot, the chat window remains available for any further questions. Users can close or minimise the chat window at any time. ChefBot will be available if the user decides to ask further questions later on.

## **CONCLUSION**

By leveraging **LLM APIs**, we can significantly enhance any website with minimal coding effort. This project demonstrated how to implement a **sentiment analysis tool** and a **chatbot**, transforming user interactions and feedback into valuable insights. However, the possibilities extend far beyond these examples. With the right dynamic prompts, LLM APIs can be utilised to create a wide range of intelligent, user-focused features, making websites more interactive, responsive, and valuable to their audiences.