**OpenAI x NxtWave Buildathon | TS**

**AI-POWERED CITIZEN COMPLAINT MANAGEMENT SYSTEM**

**By**

**RAJARAM LAGISHETTI**

**SANDEEP RATHOD BANOTHU**

**DECLARATION**

We hereby declare that the work which is being presented in this Project entitled **“AI-POWERED CITIZEN COMPLAINT MANAGEMENT SYSTEM”** submitted to OpenAI x NxtWave Buildathon | TS.

Place: SECUNDERABAD

Date:

Rajaram Lagishetti

Sandeep Rathod Banothu

**1. PROBLEM STATEMENT**

Urban residents are confronted with difficulties in complaining about problems related to civic issues such as potholes, power cuts, water, and sanitation.

Today, grievances tend to:

Fail the proper validation (absence of proper info such as absence of name, incorrect email address, or incorrect phone numbers).

Manually sorted, which causes jamming in routing grievances to the correct department.

Fail to report any action taken upon submission.

This brings about frustrations to citizens and inefficiency to local authorities. The worst affected are everyday citizens who require immediate response to their local issues but are left to contend with bureaucracy and miscommunication.

**2. OUR SOLUTION**

We offer an AI-oriented Citizen Complaint Management System built using n8n that is connected to Google Sheets, Gmail, and OpenAI.

Components:

* Data Validation: Validating name, email address, phone number, description and location.
* Unique Complaint ID Generation: Generates a unique ID for each complaint as a way of tracking.
* AI Categorization: OpenAI will classify complaints (Road, Water, Power, Sanitation, etc.) as well as Assign Priority, and Urgency and auto-route to the correct department.
* Automated Updates:Citizens will get an acknowledgement email when their complaint is entered.
  + If it fails validation the citizen receives an error message so that they have confirmation of what to correct and re-submit the complaint.

Once the complaint is Assigned, there is a status update in Google Sheets, and the citizen is notified by email.

* Transparency & Effectiveness: Citizens are informed of where their complaint is at in the system and departments are receiving Properly Categorized Validated complaints.

How it is different:

* End-To-End low-code Automation in N8N.
* AI Assignment eliminates manual triage.
* Real time updates to citizens providing them with acknowledgement and error information.

**3. USE OF OPENAI APIS**

GPT-4.1-mini (Chat Completions API): For example, it takes in a description of the complaint.

It returns a structured JSON output: { category, priority, urgency\_keywords, assigned\_department }.

Example: "Complaint: Street light not working for 3 days" results in AI classifying the complaint as { "category": "Power", "priority": "High", "assigned\_department": "Electricity Department" }.

Future Improvements (Subject to Change):

* Whisper API to automatically transcribe voice complaints.
* DALL·E to automatically create infographic dashboards for reports.

Integration:

The n8n HTTP Request Node sends a call to the OpenAI API with the complaint text. Then, a Code Node processes the AI response, assigns departments, and sets the complaint status.

**4. FEASIBILITY**

Tech Stack & Tools:

1. n8n (workflow automation platform).
2. Google Sheets (complaint storage + status tracking).
3. Gmail API (citizen acknowledgment + error notifications).
4. OpenAI API (GPT-4.1-mini) for classification.
5. JavaScript (in Code Nodes) for validation and parsing AI responses.

Architecture:

* Trigger: Google Sheets → new complaint row.
* Validation: Code Node validates citizen info (name, email, phone, description, location).
  + If Node:
    - If Passed → AI Categorization + update sheet + acknowledgment email.
    - If Failed → error email with missing fields.
* AI Categorizer: OpenAI API → structured JSON output.
* Update: Google Sheets row updated with Category, Department, Priority, Status.
* Email: Citizen notified about complaint acknowledgment or failure.

Constraints:

Needs a stable OpenAI API key and Gmail API credentials.

Works best for text-based complaints (voice/image handling could be a future upgrade).

Requires good categorization prompts to avoid wrong AI classifications.

**5.SOURCE CODE**

{

"name": "AI Citizen Complaint Management\_Project",

"nodes": [

{

"parameters": {

"conditions": {

"options": {

"caseSensitive": true,

"leftValue": "",

"typeValidation": "strict",

"version": 1

},

"conditions": [

{

"id": "8fb5aa18-71cb-4bd9-93eb-886d2a56f2b3",

"leftValue": "={{ $json['Validation Status(Auto Generated)'] }}",

"rightValue": "Passed",

"operator": {

"type": "string",

"operation": "equals",

"name": "filter.operator.equals"

}

}

],

"combinator": "and"

},

"options": {}

},

"id": "cc0137d6-d44a-4875-a3f4-0e303c8d953f",

"name": "If Validation Passed",

"type": "n8n-nodes-base.if",

"typeVersion": 2,

"position": [

-672,

48

]

},

{

"parameters": {

"sendTo": "={{ $('Validate Data').item.json.Email }}",

"subject": "=Complaint Received - ID {{ $json['Complaint ID(Auto Generated)'] }}",

"message": "=Hello {{ $('Validate Data').item.json['Citizen Name'] }}, \nThank you for submitting your complaint regarding {{ $json['Category(Auto Generated)'] }}. Your complaint ID is {{ $json['Complaint ID(Auto Generated)'] }}. We have received your request and it is currently {{ $json.Status }}. Our team will review and take appropriate action. You will be updated once your complaint is processed.",

"options": {}

},

"id": "395f7f90-0253-431d-a4fb-b6b7f8944b3a",

"name": "Citizen Acknowledgment",

"type": "n8n-nodes-base.gmail",

"typeVersion": 2,

"position": [

704,

-144

],

"webhookId": "a2248ab6-72e2-4f64-828f-2927418530d6",

"credentials": {

"gmailOAuth2": {

"id": "xhyADyAxJOj0ZOCH",

"name": "Gmail account"

}

}

},

{

"parameters": {

"pollTimes": {

"item": [

{

"mode": "everyMinute"

}

]

},

"documentId": {

"\_\_rl": true,

"value": "12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI",

"mode": "list",

"cachedResultName": "Report",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit?usp=drivesdk"

},

"sheetName": {

"\_\_rl": true,

"value": 2104398066,

"mode": "list",

"cachedResultName": "Form responses 1",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit#gid=2104398066"

},

"event": "rowAdded",

"options": {}

},

"type": "n8n-nodes-base.googleSheetsTrigger",

"typeVersion": 1,

"position": [

-1168,

48

],

"id": "cdc86d0e-695b-4d46-b38a-4da3542a9392",

"name": "Google Sheets Trigger1",

"credentials": {

"googleSheetsTriggerOAuth2Api": {

"id": "qd7cU6EXMOFDGEf3",

"name": "Google Sheets Trigger account"

}

}

},

{

"parameters": {

"jsCode": "const items = $input.all();\n\n// Function to generate unique Complaint ID\nfunction generateComplaintID() {\n return 'CID-' + Date.now(); // timestamp-based unique ID\n}\n\n// Email validation\nfunction validateEmail(email) {\n const regex = /^[\\w-]+(\\.[\\w-]+)\*@([\\w-]+\\.)+[a-zA-Z]{2,7}$/;\n return regex.test(email);\n}\n\n// Phone validation\nfunction validatePhone(phone) {\n const regex = /^[0-9+\\-\\s()]{10,15}$/;\n return regex.test(phone);\n}\n\nreturn items.map(item => {\n const data = item.json;\n\n // Always generate Complaint ID\n data['Complaint ID(Auto Generated)'] = generateComplaintID();\n\n let errorMessages = [];\n let missingFields = [];\n\n // Validation checks\n if (!data['Citizen Name'] || data['Citizen Name'].trim() === '') {\n errorMessages.push('Missing Citizen Name');\n missingFields.push('Citizen Name');\n }\n if (!data['Email'] || !validateEmail(data['Email'])) {\n errorMessages.push('Invalid or Missing Email');\n missingFields.push('Email');\n }\n if (!data['Phone'] || !validatePhone(data['Phone'])) {\n errorMessages.push('Invalid or Missing Phone');\n missingFields.push('Phone');\n }\n if (!data['Complaint Description'] || data['Complaint Description'].trim() === '') {\n errorMessages.push('Missing Complaint Description');\n missingFields.push('Complaint Description');\n }\n if (!data['Full Location(Vi, M, Dis, PIN)'] || data['Full Location(Vi, M, Dis, PIN)'].trim() === '') {\n errorMessages.push('Missing Full Location');\n missingFields.push('Full Location');\n }\n\n // Set validation status\n const validationStatus = errorMessages.length > 0 ? 'Failed' : 'Passed';\n\n // Set notes\n const notes = validationStatus === 'Passed' \n ? 'Complaint data validated successfully'\n : 'Please correct the following fields: ' + missingFields.join(', ');\n\n return {\n json: {\n ...data,\n 'Validation Status(Auto Generated)': validationStatus,\n 'Error Message(Auto Generated)': errorMessages.join('; '),\n 'Notes(Auto Generated)': notes,\n 'Status': 'Pending',\n 'Category(Auto Generated)': '',\n 'Priority(Auto Generated)': '',\n 'Assigned Department(Auto Generated)': '',\n 'Processing Date(Auto Generated)': ''\n }\n };\n});\n"

},

"type": "n8n-nodes-base.code",

"typeVersion": 2,

"position": [

-944,

48

],

"id": "4bc8be0c-2a9a-4df0-bee7-fc91168d8f2a",

"name": "Validate Data"

},

{

"parameters": {

"sendTo": "={{ $('Validate Data').item.json.Email }}",

"subject": "Complaint Submission Failed",

"message": "=❌ Validation Failed for Complaint ID: {{ $json['Complaint ID(Auto Generated)'] }}\nCitizen Name: {{ $json['Citizen Name '] }}\nEmail: {{ $json.Email }} \n\nErrors: {{ $json[\"Error Message(Auto Generated)\"] }}\nWarnings: {{ $json[\"Notes(Auto Generated)\"] || \"None\" }}\nProcessing Status: {{ $json[\"Processing Status\"] || \"Failed\" }} Retry Count: {{ $json[\"Retry Count\"] || 0 }}\n\n⚠ Action Required: Please review the complaint data and correct the errors before reprocessing.",

"options": {}

},

"id": "c37db75a-1a29-4a38-aec0-cd3f0037dc74",

"name": "Handle Validation Fail",

"type": "n8n-nodes-base.gmail",

"typeVersion": 2,

"position": [

688,

48

],

"webhookId": "9aa4c61f-d7d5-414c-987c-231fc90fa95d",

"credentials": {

"gmailOAuth2": {

"id": "xhyADyAxJOj0ZOCH",

"name": "Gmail account"

}

}

},

{

"parameters": {},

"type": "n8n-nodes-base.merge",

"typeVersion": 3.2,

"position": [

-352,

176

],

"id": "50f33542-2f75-41d1-a23f-57e44517a766",

"name": "Merge"

},

{

"parameters": {

"method": "POST",

"url": "https://api.openai.com/v1/chat/completions",

"authentication": "genericCredentialType",

"genericAuthType": "httpBearerAuth",

"sendHeaders": true,

"headerParameters": {

"parameters": [

{

"name": "Authorization",

"value": "={{$vars.OpenAI\_API\_KEY}}"

},

{

"name": "Content-Type",

"value": "application/json"

}

]

},

"sendBody": true,

"specifyBody": "json",

"jsonBody": "={\n \"model\": \"gpt-4.1-mini\",\n \"messages\": [\n {\n \"role\": \"system\",\n \"content\": \"You are an AI assistant that classifies municipal complaints into structured JSON with {category, priority, urgency\_keywords}. Return only JSON.\"\n },\n {\n \"role\": \"user\",\n \"content\": \"Complaint: {{ $json['Complaint Description'] }}\"\n }\n ]\n}\n",

"options": {}

},

"type": "n8n-nodes-base.httpRequest",

"typeVersion": 4.2,

"position": [

-352,

-48

],

"id": "3f61c5a2-8abf-44c0-a758-a800e20a8bcb",

"name": "Open AI Categorizer",

"credentials": {

"httpBearerAuth": {

"id": "bRDQBIPEJVmxND8X",

"name": "Bearer Auth account"

}

}

},

{

"parameters": {

"jsCode": "const items = $input.all();\n\nreturn items.map(item => {\n let aiResponseStr = item.json.choices?.[0]?.message?.content || '{}';\n let aiData = {};\n let errorMessage = '';\n\n // Remove code block markers if present\n aiResponseStr = aiResponseStr.trim()\n .replace(/^```json\\s\*/i, '') // Remove ```json at start\n .replace(/^```/, '') // Remove ``` at start if any\n .replace(/```$/, ''); // Remove ``` at end if any\n\n // Try to parse AI response JSON safely\n try {\n aiData = JSON.parse(aiResponseStr);\n } catch (e) {\n errorMessage = 'Failed to parse AI response';\n aiData = {};\n }\n\n // Extract fields or set defaults\n const category = aiData.category || '';\n const priority = aiData.priority || '';\n const urgencyKeywords = aiData.urgency\_keywords ? aiData.urgency\_keywords.join(', ') : ''; // Combine into string\n let assignedDepartment = aiData.assigned\_department || '';\n\n // Auto-assign department based on category if not provided\n if (!assignedDepartment) {\n switch(category.toLowerCase()) {\n case 'road maintenance':\n assignedDepartment = 'Public Works';\n break;\n case 'power':\n assignedDepartment = 'Electricity Department';\n break;\n case 'water supply':\n assignedDepartment = 'Water Department';\n break;\n case 'sanitation':\n assignedDepartment = 'Sanitation Department';\n break;\n // Add more rules as needed\n default:\n assignedDepartment = ''; // remain empty if no rule matches\n }\n }\n\n // Determine status\n const status = assignedDepartment ? 'Assigned' : 'Pending';\n\n // Add simple error message if all fields are missing\n if (!category && !priority && !assignedDepartment && !urgencyKeywords && !errorMessage) {\n errorMessage = 'AI response is missing expected fields';\n }\n\n const processingDate = new Date().toISOString();\n\n return {\n json: {\n 'Category(Auto Generated)': category,\n 'Priority(Auto Generated)': priority,\n 'Assigned Department(Auto Generated)': assignedDepartment,\n 'Urgency Keywords(Auto Generated)': urgencyKeywords,\n 'Processing Date(Auto Generated)': processingDate,\n 'Status': status,\n 'Error Message(Auto Generated)': errorMessage\n }\n };\n});\n"

},

"type": "n8n-nodes-base.code",

"typeVersion": 2,

"position": [

-128,

-48

],

"id": "f6415886-66b8-4efc-adf3-48cf45382831",

"name": "Parse AI Response"

},

{

"parameters": {

"operation": "update",

"documentId": {

"\_\_rl": true,

"value": "12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI",

"mode": "list",

"cachedResultName": "Report",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit?usp=drivesdk"

},

"sheetName": {

"\_\_rl": true,

"value": 2104398066,

"mode": "list",

"cachedResultName": "Form responses 1",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit#gid=2104398066"

},

"columns": {

"mappingMode": "defineBelow",

"value": {

"Category(Auto Generated)": "={{ $json['Category(Auto Generated)'] }}",

"Priority(Auto Generated)": "={{ $json['Priority(Auto Generated)'] }}",

"Complaint ID(Auto Generated)": "={{ $('Validate Data').item.json['Complaint ID(Auto Generated)'] }}",

"Assigned Department(Auto Generated)": "={{ $json['Assigned Department(Auto Generated)'] }}",

"Urgency Keywords(Auto Generated)": "={{ $json['Urgency Keywords(Auto Generated)'] }}",

"Processing Date(Auto Generated)": "={{ $json['Processing Date(Auto Generated)'] }}",

"Validation Status(Auto Generated)": "={{ $('Validate Data').item.json['Validation Status(Auto Generated)'] }}",

"Error Message(Auto Generated)": "=",

"Notes(Auto Generated)": "={{ $('Validate Data').item.json['Notes(Auto Generated)'] }}",

"Timestamp": "={{ $('Validate Data').item.json.Timestamp }}",

"Status": "={{ $json.Status }}"

},

"matchingColumns": [

"Timestamp"

],

"schema": [

{

"id": "Timestamp",

"displayName": "Timestamp",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Citizen Name",

"displayName": "Citizen Name",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Email",

"displayName": "Email",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Phone",

"displayName": "Phone",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Full Location(Vi, M, Dis, PIN)",

"displayName": "Full Location(Vi, M, Dis, PIN)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Complaint Description",

"displayName": "Complaint Description",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Complaint ID(Auto Generated)",

"displayName": "Complaint ID(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Category(Auto Generated)",

"displayName": "Category(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Priority(Auto Generated)",

"displayName": "Priority(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Status",

"displayName": "Status",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Assigned Department(Auto Generated)",

"displayName": "Assigned Department(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Urgency Keywords(Auto Generated)",

"displayName": "Urgency Keywords(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Processing Date(Auto Generated)",

"displayName": "Processing Date(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Validation Status(Auto Generated)",

"displayName": "Validation Status(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Error Message(Auto Generated)",

"displayName": "Error Message(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Notes(Auto Generated)",

"displayName": "Notes(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "row\_number",

"displayName": "row\_number",

"required": false,

"defaultMatch": false,

"display": true,

"type": "number",

"canBeUsedToMatch": true,

"readOnly": true,

"removed": true

}

],

"attemptToConvertTypes": false,

"convertFieldsToString": false

},

"options": {}

},

"type": "n8n-nodes-base.googleSheets",

"typeVersion": 4.7,

"position": [

416,

-144

],

"id": "70ae5160-0dbe-48a6-abca-236503d82507",

"name": "Update row in sheet",

"credentials": {

"googleSheetsOAuth2Api": {

"id": "QbYU1ngLce8wr4vg",

"name": "Google Sheets account"

}

}

},

{

"parameters": {

"conditions": {

"options": {

"caseSensitive": true,

"leftValue": "",

"typeValidation": "strict",

"version": 2

},

"conditions": [

{

"id": "2acbe76f-4a3a-4614-9b18-3c972c954e83",

"leftValue": "={{ $json.Status }}",

"rightValue": "Assigned",

"operator": {

"type": "string",

"operation": "equals",

"name": "filter.operator.equals"

}

}

],

"combinator": "and"

},

"options": {}

},

"type": "n8n-nodes-base.if",

"typeVersion": 2.2,

"position": [

112,

-48

],

"id": "dd19b0fc-c4db-4115-82e1-4d659d638830",

"name": "If"

},

{

"parameters": {

"operation": "update",

"documentId": {

"\_\_rl": true,

"value": "12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI",

"mode": "list",

"cachedResultName": "Report",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit?usp=drivesdk"

},

"sheetName": {

"\_\_rl": true,

"value": 2104398066,

"mode": "list",

"cachedResultName": "Form responses 1",

"cachedResultUrl": "https://docs.google.com/spreadsheets/d/12kUmXWwd\_qov01A1R6eWEvoVXEpk2M7dTtreKq-D0mI/edit#gid=2104398066"

},

"columns": {

"mappingMode": "defineBelow",

"value": {

"Category(Auto Generated)": "=",

"Priority(Auto Generated)": "=",

"Status (Auto Generated)": "={{ $('Validate Data').item.json.Status }}",

"Complaint ID(Auto Generated)": "={{ $('Validate Data').item.json['Complaint ID(Auto Generated)'] }}",

"Assigned Department(Auto Generated)": "=",

"Urgency Keywords(Auto Generated)": "=",

"Processing Date(Auto Generated)": "=",

"Citizen Name ": "=",

"Validation Status(Auto Generated)": "={{ $json['Validation Status(Auto Generated)'] }}",

"Error Message(Auto Generated)": "={{ $json['Error Message(Auto Generated)'] }}",

"Notes(Auto Generated)": "={{ $json['Notes(Auto Generated)'] }}",

"Timestamp": "={{ $('Validate Data').item.json.Timestamp }}"

},

"matchingColumns": [

"Timestamp"

],

"schema": [

{

"id": "Timestamp",

"displayName": "Timestamp",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Citizen Name ",

"displayName": "Citizen Name ",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Email",

"displayName": "Email",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Phone",

"displayName": "Phone",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Full Location(Vi, M, Dis, State, PIN)",

"displayName": "Full Location(Vi, M, Dis, State, PIN)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Complaint Description",

"displayName": "Complaint Description",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Complaint ID(Auto Generated)",

"displayName": "Complaint ID(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true,

"removed": false

},

{

"id": "Category(Auto Generated)",

"displayName": "Category(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Priority(Auto Generated)",

"displayName": "Priority(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Status (Auto Generated)",

"displayName": "Status (Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Assigned Department(Auto Generated)",

"displayName": "Assigned Department(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Urgency Keywords(Auto Generated)",

"displayName": "Urgency Keywords(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Processing Date(Auto Generated)",

"displayName": "Processing Date(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Validation Status(Auto Generated)",

"displayName": "Validation Status(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Error Message(Auto Generated)",

"displayName": "Error Message(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "Notes(Auto Generated)",

"displayName": "Notes(Auto Generated)",

"required": false,

"defaultMatch": false,

"display": true,

"type": "string",

"canBeUsedToMatch": true

},

{

"id": "row\_number",

"displayName": "row\_number",

"required": false,

"defaultMatch": false,

"display": true,

"type": "number",

"canBeUsedToMatch": true,

"readOnly": true,

"removed": true

}

],

"attemptToConvertTypes": false,

"convertFieldsToString": false

},

"options": {}

},

"type": "n8n-nodes-base.googleSheets",

"typeVersion": 4.7,

"position": [

416,

48

],

"id": "b1b961d7-456e-41f4-b3be-731aaf83acee",

"name": "Update row in sheet1",

"credentials": {

"googleSheetsOAuth2Api": {

"id": "QbYU1ngLce8wr4vg",

"name": "Google Sheets account"

}

}

},

{

"parameters": {},

"type": "n8n-nodes-base.noOp",

"typeVersion": 1,

"position": [

912,

-144

],

"id": "471809fe-6602-46a6-a2a0-4340769e8fc4",

"name": "No Operation, do nothing"

},

{

"parameters": {},

"type": "n8n-nodes-base.noOp",

"typeVersion": 1,

"position": [

896,

48

],

"id": "b6f98f98-3e62-4b17-b022-26142321319a",

"name": "No Operation, do nothing1"

}

],

"pinData": {},

"connections": {

"If Validation Passed": {

"main": [

[

{

"node": "Open AI Categorizer",

"type": "main",

"index": 0

}

],

[

{

"node": "Merge",

"type": "main",

"index": 0

}

]

]

},

"Google Sheets Trigger1": {

"main": [

[

{

"node": "Validate Data",

"type": "main",

"index": 0

}

]

]

},

"Validate Data": {

"main": [

[

{

"node": "If Validation Passed",

"type": "main",

"index": 0

}

]

]

},

"Merge": {

"main": [

[

{

"node": "Update row in sheet1",

"type": "main",

"index": 0

}

]

]

},

"Open AI Categorizer": {

"main": [

[

{

"node": "Parse AI Response",

"type": "main",

"index": 0

}

]

]

},

"Parse AI Response": {

"main": [

[

{

"node": "If",

"type": "main",

"index": 0

}

]

]

},

"If": {

"main": [

[

{

"node": "Update row in sheet",

"type": "main",

"index": 0

}

],

[

{

"node": "Update row in sheet1",

"type": "main",

"index": 0

}

]

]

},

"Update row in sheet1": {

"main": [

[

{

"node": "Handle Validation Fail",

"type": "main",

"index": 0

}

]

]

},

"Update row in sheet": {

"main": [

[

{

"node": "Citizen Acknowledgment",

"type": "main",

"index": 0

}

]

]

},

"Citizen Acknowledgment": {

"main": [

[

{

"node": "No Operation, do nothing",

"type": "main",

"index": 0

}

]

]

},

"Handle Validation Fail": {

"main": [

[

{

"node": "No Operation, do nothing1",

"type": "main",

"index": 0

}

]

]

}

},

"active": false,

"settings": {

"executionOrder": "v1"

},

"versionId": "3677919d-471a-4e2e-a928-9d68c05a63dd",

"meta": {

"templateCredsSetupCompleted": true,

"instanceId": "83314bcad929b40985259bedcb332ba9efe6c182f18e5f65d3884f60d3a604fc"

},

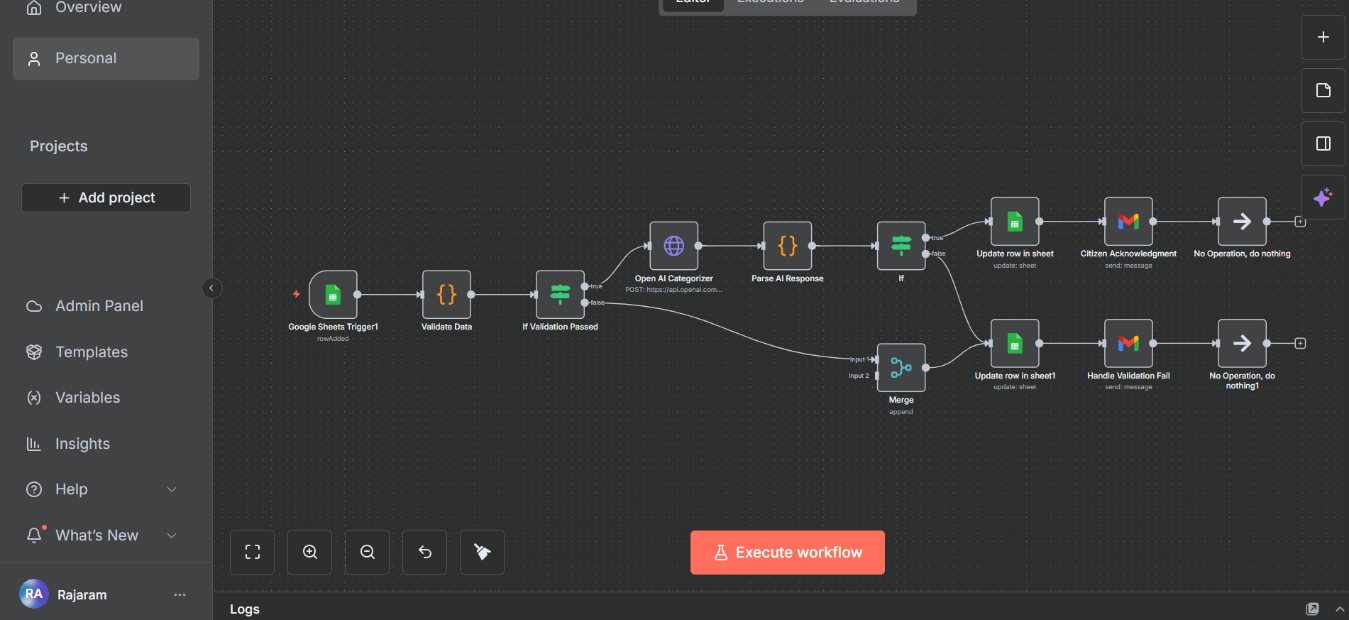
"id": "RjWJ3mXZayBJrcij",

"tags": []

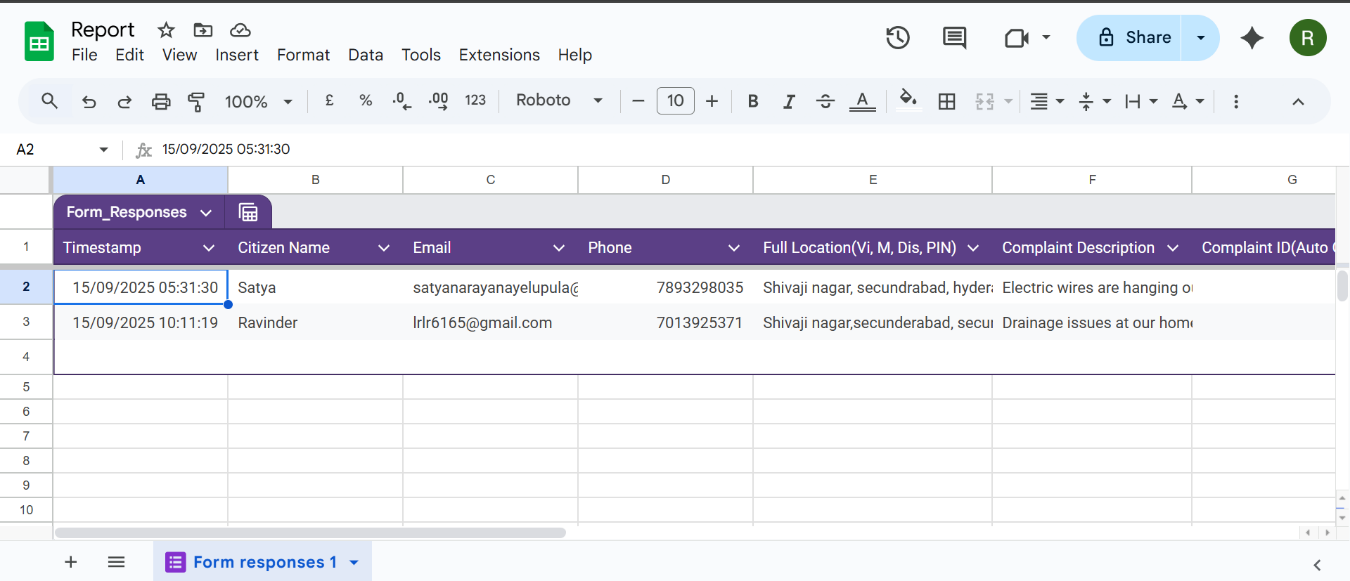
}

**OUTPUT SCREENS**

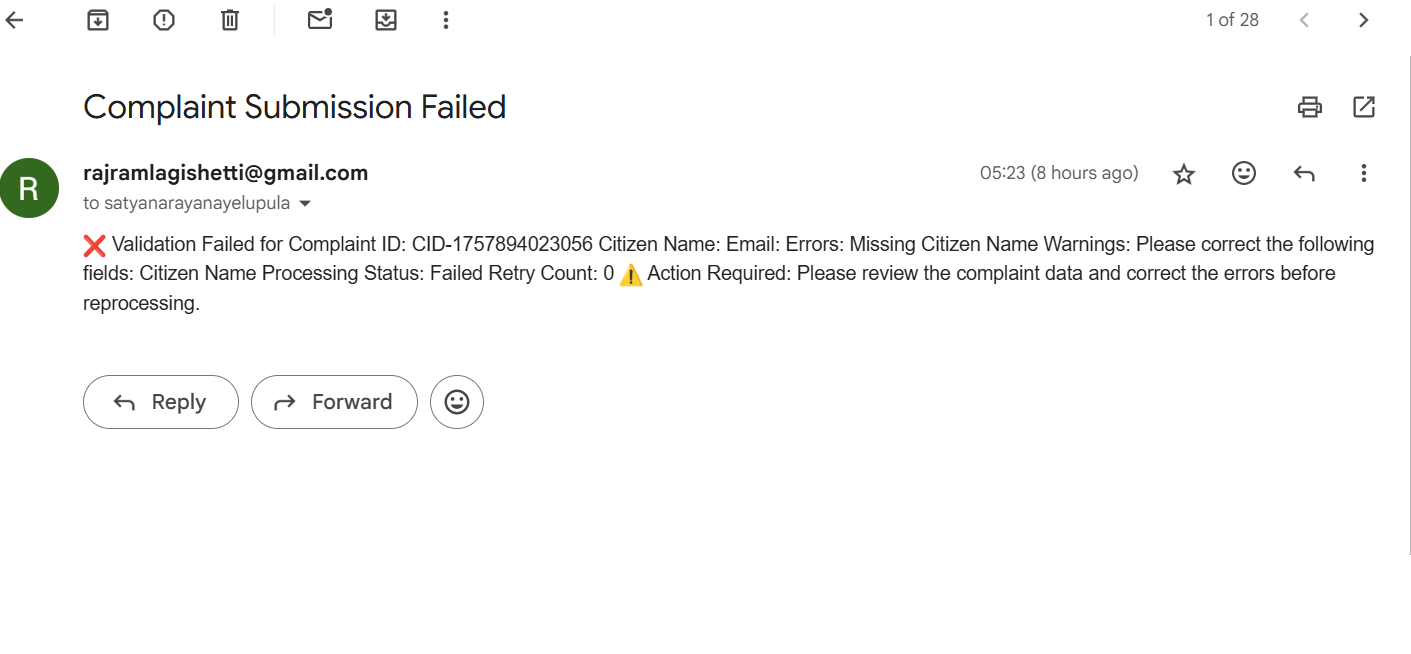
**n8n Workflow:**



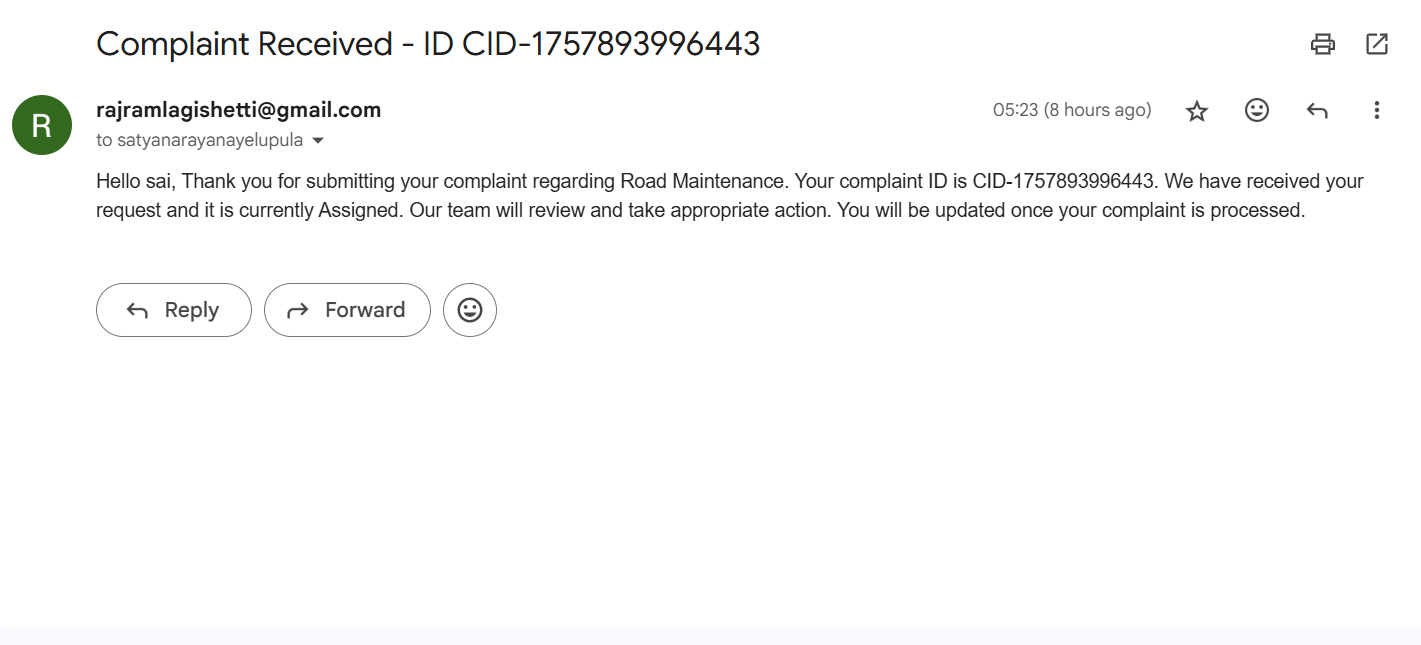
**Google Sheet -FORM**



**COMPLAINT FAILURE:**



**COMPLAINT RECEIVED:**



**REFERENCE**

n8n workflow link: