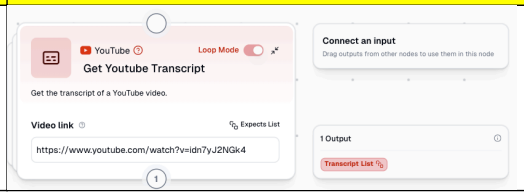
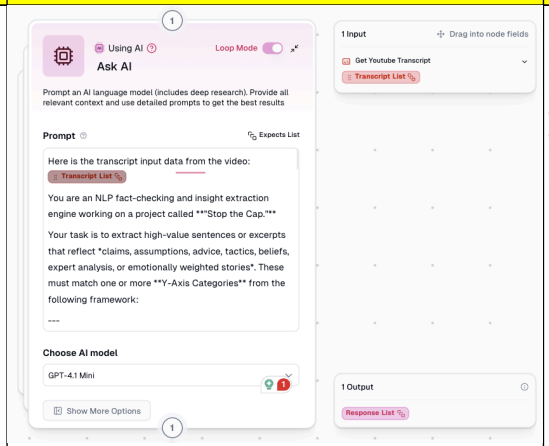
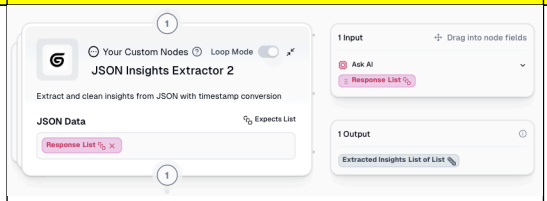
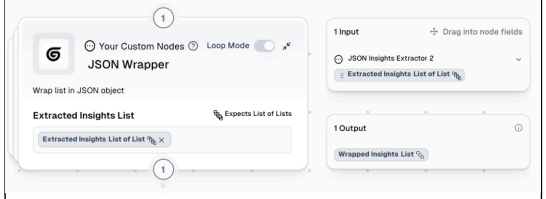
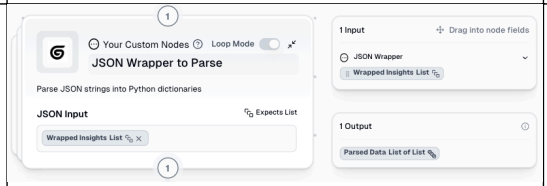
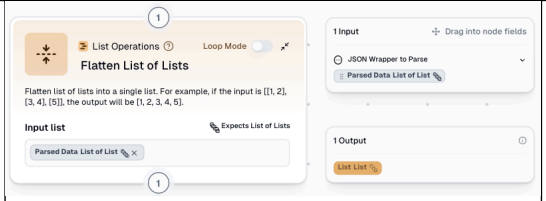
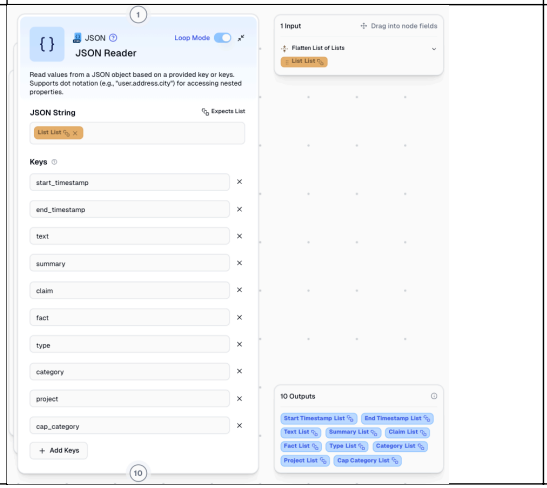
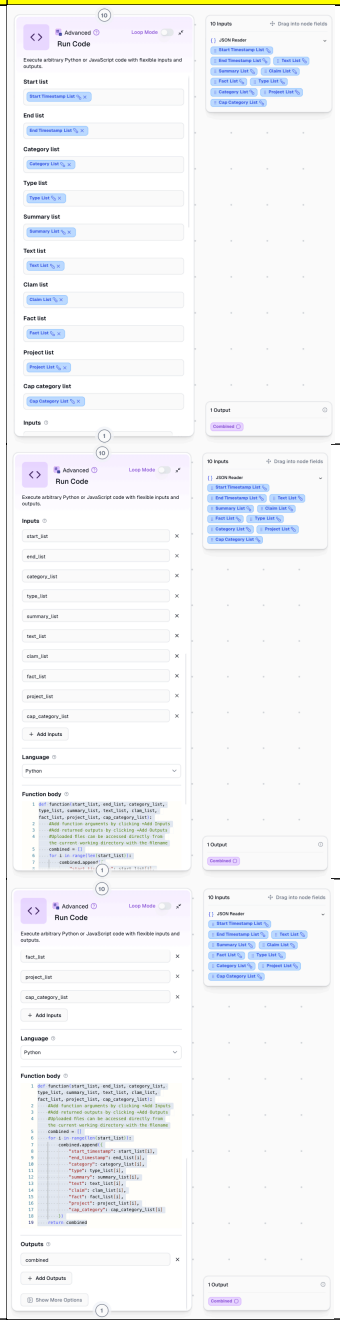
Order	Gumloop Node Name	Platform Integration	Loop Mode	Gumloop Node Image	Node Description	Input Fields & Values	Additional Metadata
1	Get Youtube Transcript	YouTube	Off		Get the transcript of a YouTube Video	**Video link:** https://www.youtube.com/shorts/i2yk_yFafRY	Input is a static YouTube Shorts link; output is Transcript List; expects list output; first node in workflow sequence

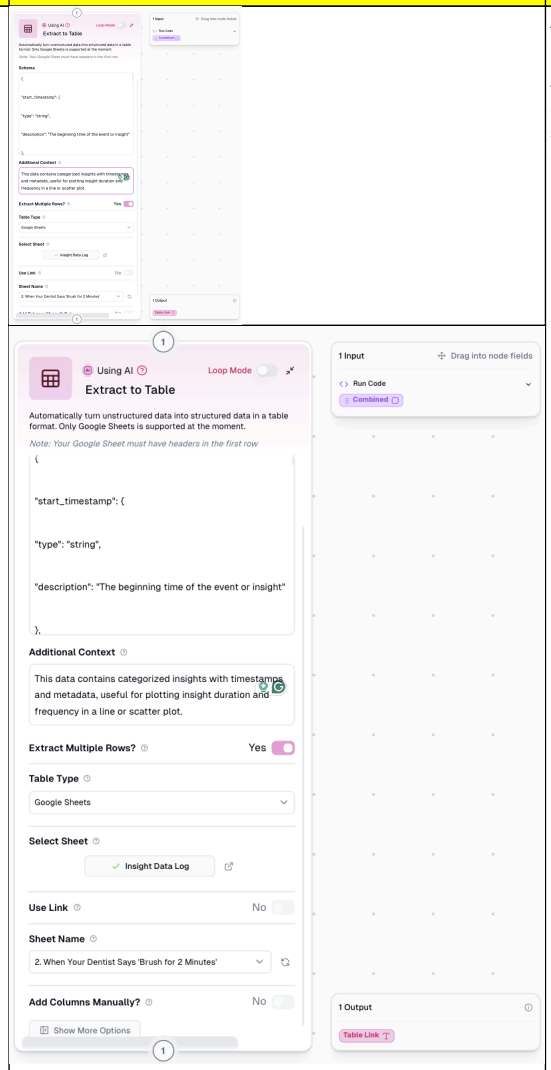
Order	Gumloop Node Name	Platform Integration	Loop Mode	Gumloop Node Image	Node Description	Input Fields & Values	Additional Metadata

2	Ask AI	Using AI	On		Prompt an AI language model (includes deep research). Provide all relevant context and use detailed prompts to get the best results.	**Prompt:**
						Here is the transcript input data from the video:
						{transcript__NODE_ID__:im4MuBv9jgruhvsfvb7WSg}
						You are an NLP fact-checking and insight extraction engine working on a project called **"Stop the Cap."**
						Your task is to extract high-value sentences or excerpts that reflect *claims, assumptions, advice, tactics, beliefs, expert analysis, or emotionally weighted stories*. These must match one or more **Y-Axis Categories** from the following framework:
						--- 	
						🧭 **Y-Axis Categories + Insight Types to Use in `type` field:**
						1. **Cap** – Bold claim, falsehood, misleading statement
						2. **Fact** – Truth, evidence, counterpoint
						3. **Reframe** – Perspective shift or moral/psych framing
						4. **Strategic Insight** – Logical reasoning, tactics
						5. **Contextual Depth** – Expert or high-authority opinion
						6. **New Info** – Brand-new or surprising factual detail
						7. **Impactful Example** – Anecdote or stat with emotional or social weight
						8. **First-Time Revelation** – Rare or previously unknown insight
						9. **Personal Wisdom** – Life lesson, belief, or vulnerable moment
						10. **Other** – Doesn't fit above but worth extracting
						--- 	
						🧠 **Instructions:**
						1. Read the full transcript and break it into sentence- or excerpt-level segments.
						2. Extract only segments that qualify under the above categories (even if the claim is implied or emotional).
						3. For each, extract the following fields:
						- `start_timestamp`: in ISO (e.g., `00:01:03`)
						- `end_timestamp`
						- `text`: direct quote from transcript
						- `summary`: short paraphrase or insight summary
						- `claim`: core claim or idea being expressed
						- `fact`: verified truth, supporting evidence, or clarification (use `"Verified True"` if no correction is needed)
						- `type`: use **exactly one label** from the 10 Insight Types above
						- `category`: domain or topic (e.g., Health, Culture, Finance)
						- `project`: always `"Stop the Cap"`
						- `cap_category`: only if `type` is `"Cap"` or `"Partial Cap"` — choose from: `Lie`, `Exaggeration`, `Contradiction`, `Outdated`, `Missing Context`, `Debatable`
						--- 	
						Return the results as a JSON list of objects like this:
						[
						{
						"start_timestamp": "00:00:03",
						"end_timestamp": "00:00:06",
						"text": "Dentists say brush for 2 minutes a day.",
						"summary": "Standard brushing guideline claim.",
						"claim": "Brush for 2 minutes twice a day.",
						"fact": "Verified True",
						"type": "Truth",
						"category": "Health",
						"project": "Stop the Cap",
						"cap_category": ""
						},
						{
						"start_timestamp": "00:00:08",
						"end_timestamp": "00:00:12",
						"text": "Flossing is actually optional.",
						"summary": "Claim that flossing is not necessary.",
						"claim": "Flossing is optional.",
						"fact": "The ADA and CDC both recommend flossing daily. Saying it's optional is misleading.",
						"type": "Cap",
						"category": "Health",
						"project": "Stop the Cap",
						"cap_category": "Misleading"
						}
						]
						Do not include the video title or any explanation. Do not wrap it in strings or an outer object. Only return the array of JSON objects.
							Input connected to Get Youtube Transcript (Transcript List); expects list output; output is Response List; AI processing node for extracting insights based on custom NLP prompt

Order	Gumloop Node Name	Platform Integration	Loop Mode	Gumloop Node Image	Node Description	Input Fields & Values	Additional Metadata
3	JSON Insights Extractor 2	Your Custom Nodes	Off		Extract and clean insights from JSON with timestamp conversion	**JSON Data:** Response List	Input connected to Ask AI (Response List); expects list input; output is Extracted Insights List of List; custom node designed for structured JSON
4	JSON Wrapper	Your Custom Nodes	Off		Wrap list in JSON object	**Extracted Insights List:** Extracted Insights List of List	Input connected to JSON Insights Extractor 2; expects List of Lists; Output: Wrapped Insights List
5	JSON Wrapper to Parse	Your Custom Nodes	Off		Parse JSON strings into Python dictionaries	**JSON Input:** Wrapped Insights List	Input connected to JSON Wrapper; expects List; Output: Parsed Data List of List
6	Flatten List of Lists	List Operations	Off		Flatten list of lists into a single list. Example: if input is [[1, 2], [3, 4], [5]] → output will be [1, 2, 3, 4, 5].	**Input list:** Parsed Data List of List	Input connected to JSON Wrapper to Parse; expects List of Lists; Output: List List; used for consolidating nested list structures into a
7	JSON Reader	JSON	On		Read values from a JSON object based on a provided key or keys; supports dot notation for nested properties.	**JSON String:** List List **Keys:** start_timestamp, end_timestamp, text, summary, claim, fact, type, category, project, cap_category	Input connected to Flatten List of Lists; expects list input; outputs 10 separate lists: Start Timestamp List, End Timestamp List, Text List, Summary List, Claim List, Fact List, Type List, Category List, Project List, Cap Category List

Order	Gumloop Node Name	Platform Integration	Loop Mode	Gumloop Node Image	Node Description	Input Fields & Values	Additional Metadata

8	Run Code	Advanced	Off		Execute Python or JavaScript with flexible inputs/outputs. Used here to combine the 10 parallel lists into a single list of records.	**Inputs:**
						start_list ← Start Timestamp List; end_list ← End Timestamp List; category_list ← Category List; type_list ← Type List; summary_list ← Summary List; text_list ← Text List; clam_list (claim) ← Claim List; fact_list ← Fact List; project_list ← Project List; cap_category_list ← Cap Category List. Language: Python.
						
						Function Body:
						def function(start_list, end_list, category_list, type_list, summary_list, text_list, clam_list, fact_list, project_list, cap_category_list):
						#Add function arguments by clicking +Add Inputs
						#Add returned outputs by clicking +Add Outputs
						#Uploaded files can be accessed directly from the current working directory with the filename
						combined = []
						for i in range(len(start_list)):
						combined.append({
						"start_timestamp": start_list[i],
						"end_timestamp": end_list[i],
						"category": category_list[i],
						"type": type_list[i],
						"summary": summary_list[i],
						"text": text_list[i],
						"claim": clam_list[i],
						"fact": fact_list[i],
						"project": project_list[i],
						"cap_category": cap_category_list[i]
						})
						return combined
							Input source: outputs from JSON Reader (10 lists). Output: Combined (list of objects). Notes: aligns items by index; watch the input name “clam_list” (spelling) which represents “claim.”

Order	Gumloop Node Name	Platform Integration	Loop Mode	Gumloop Node Image	Node Description	Input Fields & Values	Additional Metadata
9	Extract to Table	Using AI	Off		Automatically turns unstructured data into a structured table format. Currently only supports Google Sheets.	**Text:** Combined (output from Run Code) **Schema:** (full JSON schema provided) for start_timestamp, end_timestamp, category, type, summary, text, claim, fact, project, cap_category **Additional Context:** “This data contains categorized insights with timestamps and metadata, useful for plotting insight duration and frequency in a line or scatter plot.” **Extract Multiple Rows?** Yes **Table Type:** Google Sheets **Select Sheet:** Insight Data Log **Sheet Name:** Existing sheet name selected from document	Input connected to Run Code node. Output: Table Link. Requires Google Sheet to have headers in the first row. Used to persist AI-processed insights into a live spreadsheet for visualization/analysis.

						Schema Input:
						{
						"start_timestamp": {
						"type": "string",
						"description": "The beginning time of the event or insight"
						},
						"end_timestamp": {
						"type": "string",
						"description": "The ending time of the event or insight"
						},
						"category": {
						"type": "string",
						"description": "The general category or domain the insight belongs to"
						},
						"type": {
						"type": "string",
						"description": "The specific type of the event within its category"
						},
						"summary": {
						"type": "string",
						"description": "A brief overview of the event or topic"
						},
						"text": {
						"type": "string",
						"description": "The full insight text"
						},
						"claim": {
						"type": "string",
						"description": "The primary claim made in the insight"
						},
						"fact": {
						"type": "string",
						"description": "A supporting fact or reference"
						},
						"project": {
						"type": "string",
						"description": "The associated project or topic name"
						},
						"cap_category": {
						"type": "string",
						"description": "A capitalized category name for visualization or filtering"
						}
						}	