## Project: AI BASED INTELLIGENT Waste SEGRIGATOR

## I. Introduction-

This report details the technical architecture and implementation of a **decentralized, AI-powered system for real-time waste segregation**. The project leverages the **Gemini multimodal model** for highly accurate image analysis, integrated via the **n8n backend platform** and deployed on the **Telegram messaging application**. The primary outcome is the creation of a ubiquitous, instant, and user-centric classification service accessible via any smartphone camera. This approach effectively bypasses traditional infrastructural barriers, offering a scalable solution to improve waste diversion rates and reduce occupational hazards associated with manual sorting. The system's novelty lies in its seamless combination of powerful AI intelligence (Gemini) with public-facing accessibility (Telegram) orchestrated through a robust, self-hosted integration layer (n8n). **A core capability is the analysis of images containing multiple distinct waste objects, providing a comprehensive classification for all items in a single user request.**

## 2. Problem Definition and Gap Analysis-

### Problem Statement:

The global challenge of inefficient solid waste management is characterized by fragmented processes and low resource recovery rates. Manually intensive and inconsistent segregation at the source results in high volumes of mixed waste, leading to increased landfill burden, environmental pollution (soil and water contamination), and elevated economic costs for municipal operations. Furthermore, relying on manual segregation poses significant **labour welfare concerns and occupational hazards**, exposing waste pickers and facility workers to biological and chemical risks. The critical gap is the lack of an immediate, universally accessible, and accurate tool for educating and guiding citizens on proper waste categorization *at the moment of disposal*.

### Comparative Analysis: AI vs. Traditional Segregation:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Traditional Manual Segregation** | **AI-Powered System (Gemini/n8n)** |
| **Speed** | Highly variable, dependent on human throughput. | **Fast, Consistent** (within one minute response time). |
| **Accuracy** | Subject to human error, fatigue, and lack of specialized knowledge. | **High Accuracy**, leveraging multimodal large models trained on vast datasets. |
| **Data Collection** | None, or highly inconsistent paper-based logging. | **Automatic, structured data generation** for waste types and geolocations. |
| **Occupational Hazard** | **High** (Exposure to toxins, sharps, and biohazards for high time ). | **Low** (Classification is easy and fast ). |
| **Scalability** | Linear scaling, requires high capital expenditure (labour/infrastructure). | **Exponential scaling** via cloud/self-hosted infrastructure; user-base is virtually unlimited. |

### The Need for a Decentralized Solution:

The selection of **Telegram** as the primary User Interface (UI) is a strategic decision to enable maximum community penetration. Traditional waste management solutions often require dedicated mobile applications or expensive hardware installations. Telegram, being a **widely available, easy-to-use, and open-source** platform, allows the system to be deployed instantly, leveraging simple entry methods such as a **QR code scan**. This decentralized approach minimizes the barrier to entry, ensuring that nearly any citizen with a smartphone can access the classification service, accelerating public engagement and source segregation compliance.

## 3. Project Objectives, Scope, and Novelty-

### Project Objectives:

The primary objective was to architect, develop, and deploy a robust, end-to-end classification pipeline capable of receiving an image of waste material and returning its correct, actionable segregation categories in real-time. The project aimed to prove that an integrated architecture—specifically **n8n, Gemini, and Telegram**—could serve as a viable, scalable, and low-cost mechanism for deploying complex multimodal AI services directly to the public.

### Scope of Work:

The project's scope was precisely defined to establish a robust, end-to-end classification pipeline based on three core in-scope activities. The first is **Real-Time Image Analysis**, executed using the Gemini multimodal model to accurately interpret waste material, including the critical capability to detect and analyse multiple distinct objects within a single submission. The second activity is **Detailed Segregation Categorization**, implemented to rigorously map each identified waste item to specific, actionable processing streams based on the 13-category framework. Finally, the **User Feedback Loop** constitutes the delivery mechanism, ensuring the sanitized, accurate, and actionable classification result is instantly delivered to the user via Telegram.

### Novel Aspects / Key Innovation:

The core innovation is the creation of an **automated, user-driven, multimodal AI classification service** deployed entirely through a **backend platform**/no-code framework. This integration is novel because it connects a cutting-edge multimodal intelligence layer (**Gemini**) to a globally ubiquitous messaging service (**Telegram**) using a self-hosted, easily maintainable integration platform (**n8n**). This architecture democratizes access to advanced machine vision, shifting the intelligence from static, centralized hardware (e.g., in a sorting facility) to a dynamic, decentralized utility available to the end-user. **Crucially, the Gemini model's multimodal capacity allows it to process and categorize multiple distinct items within a single submitted photograph, maximizing user efficiency.**

## 4. Technical Architecture and Workflow Deep Dive-

### Tools, Technologies, and Platforms Used:

|  |  |
| --- | --- |
| **Technology** | **Role in System** |
| **n8n** | **Integration Layer/Orchestrator.** Manages the workflow, data routing, error handling, and hosting of the logic nodes. |
| **Google Gemini Model** | **Intelligence Layer.** Multimodal AI engine responsible for visual analysis, identification of waste items, and classification based on the strict System Message. |
| **Telegram API** | **User Interface (UI) Layer.** Provides the public entry point (via Telegram Bot) and the communication channel for delivering results. |
| **Docker** | **Deployment Container.** Used to self-host the n8n application, ensuring portability and isolated environment management. |
| **ngrok** | **Networking.** Provides a secure HTTPS tunnel to expose the self-hosted n8n webhook endpoint to the public Telegram API. |

### System Components and Architecture Overview:

The system operates across three distinct layers:

1. **User Interface Layer (Telegram Bot):** The user initiates the process by sending a photo of waste to the dedicated Telegram Bot.
2. **Integration Layer (n8n Workflow):** The Telegram Trigger node receives the webhook event, passes the image data to the AI Agent, receives the raw text output, post-processes it, and routes the final, sanitized message back to Telegram.
3. **Intelligence Layer (Gemini/AI Agent):** The AI Agent, connected to the Gemini Chat Model, performs the computationally intensive task of image recognition and waste category classification based on the predefined constraints.

## Workflow Description:

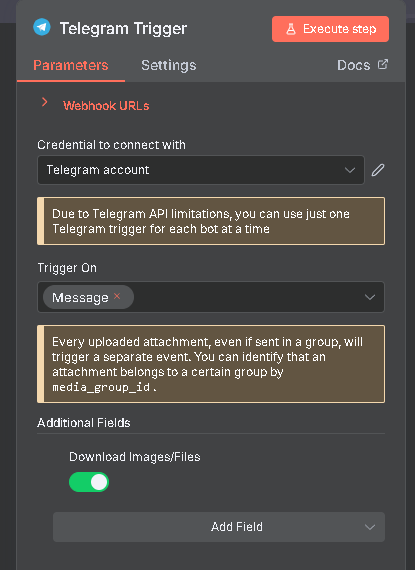
The WSAI workflow follows a linear, four-node execution path, supplemented by two configuration nodes:

1. **Telegram Trigger:** Initiates the workflow upon receiving a user message (containing an image).
2. **AI Agent:** Uses the Google Gemini Chat Model and Simple Memory (for basic state management) to process the image input and generate the segregation advice.
3. **Code in JavaScript:** Sanitizes the raw AI output string to ensure strict HTML compliance for Telegram.
4. **Send a text message1:** Delivers the final, formatted message back to the user via Telegram.

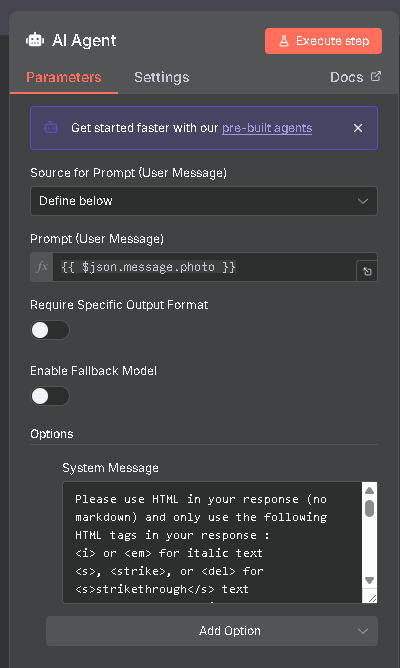
workflow image


### Detailed n8n Workflow Analysis (Node-by-Node Explanation):

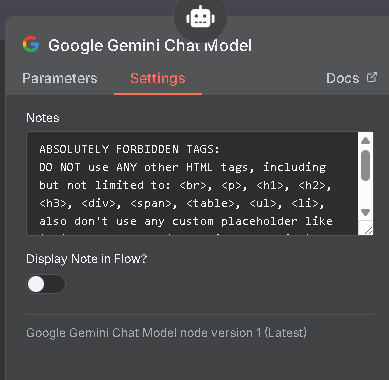
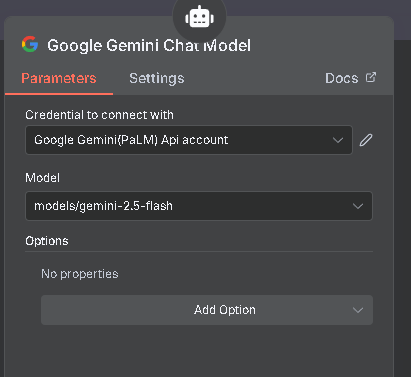
1. **Telegram Trigger:**
   * **Role:** The primary entry point for the workflow.
   * **Configuration:** Configured to listen for updates: ["message"]. Crucially, it sets download: true under additional Fields, ensuring that when a photo is received, the node downloads the file data and attaches it to the workflow item, making the image content available for the subsequent AI processing node.



1. **AI Agent:**
   * **Role:** Orchestrates the multimodal classification task.
   * **Input Data:** The node receives the image data via an n8n expression: ={{ $json.message.photo }}. This expression dynamically injects the file data (downloaded by the Telegram Trigger) into the AI model's prompt for visual analysis.
   * **System Message Analysis:** This message imposes severe constraints on the Gemini model, enforcing the persona of a "professional waste segregator." It strictly dictates the use of a minimal subset of HTML tags (<i>, <em>, <s>, <u>, etc.) and explicitly forbids newline characters and common HTML tags to force the model to produce a response compatible with Telegram's HTML parsing mode, relying on post-processing for line breaks. It directs the model to classify waste based on a comprehensive, non-overlapping list of **13 categories**: Organic, non-organic, degradable, non-degradable, plastic, non-plastic, recyclable, non-recyclable, hazardous, wet, electronic, dry, and chemical wastes.



1. **Google Gemini Chat Model & Simple Memory:**
   * **Language Model:** The AI Agent is connected to the **Google Gemini Chat Model** for generating the classification text.
   * **Memory Context:** The **Simple Memory** node is configured with a context Window Length of 1. Coupled with a session Key based on the Telegram chat.id, this ensures the system is **stateless** and **single turn**. Each user photo initiates a fresh classification request, preventing context drift and resource misuse.
   * **Setting Note of language Model:** setting note is configured for **Google Gemini Chat Model with** instructions avoiding tags or functions which are unsupported for Telegram API key messages for sending messages.



A screenshot of a computer

AI-generated content may be incorrect.

1. **Code in JavaScript:**
   * **Role:** Performs essential data sanitation and output reformatting, acting as a critical bridge between raw AI output and strict API requirements.
   * **Function:** Cleans the raw output from the AI Agent, which, despite the negative prompt, often includes unwanted newlines or placeholder remnants. Its primary task is to convert these into the explicit <br> tag required for proper line breaks in the final Telegram message.

#### CRITICAL: Custom JavaScript Code Logic:

The final Html Strip(items) function is a critical middleware component ensuring API compatibility and stable output presentation. Its logic is a custom-engineered solution to resolve inconsistencies between the AI model's raw text generation and the Telegram API's rigid HTML subset requirements. The primary goal of this JavaScript node is to sanitize the AI's output by transforming unsupported structural HTML elements (like <ul> and <li> which the AI may generate despite negative prompting) into simple, visible text and line breaks (<br>) that Telegram's HTML parsing mode can successfully render. This process ensures the classification output is readable without triggering API parsing errors.

To view the code of the JavaScript click on the link: [**JavaScript code n8n**](https://github.com/aviangi/waste-segregationAI-files/blob/main/javascript%20code%20n8n.txt)

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
| **Manipulation** | **Regex/Action** | **Rationale** |
| **List Start Conversion** | replace(/<ul[^>]\*>/g, '<br>') | Converts unsupported opening <ul> tag to a line break (<br>). |
| **List End Conversion** | replace(/<\/\ul>/g, '<br><br>') | Converts unsupported closing </ul> tag to two line breaks. |
| **List Item Start** | replace(/<li[^>]\*>/g, '— ') | Converts unsupported opening <li> tag to a visible dash/space (— ). |
| **List Item End** | replace(/<\/\li>/g, '<br>') | Converts unsupported closing </li> tag to a line break. |
| **Span Tag Removal** | `replace(/<span[^>]\*> | It removes all <span> tags from the text so the content can be sent to Telegram without formatting errors. |
| **Newline Fix** | replace(/\\n/g, ' ') | Replaces JSON-encoded newlines (\n) that sometimes appear in the raw output with a space for cleaner processing. |
| **Entity Decode** | replace(/–/g, '—') | Decodes the HTML entity for the en-dash to a standard em-dash character. |

1. **Send a text message1:**
   * **Role:** Final output delivery.
   * **Configuration:** The node dynamically retrieves the chat Id and pulls the cleaned final message. Crucially, parse mode: "HTML" is set, instructing the Telegram API to interpret the incoming text as HTML, allowing the sanitized line breaks and basic formatting tags to render correctly.

A screenshot of a chat

AI-generated content may be incorrect.

1. JSON Code:

The project ( n8n ) is exported in a n8n code format which people can access from this link: [**n8ntelegram workflow. Json**](https://github.com/aviangi/waste-segregationAI-files/blob/main/n8ntelegram%20workflow.json).

People can view full project in code format (Json) [n8n format code] in **Visual Studio Code** (VS Code) for better understanding in code format. This code helps anyone import this project though this code and make any modifications or any upgrade on existing project. People can import this in n8n to test and experiment with this project.

## 5. Implementation and Development Process-

**Methodology:** The project utilized an **iterative prototyping methodology** focused heavily on prompt engineering and integration debugging within the **n8n backend environment**.

1. **Initial Prototype:** Basic integration established the core data flow: Telegram Trigger \right arrow AI Agent (simple text prompt).
2. **Prompt Refinement:** Iteratively testing and refining the Gemini **System Instruction** to ensure accurate image classification and categorization, particularly focusing on defining the **professional waste segregator persona** and the comprehensive list of 13 waste categories.
3. **Integration Debugging (The Critical Phase):** This involved extensive testing of the raw AI output against Telegram's API constraints. This critical phase necessitated the creation of the custom **Code in JavaScript** node to resolve incompatibilities found in list elements and rogue formatting tags, ensuring a stable, compliant output channel capable of rendering basic HTML formatting and proper line breaks.

The core achievement is the successful deployment of a self-hosted, scalable, multi-modal AI classification service accessible to any smartphone user via a common messaging app, demonstrating a proof-of-concept for citizen-level environmental AI utility.

## 6. Use-Cases and Key Results-

### User Workflow:

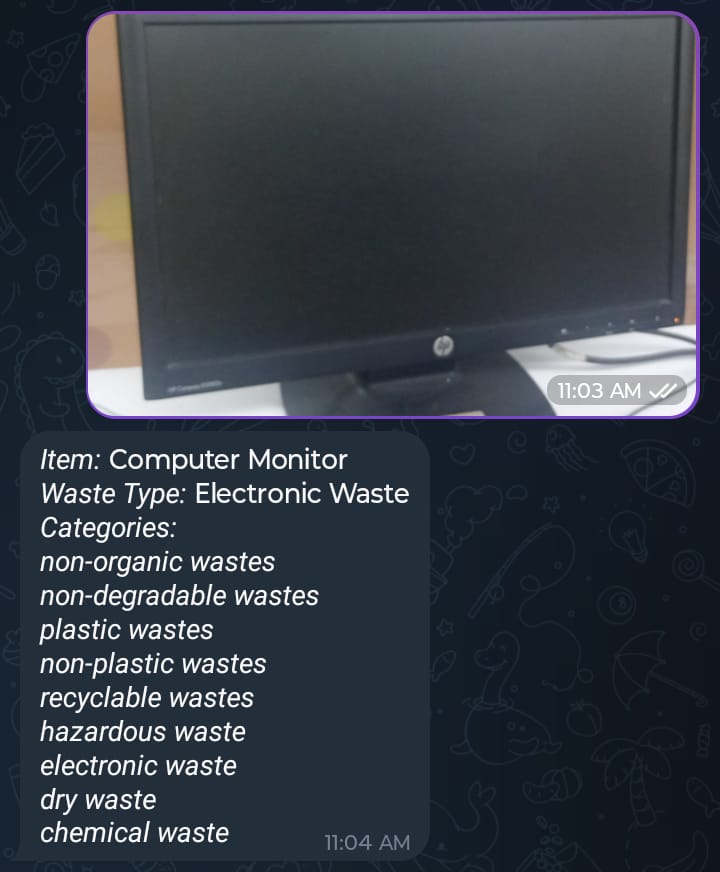
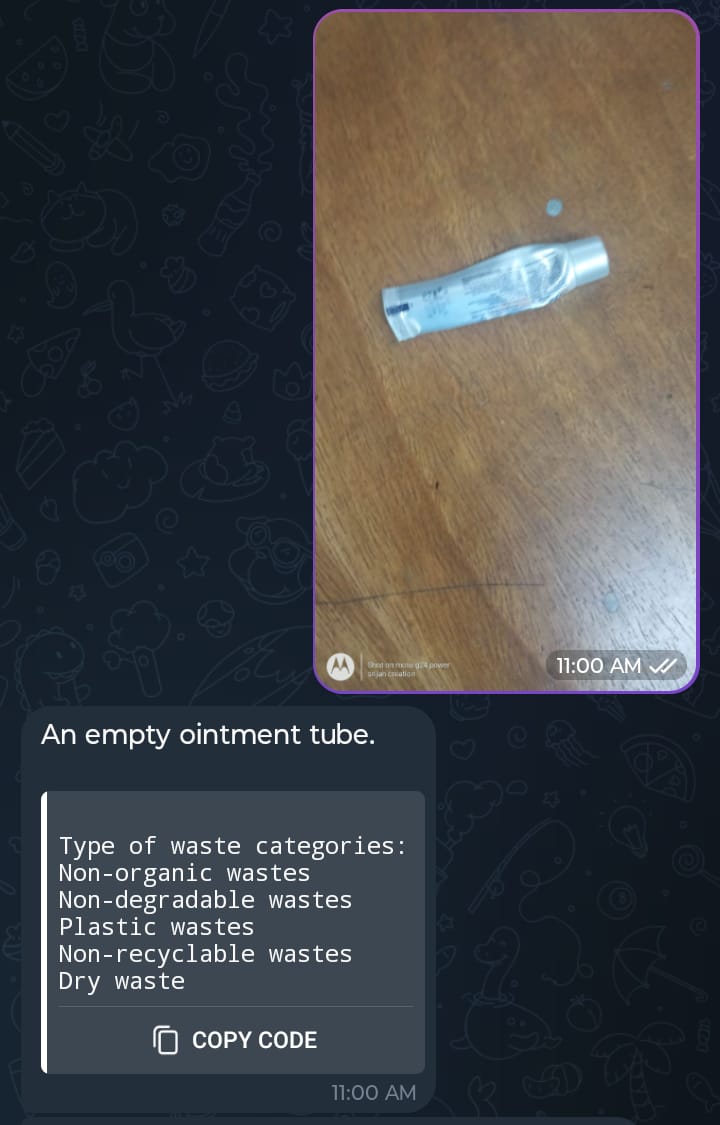
1. **System Entry:** A user scans a published QR code (or accesses the link) associated with the Telegram Bot.
2. **Input:** The user takes a photograph of the item(s) of waste they wish to dispose of and sends the image to the bot. **The image may contain multiple distinct waste items.**
3. **Processing:** The Telegram Trigger initiates the n8n workflow, which passes the image to the Gemini AI Agent for classification against the 13 categories.
4. **Output:** The system returns a concise, single-line summary listing the correct waste categories (e.g., "Recyclable, Plastic, Non-degradable, Dry Waste") directly to the user in the Telegram chat **within one minute**.

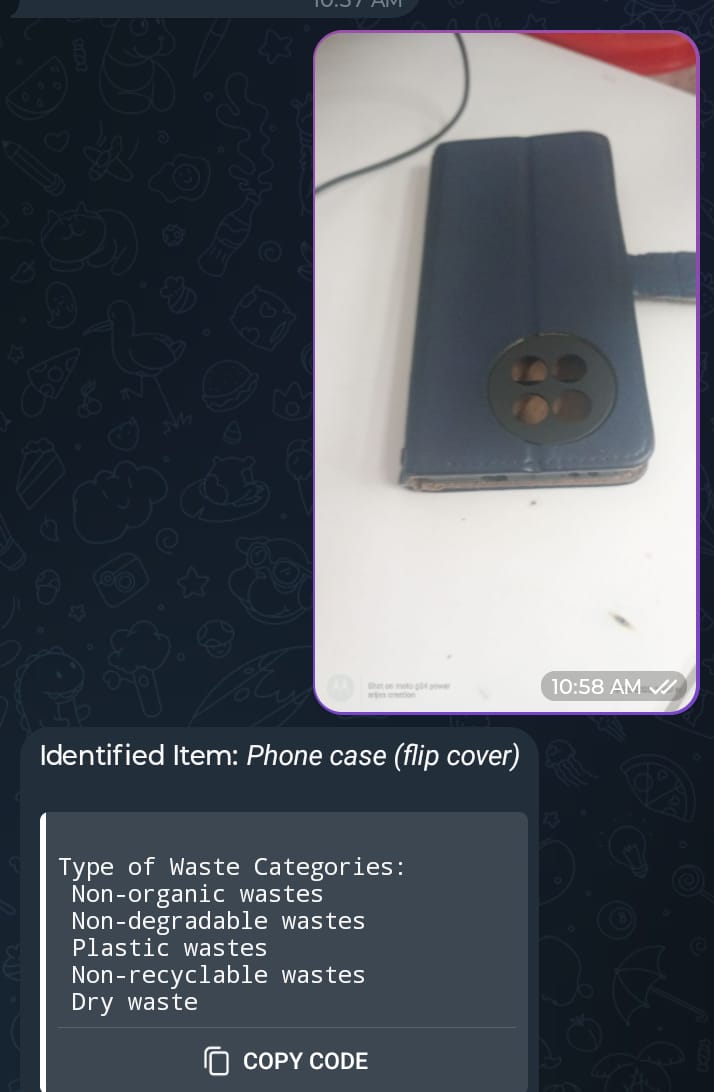
### Key Results/Outcomes:

The deployed system provides tangible benefits:

* **Fast Classification:** Users receive actionable segregation guidance within one minute, solving the "moment of disposal" problem.
* **High Accuracy:** The use of the Gemini multimodal model ensures high accuracy in identifying diverse waste items.
* **Multi-Object Analysis:** The system is capable of analysing and classifying **multiple distinct waste objects within a single photograph**, drastically reducing the number of submissions required for complex disposal tasks.
* **Accessibility:** The Telegram-based interface achieves maximum accessibility, requiring no special hardware or application download.
* **Data Generation Capability:** Each request generates structured metadata (classification, time, user ID), laying the foundation for municipal data collection on community waste patterns.

## Output Examples-



A group of boxes with clothes

AI-generated content may be incorrect.A screenshot of a cell phone

AI-generated content may be incorrect.A screenshot of a phone call

AI-generated content may be incorrect.Screens screenshot of two puppies

AI-generated content may be incorrect.A close-up of a pen

AI-generated content may be incorrect.

## 8. Limitations-

The project runs on a self-hosted server which runs on a local computer. We need a server which runs 24 hours which would let this workflow always accept user’s request and remains always active 24 hours. It also has a limitation of input pictures as user can only upload one picture at a time. For making this project production level and always active for user accepting user’s requests we need a live 24-hour active server.

## 9. Conclusion and Key Takeaways-

The **AI-Powered Real-Time Waste Segregation** project successfully integrates state-of-the-art multimodal AI with a decentralized public access platform, demonstrating a powerful new model for environmental technology deployment. By leveraging the **n8n backend environment** to manage the Telegram-Gemini API handshake, the project created a scalable, low-friction, and highly accurate system. This technical achievement has a direct, positive impact on waste management efficiency and significantly enhances public engagement in source segregation, ultimately contributing to higher recycling rates and reduced reliance on hazardous manual sorting processes. The self-hosted, open architecture ensures maintainability and future adaptability for additional features, such as localized disposal instructions or gamification. The key takeaway is the successful application of an **API-first, backend orchestration layer (n8n)** to commoditize access to a **high-intelligence multimodal model (Gemini)** for a pressing public utility.