## Project: AI BASED INTELLIGENT Waste SEGRIGATOR

## 1. Executive Summary-

The **Waste Segregation AI (WSAI)** project successfully implements a real-time, decentralized, point-of-disposal waste classification system. The solution leverages **Google Gemini's multimodal capabilities** for image analysis, orchestrated by the **N8N automation platform**, and deployed through a **Telegram chatbot** user interface. The primary objective is to provide immediate, highly accurate segregation guidance based on user-submitted photos of waste items. This architecture offers a significant benefit over traditional methods—such as static charts or manual lookup—by delivering standardized, accessible, and instantaneous advisory services, dramatically reducing confusion and cross-contamination rates in waste streams.

## 2. Introduction & Project Definition-

## 2.1. Problem Statement:

A significant gap exists in modern municipal and consumer-level waste management, primarily rooted in **user confusion and inconsistency**. Consumers, faced with complex local segregation rules, often default to incorrect disposal (e.g., "wish-cycling" or cross-contamination), leading to substantial economic and environmental costs. Municipal workers and recycling centres suffer from increased processing complexity and reduced material purity. The opportunity lies in deploying advanced, easily accessible Artificial Intelligence to bridge this knowledge gap, providing dynamic, real-time segregation advice precisely at the point of disposal.

## 2.2. Project Objectives & Novelty:

The project's objective was to build an end-to-end, zero-setup system that could:

1. Accept a photograph of a waste item via a ubiquitous messaging platform (Telegram).
2. Analyse the image using a powerful multimodal AI (Gemini).
3. Return a structured, categorized, and compliant segregation report to the user.

This approach is highly novel due to its specific technical integration: it combines **multimodal AI analysis** with a consumer-grade chat platform via an **integration middleware (N8N)**, bypassing the need for a custom mobile application or web portal. The system's robustness is achieved through specific prompt engineering and post-processing logic designed to ensure the AI's output is perfectly compatible with the Telegram API's strict HTML parsing rules.

## 3. Solution Overview-

## 3.1. Uses, Advantages, and Practicality:

|  |  |  |
| --- | --- | --- |
| **Aspect** | **WSAI Solution** | **Normal, Manual Process** |
| **Uses** | Public education, pre-segregation audits, real-time guidance, standardization across jurisdictions. | Static charts, physical signage, periodic workshops. |
| **Advantages** | Immediate feedback (seconds), high accessibility (Telegram), standardization of categorization, scalable globally with no infrastructure cost. | Dependent on user memory, inconsistent interpretation, low engagement, difficult to update. It is also 50-60% slower. |
| **Practicality** | High efficiency. Requires only a camera and Telegram access. The system is always available and instantly adaptive to any item. | Low efficiency. Requires manual search or memorization; prone to human error and guesswork. |

## 3.2. Key Results / Outcomes:

The system is designed to achieve:

* **Improved Segregation Accuracy:** Near-perfect classification of common waste items due to multimodal AI capability.
* **Reduction in Cross-Contamination:** Clear, unambiguous advice prevents mixing of waste types.
* **Increased Public Engagement:** Utilizing a familiar platform (Telegram) lowers the barrier to entry, encouraging use.
* **Standardized Categorization:** Ensures all users receive guidance based on a consistent, predefined set of waste types.

## 3.3. Who it Will Serve:

* **Users:** Residents, K-12 students, university campuses, workers in corporate offices and common people aiming for improved environmental compliance.
* **Organizations:** Municipal Corporations seeking to reduce downstream sorting costs, Waste Management Companies requiring higher material purity, and Educational Institutions integrating sustainability programs.

### 4. Technology & Architecture Overview-

The WSAI architecture utilizes a focused, serverless approach centred on N8N orchestration and Google Gemini AI Model.

#### 4.1. Scope & Toolset:

The in-scope functionality covers the entire request-response cycle initiated by an image submission.

|  |  |
| --- | --- |
| **Technology/Platform** | **Role** |
| **N8N** | Integration and Orchestration Platform (Workflow Engine) |
| **Telegram API** | User Interface (Chatbot Frontend) & Input/Output Channel |
| **Google Gemini Chat Model** | Multimodal AI Service (Image Analysis, Classification, and Text Generation) |
| **JavaScript** | Custom Code for Output Data Sanitization and Formatting |

## 4.2. System Components, Modules, and Interfaces:

1. **User Interface (Telegram Chatbot):** Acts as the entry point and delivery interface. It handles image upload and displays the final, compliant HTML segregation report.
2. **Integration Platform (N8N):** The core middleware. It manages the webhook trigger, handles the connection logic between Telegram and Gemini, and performs necessary data transformations (the custom JavaScript node).
3. **AI Service (Gemini Model):** The backend intelligence, connected via the Google Gemini Chat Model helper node. It ingests the image data from Telegram, executes the sophisticated classification logic defined in the System Instruction, and generates the raw HTML response.

## 4.3. Novel Aspects / Innovation:

The most innovative aspect is the precise **Prompt Engineering and Post-Processing Pipeline** designed to overcome the integration incompatibility between a free-form AI text generator and a strictly-validated API output. The use of a custom-constrained multimodal prompt combined with a targeted JavaScript clean-up routine (Section 5.2.C) ensures a robust, production-ready output, a method rarely seen in public N8N templates.

## 5. Detailed N8N Workflow and Code Analysis-

## 5.1. Workflow Description:

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

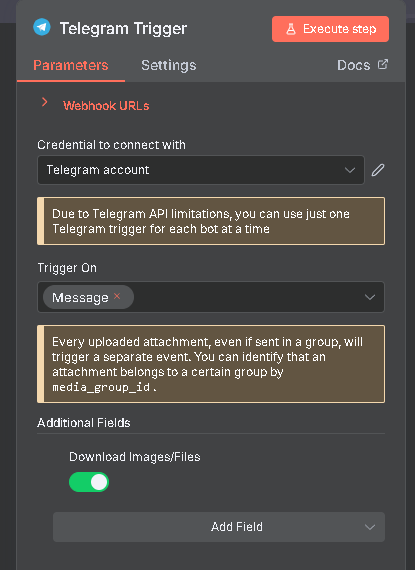
1. **Telegram Trigger:** Initiates the workflow upon receiving a user message (containing an image).
2. **AI Agent:** Uses the Google Gemini Chat Model and Simple Memory (for basic state management) to process the image input and generate the segregation advice.
3. **Code in JavaScript:** Sanitizes the 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


#### 5.2. Node-by-Node Analysis:

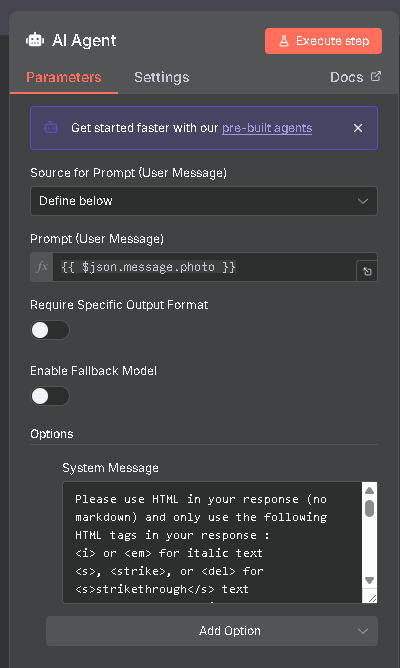
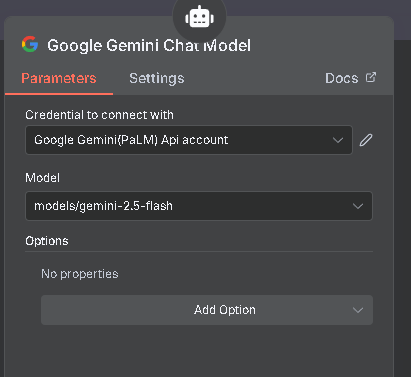
**A. Telegram Trigger**

* **Function:** Serves as the webhook entry point. It monitors for message updates.
* **Key Configuration:** The additional Fields: { "download": true } setting is critical. This ensures that when a user sends a photo, N8N automatically downloads the file data from Telegram, making the image content accessible to the subsequent AI Agent node for multimodal processing.



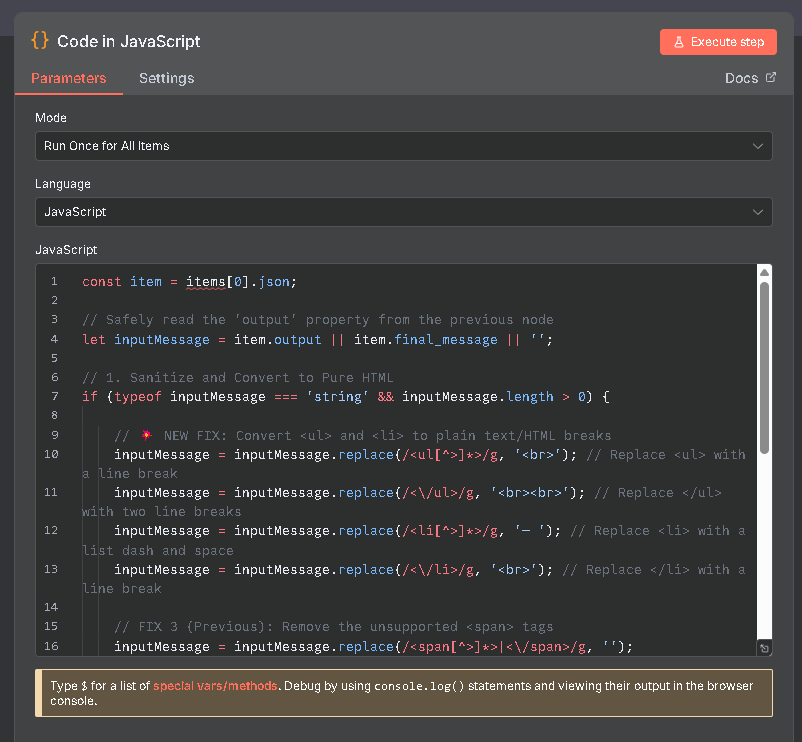
**B. AI Agent (Gemini Integration):**

* **Input Mapping:** The user does **not** need to type any text prompt. The input is dynamically mapped using text: "={{ $json.message.photo }}". This expression extracts the Base64-encoded image data blob (or file reference) passed from the Telegram Trigger. The *image data* serves as the only user input for this request, with the task fully defined by the fixed System Instruction.
* **CRITICAL: System Instruction (Prompt) Analysis:** This detailed instruction (the system prompt) serves two purposes: defining the AI's content generation logic and enforcing strict output formatting.
  + **Persona & Task:** "you are the professional waste segregator," tasked to "Analyse and identify the item correctly (single one) which is in image and identify the type of waste."
  + **Classification Mandate:** The model is strictly constrained to categorize the item using a predefined list of 13 categories (e.g., Organic, non-organic, recyclable, hazardous, wet waste, etc.) and is asked to provide a "summary or concise list instead of a detailed breakdown."
  + **CRITICAL HTML Output Constraint:** This is the core constraint for Telegram compatibility:
    - **Allowed Tags:** ONLY <i> or <em>, <s>, <strike>, or <del>, <span class="tg-spoiler"> or <tg-spoiler>, <a href="URL">, <code>, and <pre>.
    - **Explicitly Forbidden (Strictly):** <p>, <ul>, <b>, and the raw newline character <br> (and implicitly \n).
    - **Rationale:** The instruction forces the LLM to use the minimal set of HTML tags supported by Telegram's parse mode: "HTML", mitigating output errors.



**C. Code in JavaScript:**

* **Function:** This node acts as a **sanitization and conversion layer** to normalize the raw output from the AI Agent, which, despite the system instruction, may sometimes generate unsupported tags like list elements.



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

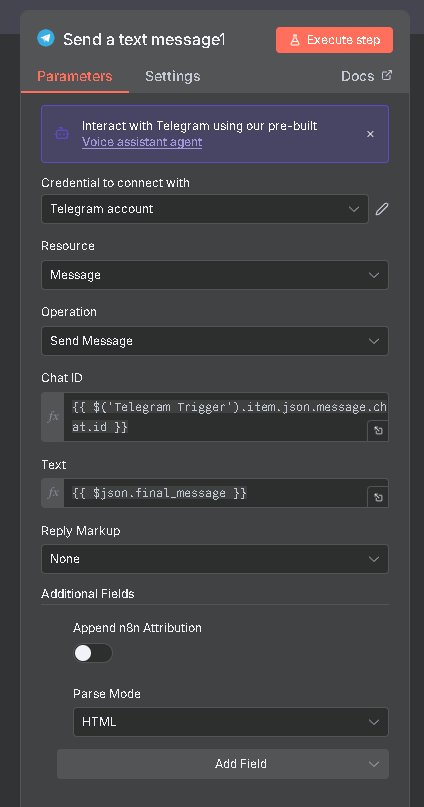
* **CRITICAL: Custom JavaScript Code Logic:**

|  |  |  |
| --- | --- | --- |
| **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[^>]\*> | </span>/g, '')` |
| **Newline Fix** | replace(/\\n/g, ' ') | Replaces JSON-encoded newlines (\\n) that sometimes appear in the raw output with a space. |
| **Entity Decode** | replace(/–/g, '—') | Decodes the HTML entity for the en-dash to a standard character. |

* **Necessity:** This step is crucial. If the AI generates an unsupported tag (like <ul>), the Telegram API will reject the entire message. This custom JavaScript code ensures that even if the AI deviates slightly from the strict instruction, the output is structurally converted into a strictly compliant format before transmission.

**D. Send a text message:**

* **Function:** Sends the final, cleaned message back to the user's chat ID.
* **Mapping:** The output from the Code node, stored in final message, is mapped as the content: text: "={{ $json.final\_message }}".
* **Key Configuration:** The setting parse mode: "HTML" is mandatory. It instructs the Telegram API to interpret the message content as HTML, enabling the rendering of the structured text defined by the AI and cleaned by the JavaScript node.



## E. 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.

## 6. Implementation / Development Process-

## 6.1. Methodology:

The project utilized an **iterative prototyping methodology** focused heavily on prompt engineering and integration debugging.

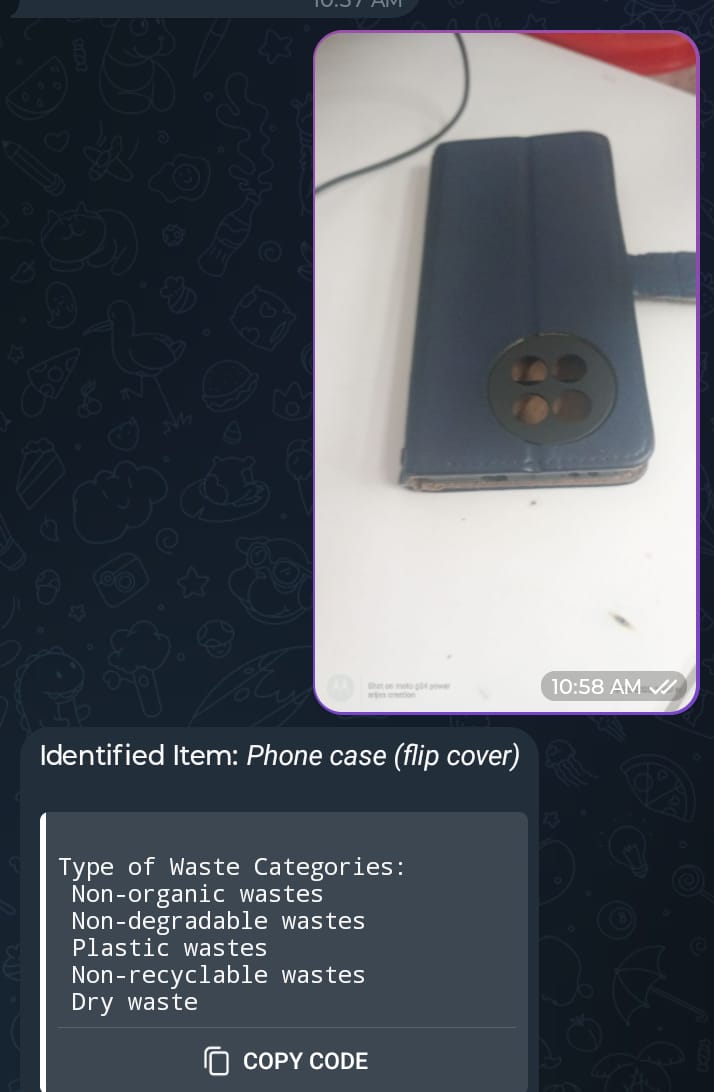
1. **Initial Prototype:** Basic integration of Telegram Trigger -> AI Agent (simple text prompt).
2. **Prompt Refinement:** Iteratively testing and refining the Gemini System Instruction to ensure accurate image classification and categorization, defining the *professional waste segregator* persona.
3. **Integration Debugging (The Critical Phase):** This involved extensive testing of the raw AI output against Telegram's API constraints, which 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.

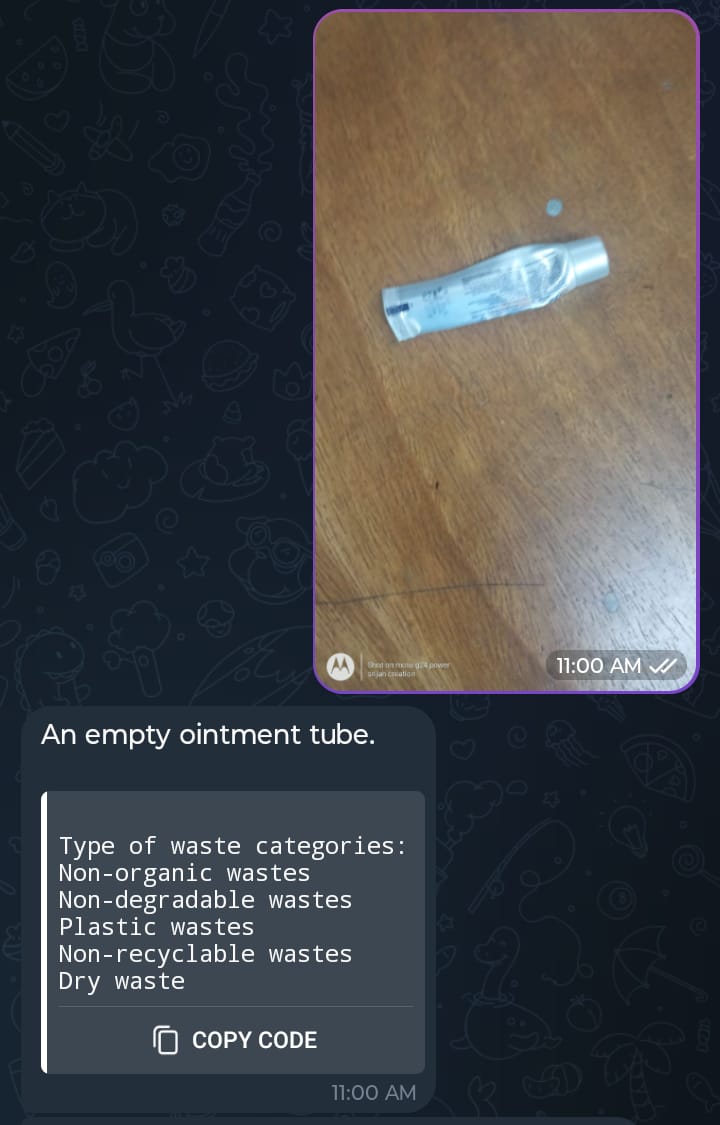
## 6.2. Use-Cases / User Workflow:

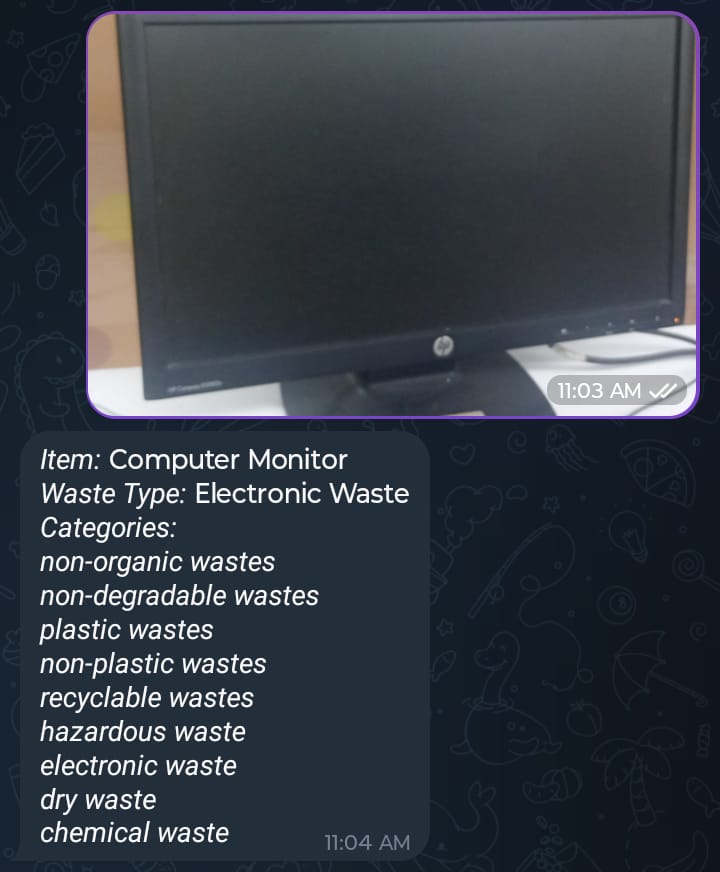
The user workflow is designed for maximum simplicity and accessibility:

1. **QR Scan:** The user scans a QR code or taps an invite link to open the WSAI Telegram Bot.
2. **Open Chat:** The user initiates the chat (e.g., with /start).
3. **Take Photo:** The user takes a picture of the waste item they wish to dispose of.
4. **Send Photo:** The user sends the photo to the chat bot.
5. **Receive Segregation Report:** The user receives a fully formatted, compliant report detailing the identified item, its waste category (e.g., *Recyclable*, *Dry Waste*), and further guidance.

## Output Examples-







## 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 & Key Take-Aways-

The WSAI project successfully transitioned a cutting-edge multimodal AI system into a practical, scalable, and highly accessible public service tool. The primary achievement was the seamless, reliable integration of the Google Gemini model with the Telegram ecosystem, a feat that required sophisticated prompt engineering coupled with a custom data sanitization layer (the JavaScript node) to enforce the strict formatting requirements of the messaging platform. This project demonstrates a powerful paradigm for deploying AI: the creation of a **scalable, user-friendly tool that addresses a critical environmental challenge**—waste mismanagement—by offering immediate, standardized, and intelligent guidance at the point of action. The WSAI model is ready for deployment and immediate impact on waste segregation accuracy.