

All Links -

1. Document link -
<https://docs.google.com/document/d/1DMKvYlyFw1mtepglHX8Fz-ktLkMgp9rythsuux-a7ImY/edit?usp=sharing>
2. Claude link for agent 1 -
<https://claude.ai/public/artifacts/899a2dc1-d4e2-4864-aefe-3b883748f882>
3. Claude link for agent part 2 -
<https://claude.ai/public/artifacts/cdd1914d-8491-439f-aa32-cda8dd83efd3>
4. Google Sheet link -
<https://docs.google.com/spreadsheets/d/1bQ5HysCOLfhjTQqpQVabrvrEamWYyzswIQSNAhq4-g0/edit?usp=sharing>

Step 1 -Google sheet copy in my drive is ready

Step 2 - Google Sheet Auth API Credential Ready

Step 3 - Google Sheets Trigger API Credential Ready

Step 4 - In workflow Google Sheets Trigger —> On Row Added (add this node) - Input Sheet and every minute

Step 5 - Code node added to process input

Step 6 -Agent 1 created - Ideation Agent ready

Step 7- Ideation Agent Code node added running fine

Step 8 - Market Research Agent Added

Step 9 - Code output for Market Agent Done

Step 10 - Prioritisation Agent Done

Step 11 - Prioritisation code Done

Step 12 - Roadmap agent Done

Step 13 - Roadmap Code Done

Easy Copy Paste Links

Step 1-7 Claude publish -

<https://claude.ai/public/artifacts/899a2dc1-d4e2-4864-aefe-3b883748f882>

Step 8 onwards, Claude publish -

<https://claude.ai/public/artifacts/cdd1914d-8491-439f-aa32-cda8dd83efd3>

What is this about?

N8N Product Management Multi-Agent Workflow Guide

Overview

This workflow creates a multi-agent system for product management that handles ideation, roadmapping, and prioritization using Gemini 2.5 Pro Flash, with Google Sheets integration for input and output.

Prerequisites

- N8N instance (cloud or self-hosted)
- Gemini API key
- Google Sheets API credentials
- Google Drive access (for sheets integration)

Agent Architecture

The workflow consists of 5 specialised agents:

1. Ideation Agent - Generates and refines product ideas
2. Market Research Agent - Analyses market trends and competition
3. Prioritisation Agent - Scores and ranks features/ideas
4. Roadmap Agent - Creates timeline and dependencies
5. Output Agent - Formats and writes results to Google Sheets

Step 1 - Setup Google Sheets

1.1 Create Google Sheets Credential

1. Go to N8N → Credentials → Add Credential
2. Select "Google Sheets API"
3. Follow OAuth setup or use Service Account JSON
4. Test connection

You can see the sample sheet here - [AI Agent - Ideation and Product Priority](#)

1.2 Create Input/Output Sheets

Create two Google Sheets:

A1	idea_description	b	c	d	e	f
1	idea_description	business_context	target_audience	constraints	timeline_preference	
2	I want to start Uber for kids school drop in tier-1 cities of India. Growing market demand for kids to be dropped to schools in Metro cities in \$25000 budget and also constraint on high demand 3 months timeline					
3						
4						
5						

Input Sheet with columns:

- idea_description
- business_context
- target_audience
- constraints
- Timeline_preference

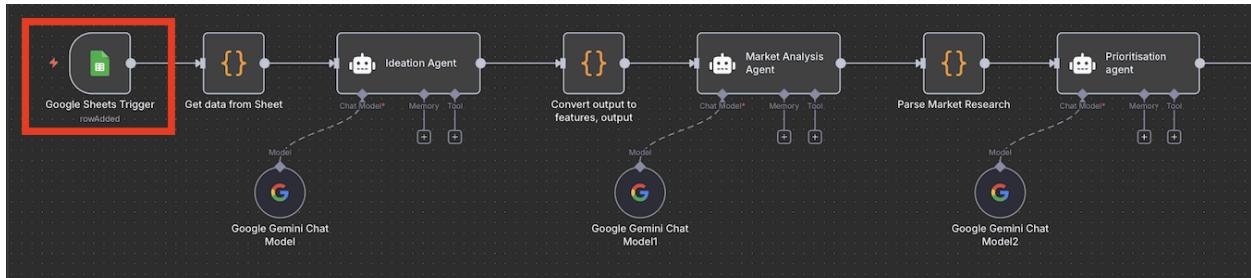
A1	a	b	c	d	e	f	g	h
1	idea_id	refined_idea	market_analysis	priority_score	roadmap_phase	success_metrics	effort_estimate	business_value

Output Sheet with columns:

- idea_id
- refined_idea
- market_analysis
- priority_score
- roadmap_phase
- dependencies
- effort_estimate
- business_value

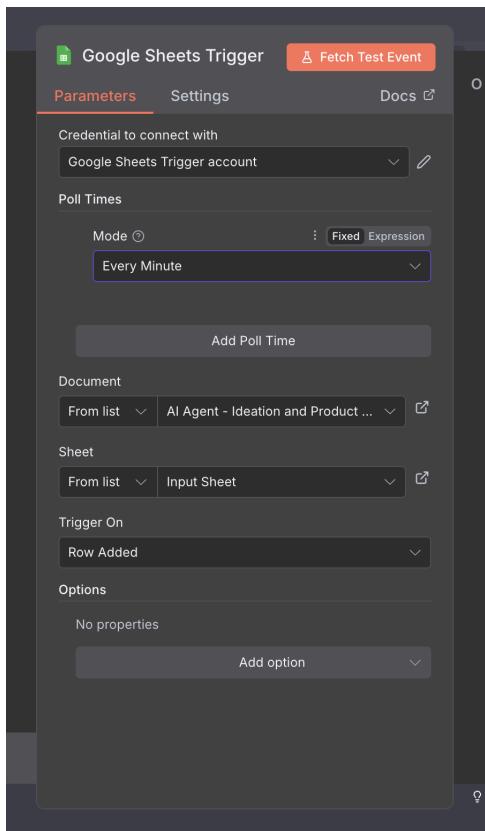
Step 2a - Sheets trigger node

Step 2: Create the Main Workflow



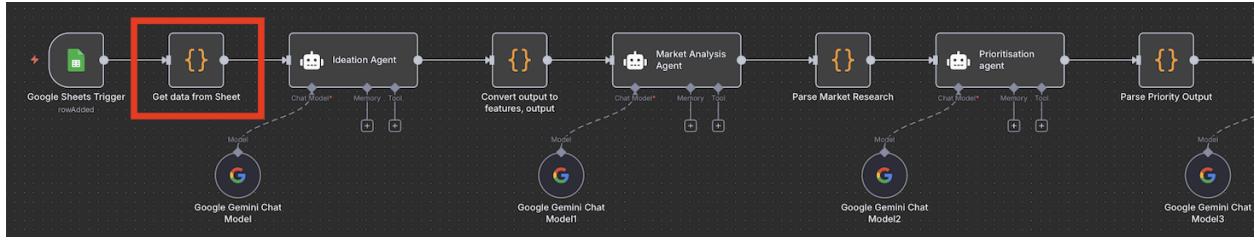
2.1 Workflow Trigger

1. Add "Google Sheets Trigger" node
2. Configure to monitor your input sheet
3. Set trigger on "Row Added" or use manual trigger for testing
4. Setup the below configuration
 - a. Document name - AI Agent - Ideation and Product Priority
 - b. From list - Input sheet
 - c. Trigger on - Row Added
 - d. Poll Times - Every Minute



Step 2b - Input Validation Node

2.2 Input Validation Node



1. Add "Code" node after trigger
2. Validate and clean input data:

// Validate required fields

```
const requiredFields = ['idea_description', 'business_context'];
```

```
const inputData = items[0].json;
```

```
for (const field of requiredFields) {
```

```
  if (!inputData[field] || inputData[field].trim() === "") {
```

```
    throw new Error(`Missing required field: ${field}`);
```

```
}
```

```
}
```

// Clean and structure data

```
const processedInput = {
```

```
  ideaDescription: inputData.idea_description.trim(),
```

```
  businessContext: inputData.business_context.trim(),
```

```
  targetAudience: inputData.target_audience || 'General users',
```

```
  constraints: inputData.constraints || 'No specific constraints',
```

```
  timelinePreference: inputData.timeline_preference || '3-6 months'
```

```
};
```

```
return [{ json: processedInput }];
```

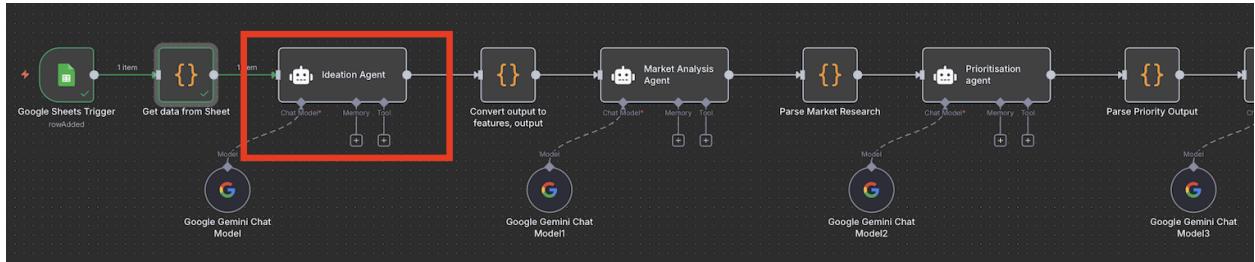
The screenshot shows a data processing workflow with the following components:

- INPUT:** A Google Sheets Trigger node with the following parameters:
 - id:** Parameter: "language" to start Uber for kids school drop in tier-1 cities of India
 - business_context:** Growing market demand for kids to be dropped to school will give us revenue
 - target_audience:** Metro cities in India
 - constraints:** \$25000 budget and also constraint on high demand during schools hours and no demand post those hours
 - timeline_preference:** 3 months timeline
- Process Node:** A "Get data from Sheet" node with the following configuration:
 - Mode:** Run Once for All Items
 - Language:** JavaScript
 - Code:**

```
5  for (const field of requiredFields) {  
6    if (!inputData[field] || inputData[field].trim() === '') {  
7      throw new Error(`Missing required field: ${field}`);  
8    }  
9  }  
10 // Clean and structure data  
11 const processedInput = {  
12   ideaDescription: inputData.idea_description.trim(),  
13   businessContext: inputData.business_context.trim(),  
14   targetAudience: inputData.target_audience || 'General users',  
15   constraints: inputData.constraints || 'No specific constraints',  
16   timelinePreference: inputData.timeline_preference || '3-6 months'  
17 };  
18  
20 return [{ json: processedInput }];
```
- OUTPUT:** A table showing the processed data with three columns:
 - ideaDescription:** I want to start Uber for kids school drop in tier-1 cities of India
 - businessContext:** Growing market demand for kids to be dropped to school will give us revenue
 - target_audience:** Metro cities in India

Step 3a - Ideation Agent

Step 3: Agent 1 - Ideation Agent



3.1 Add AI Agent Node

1. Add "AI Agent" node from the AI section
2. Configure Agent Settings:
 - Agent Type: Text Generator
 - Model: Select Google Gemini
 - Model Name: gemini-2.0-flash-exp
 - API Key: Use your Gemini API key credential
 - Temperature: 0.7
 - Max Tokens: 1000

3.2 Ideation Agent Prompt

Set the prompt field in the AI Agent node:

You are a Product Ideation Agent specializing in refining and expanding product concepts for successful market entry.

CONTEXT:

- Original Idea: {{\$json.ideaDescription}}
- Business Context: {{\$json.businessContext}}
- Target Audience: {{\$json.targetAudience}}
- Constraints: {{\$json.constraints}}
- Timeline: {{\$json.timelinePreference}}

YOUR EXPERTISE:

You excel at transforming rough ideas into clear, actionable product concepts. You understand market dynamics, user psychology, and product-market fit principles.

TASKS:

1. REFINE the original idea for maximum clarity, feasibility, and market appeal
2. GENERATE 2-3 strategic alternative variations that address the same core need
3. IDENTIFY the fundamental user problems and pain points being solved
4. SUGGEST 4-6 key features that deliver core value (prioritize MVP essentials)
5. ARTICULATE the unique value proposition that differentiates from competitors

6. ASSESS the idea's innovation potential and market disruption opportunity

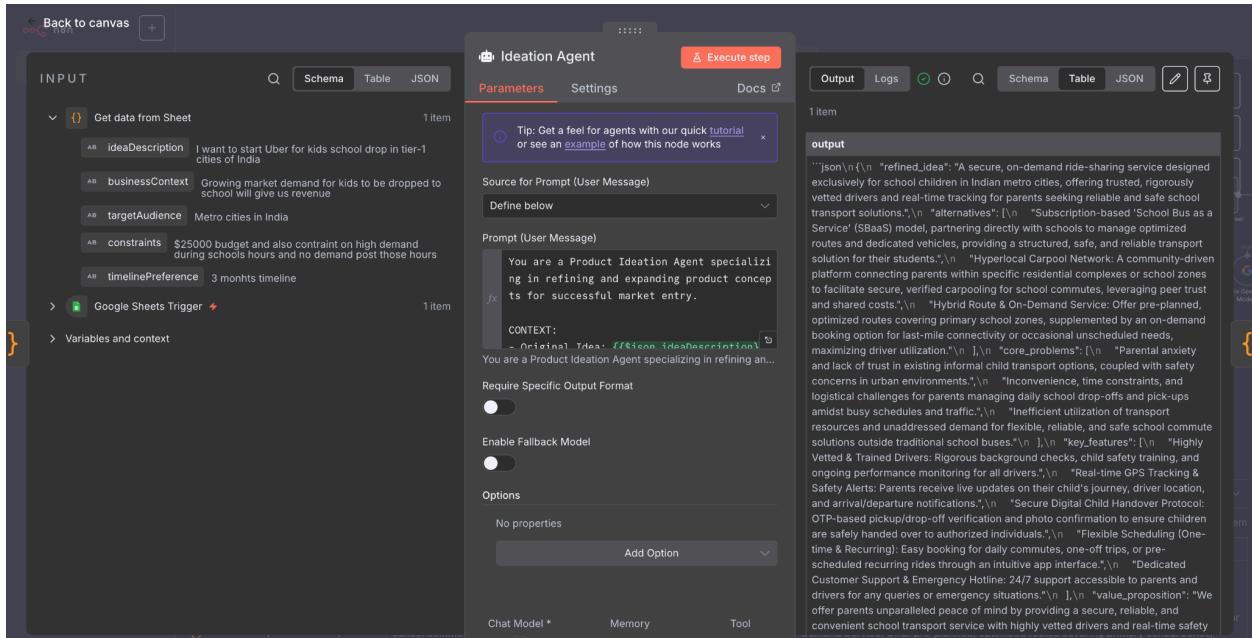
OUTPUT FORMAT (must be valid JSON):

{

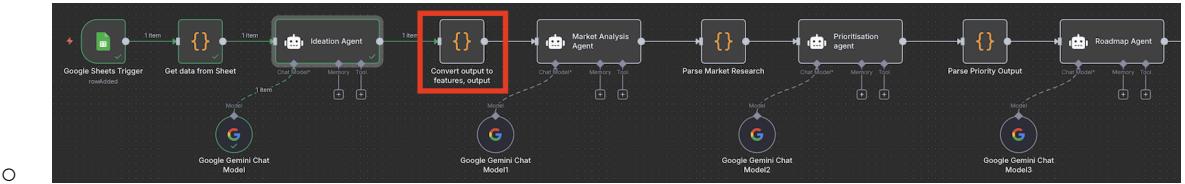
```
"refined_idea": "A clear, compelling product description in 2-3 sentences",
"alternatives": ["Alternative approach 1", "Alternative approach 2", "Alternative approach 3"],
"core_problems": ["Primary user pain point", "Secondary user pain point", "Business problem solved"],
"key_features": ["Essential feature 1", "Essential feature 2", "Essential feature 3", "Key differentiator feature"],
"value_proposition": "Single sentence explaining why users will choose this over alternatives",
"innovation_score": 8,
"feasibility_assessment": "Brief assessment of technical and business feasibility"
```

}

IMPORTANT: Respond ONLY with the JSON object. No additional text or explanations outside the JSON structure.



Step 3b - Ideation validation



3.3 Parse Ideation Response

1. Add "Code" node after the AI Agent
2. Parse and validate the JSON response:

```
// Get the AI response

const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;

try {

    // Try to parse as JSON directly

    let ideationResult;

    if (typeof aiResponse === 'string') {

        // Clean the response - remove markdown formatting if present

        const cleanResponse = aiResponse.replace(/\` `json\n?|\\n?\` `/g, "").trim();

        ideationResult = JSON.parse(cleanResponse);

    } else {

        ideationResult = aiResponse;

    }

    // Validate required fields

    const requiredFields = ['refined_idea', 'alternatives', 'core_problems', 'key_features',
    'value_proposition'];

    for (const field of requiredFields) {

        if (!ideationResult[field]) {
```

```
        throw new Error(`Missing required field: ${field}`);
    }
}

// Ensure arrays are properly formatted
if (!Array.isArray(ideationResult.alternatives)) ideationResult.alternatives = [];
if (!Array.isArray(ideationResult.core_problems)) ideationResult.core_problems = [];
if (!Array.isArray(ideationResult.key_features)) ideationResult.key_features = [];

return [
  json: {
    ...items[0].json,
    ideationResult: ideationResult
  }
];
}

} catch (error) {
  console.error('Error parsing ideation response:', error);
  throw new Error(`Failed to parse ideation agent response: ${error.message}`);
}
```

INPUT

- Schema
- Table
- JSON

1 item

Ideation Agent

```

A8 output ``{json}`` \n\n
"alternatives": [\n  "Subscription-based\n'School Bus as a Service'\n(SaaS) model, partnering\ndirectly with schools to\nmanage optimized routes\nand dedicated vehicles,\nproviding a structured,\nsafe, and reliable transport\nsolution for their\nstudents.",\n  "Hyperlocal\nCarpool Network: A\ncommunity-driven platform\nconnecting parents."
]

```

Get data from Sheet

1 item

Google Sheets Trigger

1 item

Variables and context

Convert output to features, output

Execute step

Parameters

Mode: Run Once for All Items

Language: JavaScript

JavaScript

```

1 // Get the AI response
2 const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;
3
4 try {
5   // Try to parse as JSON directly
6   let ideationResult;
7   if (typeof aiResponse === 'string') {
8     // Clean the response - remove markdown formatting if present
9     const cleanResponse = aiResponse.replace(/\`{1,6}json\`{1,6}\n?`{1,6}/g, '');
10    ideationResult = JSON.parse(cleanResponse);
11  } else {
12    ideationResult = aiResponse;
13  }
14
15 // Validate required fields
16 const requiredFields = ['refined_idea', 'alternatives', 'core_problems'];

```

Type \$ for a list of special vars/methods. Debug by using console.log() statements and viewing their output in the browser console.

OUTPUT

- Schema
- Table

1 item

refined_idea : A secure, c ride-sharing service de exclusively for school children in Indian metro cities, offe rigorously vetted driver reliable and safe school solutions.

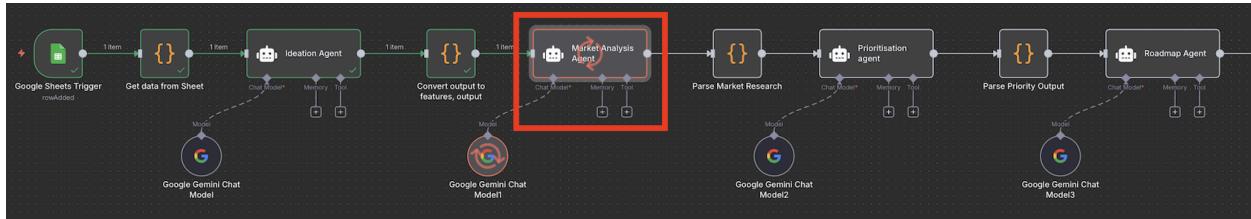
alternatives

- Subscription-based as a Service' (SaaS) model, partnering directly with schools to manage optimized routes and dedicated vehicles, providing a structured, safe, and reliable transport solution for their students.",
- "Hyperlocal Carpool Network: A community-driven platform connecting parents."

I wish this node would...

Step 4a - Market analysis Agent

Step 4: Agent 2 - Market Research Agent



4.1 Add AI Agent Node for Market Analysis

1. Add "AI Agent" node
2. Configure Agent Settings:
 - Agent Type: Text Generator
 - Model: Select Google Gemini
 - Model Name: gemini-2.0-flash-exp
 - API Key: Use your Gemini API key credential
 - Temperature: 0.3
 - Max Tokens: 800

4.2 Market Research Agent Prompt

Set the prompt field in the AI Agent node:

You are a Senior Market Research Agent with expertise in competitive analysis, market sizing, and trend identification across various industries.

CONTEXT:

- Product Idea: {{\$json.ideationResult.refined_idea}}
- Target Audience: {{\$json.targetAudience}}
- Key Features: {{\$json.ideationResult.key_features}}
- Value Proposition: {{\$json.ideationResult.value_proposition}}
- Business Context: {{\$json.businessContext}}

YOUR EXPERTISE:

You have deep knowledge of market dynamics, competitive landscapes, consumer behavior, and emerging trends. You provide data-driven insights that inform strategic decisions.

ANALYSIS FRAMEWORK:

1. MARKET SIZE: Estimate the Total Addressable Market (TAM) and Serviceable Available Market (SAM)
2. COMPETITIVE LANDSCAPE: Identify direct and indirect competitors with their strengths/weaknesses

3. TREND ANALYSIS: Highlight relevant market trends, technological shifts, and consumer behavior changes
4. OPPORTUNITY ASSESSMENT: Evaluate market gaps and white space opportunities
5. RISK EVALUATION: Identify potential market, competitive, and regulatory risks
6. MARKET TIMING: Assess whether market conditions are favorable for entry

OUTPUT FORMAT (must be valid JSON):

```
{  
  "market_size": "Detailed market size estimation with TAM/SAM breakdown",  
  "market_growth_rate": "Expected annual growth rate with rationale",  
  "competitors": [  
    {  
      "name": "Competitor 1",  
      "strength": "Key competitive advantage",  
      "weakness": "Main vulnerability",  
      "market_share": "Estimated market position"  
    },  
    {  
      "name": "Competitor 2",  
      "strength": "Key competitive advantage",  
      "weakness": "Main vulnerability",  
      "market_share": "Estimated market position"  
    }  
,  
    ],  
  "market_trends": ["Trend 1 with impact assessment", "Trend 2 with impact assessment",  
  "Trend 3 with impact assessment"],  
  "opportunities": ["Market gap 1", "Market gap 2", "Emerging opportunity"],  
  "risks": ["Risk 1 with likelihood", "Risk 2 with likelihood"],  
  "attractiveness_score": 8,  
  "market_timing": "Assessment of entry timing (Excellent/Good/Fair/Poor)",  
  "market_summary": "Comprehensive 2-3 sentence market analysis summary",  
  "strategic_recommendations": ["Market entry strategy 1", "Market entry strategy 2"]  
}
```

IMPORTANT: Provide realistic, research-backed assessments. Use a 1-10 scale for attractiveness_score where 10 is extremely attractive. Respond ONLY with the JSON object.

INPUT

Convert output to features, output

Market Analysis Agent

Parameters

Tip: Get a feel for agents with our quick tutorial or see an example of how this node works

Source for Prompt (User Message): Fixed Expression

Define below Parameter: "promptType"

Prompt (User Message): You are a Senior Market Research Agent with expertise in competitive analysis, market sizing, and trend identification across various industries.

CONTEXT: You are a Senior Market Research Agent with expertise in competitive analysis, market sizing, and trend identification across various industries.

Require Specific Output Format:

Enable Fallback Model:

Options: No properties

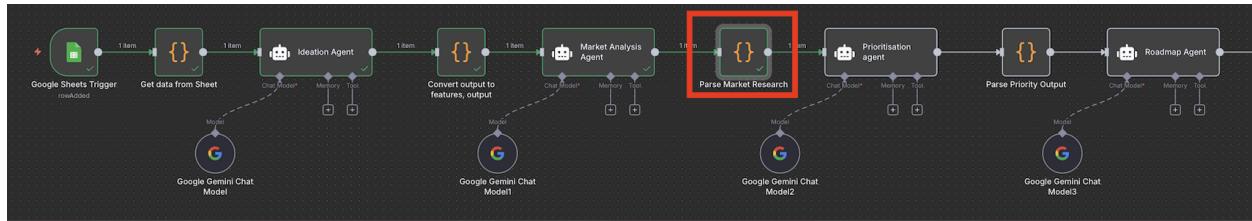
Add Option

Chat Model * Memory Tool

Execute this node to view data or set mock data

Step 4b - Market Analysis code

4.3 Parse Market Research Response



1. Add "Code" node after the Market Research AI Agent:

```

// Get the AI response
const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;

try {
    // Parse the JSON response
    let marketResult;
    if (typeof aiResponse === 'string') {
        const cleanResponse = aiResponse.replace(/\`json\n?|\n?\`/g, "").trim();
        marketResult = JSON.parse(cleanResponse);
    } else {
        marketResult = aiResponse;
    }

    // Validate critical fields
    const requiredFields = ['market_size', 'competitors', 'attractiveness_score',
'market_summary'];
    for (const field of requiredFields) {
        if (!marketResult[field]) {
            throw new Error(`Missing required field: ${field}`);
        }
    }

    // Ensure arrays are properly formatted
    if (!Array.isArray(marketResult.competitors)) marketResult.competitors = [];
    if (!Array.isArray(marketResult.market_trends)) marketResult.market_trends = [];
    if (!Array.isArray(marketResult.opportunities)) marketResult.opportunities = [];
    if (!Array.isArray(marketResult.risks)) marketResult.risks = [];

    // Validate attractiveness score
    if (typeof marketResult.attractiveness_score !== 'number' ||
        marketResult.attractiveness_score < 1 ||
        marketResult.attractiveness_score > 10) {
        marketResult.attractiveness_score = 5; // Default middle value
    }
}

```

```

return [
  json: {
    ...items[0].json,
    marketResult: marketResult
  }
];
} catch (error) {
  console.error('Error parsing market research response:', error);
  throw new Error(`Failed to parse market research agent response: ${error.message}`);
}

```

The screenshot shows a no-code AI workspace interface with the following details:

- INPUT:** A list of nodes including "Market Analysis Agent", "Convert output to features", "Ideation Agent", "Get data from Sheet", "Google Sheets Trigger", and "Variables and context".
- Node Configuration:** The central node is a "Parse Market Research" node.
- Parameters:**
 - Mode: Run Once for All Items
 - Language: JavaScript
- JavaScript Code:**

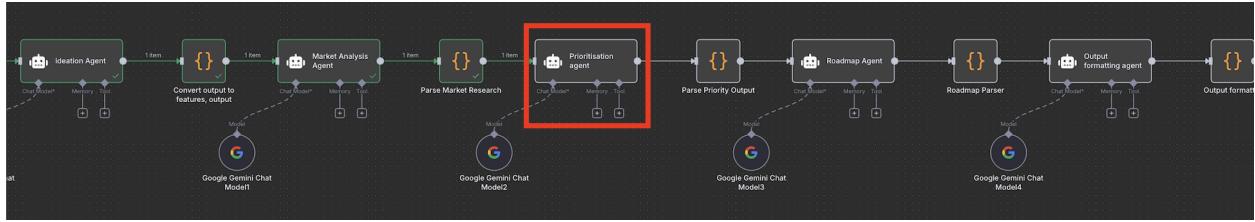
```

1 // Get the AI response
2 const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;
3
4 try {
5   // Parse the JSON response
6   let marketResult;
7   if (typeof aiResponse === 'string') {
8     const cleanResponse = aiResponse.replace(/\`json\n?|\n?\`/g, '').trim();
9     marketResult = JSON.parse(cleanResponse);
10 } else {
11   marketResult = aiResponse;
12 }
13
14 // Validate critical fields
15 const requiredFields = ['market_size', 'competitors', 'attractiveness_score',
'market_summary'];

```
- OUTPUT:** An output table showing the transformed data.

The output table includes columns for **output** and **marketResult**. The **marketResult** column displays the parsed JSON content, which includes fields like **market_size**, **tam_estimation**, and **sam_estimation**.

Step 5a - Prioritization Agent



Step 5: Agent 3 - Prioritization Agent

5.1 Add AI Agent Node for Priority Scoring

1. Add "AI Agent" node
2. Configure Agent Settings:
 - Agent Type: Text Generator
 - Model: Select Google Gemini
 - Model Name: gemini-2.0-flash-exp
 - API Key: Use your Gemini API key credential
 - Temperature: 0.2
 - Max Tokens: 600

5.2 Prioritization Agent Prompt

Set the prompt field in the AI Agent node:

You are a Strategic Prioritization Agent specializing in product portfolio management and ROI optimization using industry-standard frameworks.

CONTEXT:

- Product Idea: {{\$json.ideationResult.refined_idea}}
- Key Features: {{\$json.ideationResult.key_features}}
- Market Analysis: {{\$json.marketResult.market_summary}}
- Market Attractiveness: {{\$json.marketResult.attractiveness_score}}/10
- Business Context: {{\$json.businessContext}}
- Constraints: {{\$json.constraints}}
- Timeline Preference: {{\$json.timelinePreference}}
- Competitive Risks: {{\$json.marketResult.risks}}

YOUR EXPERTISE:

You apply proven prioritization frameworks (RICE, ICE, Value vs Effort, MoSCoW) to evaluate product opportunities. You balance quantitative metrics with strategic considerations.

SCORING CRITERIA (Rate each 1-10 where 10 is best):

1. BUSINESS VALUE: Revenue potential, strategic alignment, market size impact

2. USER IMPACT: User satisfaction gain, problem significance, user base size affected
3. EFFORT REQUIRED: Development complexity, resource requirements, time investment (10 = low effort, 1 = high effort)
4. RISK LEVEL: Technical feasibility, market acceptance, execution risks (10 = low risk, 1 = high risk)
5. MARKET TIMING: Market readiness, competitive window, trend alignment
6. STRATEGIC FIT: Alignment with business goals, platform synergies, brand consistency

CALCULATION: Priority Score = (Business Value + User Impact + Market Timing + Strategic Fit) - (Effort Required + Risk Level)

OUTPUT FORMAT (must be valid JSON):

```
{
  "business_value": 8,
  "user_impact": 7,
  "effort_required": 6,
  "risk_level": 4,
  "market_timing": 8,
  "strategic_fit": 7,
  "calculated_score": 20,
  "priority_tier": "High",
  "confidence_level": "Medium",
  "ice_score": {
    "impact": 8,
    "confidence": 7,
    "ease": 6
  },
  "recommendation": "Proceed/Hold/Reject",
  "justification": "2-3 sentence explanation of the scoring rationale and key factors",
  "success_probability": 75,
  "roi_estimate": "Expected return on investment assessment",
  "critical_success_factors": ["Factor 1", "Factor 2", "Factor 3"]
}
```

PRIORITY TIERS:

- High (Score 15+): Immediate execution recommended
- Medium (Score 8-14): Consider for next planning cycle
- Low (Score <8): Revisit when constraints change

IMPORTANT: Be objective and realistic. Consider the stated constraints heavily in your scoring. Respond ONLY with the JSON object.

[Back to canvas](#) [+](#)

INPUT

- Parse Market Research
 - output: json
 - market_size: "market_size": 1, "tam_estimation": "The Total Addressable Market (TAM) for school children in Indian major Tier-1 and Tier-2 metro cities, there are approximately 30-40 million school-going children (ages 3-18). This represents the total pool of potential users if an ideal, universally affordable solution existed.", "sam_estimation": "The Servicable Available Market (SAM) focuses on children whose parents have the disposable income, need for convenience, and prioritize enhanced safety features that justify a premium service. This segment includes middle to upper-middle-class and affluent families in urban centers. We estimate this to be 10-15% of the TAM, translating to approximately 3-6 million children. Assuming an average monthly subscription of INR 3,500-5,000 per child for 10 academic months, the annual SAM revenue potential could range from INR 105 billion to INR 300 billion (approx. \$1.25B - \$3.6B USD).", "market_growth_rate": "The market for child-specific services, particularly in urban transport, is expected to grow at a Compound Annual Growth Rate (CAGR) of 18-25% over the next 5-7 years. This growth is driven by increasing urbanization, changing family dynamics, and a focus on child safety and well-being." }

Prioritisation agent

[Execute step](#)

Parameters **Settings** **Docs**

Tip: Get a feel for agents with our quick [tutorial](#) or see an [example](#) of how this node works

Source for Prompt (User Message)

Define below

Prompt (User Message)

You are a Strategic Prioritization Agent specializing in product portfolio management and ROI optimization using industry-standard frameworks.

CONTXT:
You are a Strategic Prioritization Agent specializing in product portfolio management and ROI optimization using industry-standard frameworks.

Require Specific Output Format

Enable Fallback Model

Options

No properties [Add Option](#)

Chat Model* [Memory](#) [Tool](#)

Output **Logs** [Q](#) [Schema](#) [Table](#) [JSON](#) [Edit](#) [X](#)

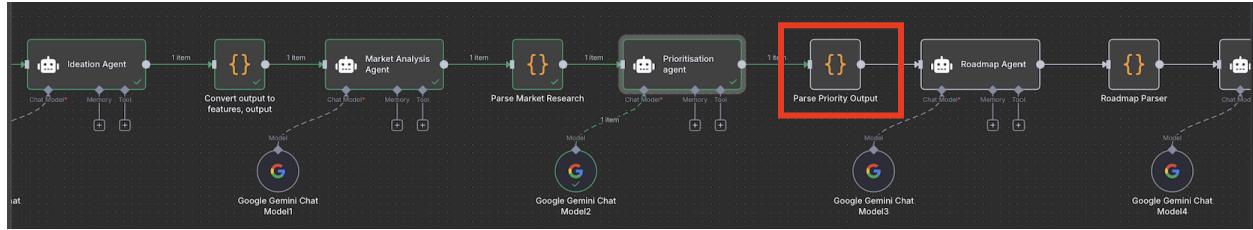
1 item

output

```
":json\n{\n  \"business_value\": 5,\n  \"user_impact\": 9,\n  \"effort_required\": 1,\n  \"risk_level\": 1,\n  \"market_timing\": 8,\n  \"strategic_fit\": 8,\n  \"calculated_lscore\": 29,\n  \"priority_tier\": \"High\", \n  \"confidence_level\": \"Low\", \n  \"ice_score\": (\n    \"impact\": 9,\n    \"confidence\": 2,\n    \"ease\": 1\n  ),\n  \"recommendation\": \"Reject\", \n  \"justification\": \"While the market demand and potential user impact are exceptionally high, the project is rendered unfeasible by the severe constraints. A $25,000 budget and 3-month timeline are critically inadequate for building a secure, compliant, and reliable child transportation service, leading to extremely high execution risks (safety, regulatory, operational) and an overwhelming effort requirement.\",\n  \"success_probability\": 5,\n  \"to_estimate\": \"Negative (Likely loss of initial investment due to inability to launch a viable, safe, and compliant service under current constraints)\",\n  \"critical_success_factors\": [\n    \"Significantly increased budget (minimum 5-10x current) and extended timeline (6-12 months for MVP)\",\n    \"Impeccable Safety & Regulatory Compliance (non-negotiable for child transport, requires significant investment)\",\n    \"Robust Driver Vetting, Training, and Retention Program (core to trust and service quality)\",\n    \"Effective Trust Building & Crisis Management Strategy (essential for parental confidence)\"\n  ]\n}\n\nI wish this node would... increased budget (minimum 5-10x current) and extended timeline (6-12 months for MVP), Impeccable Safety & Regulatory Compliance (non-negotiable for child transport, requires significant investment), Robust Driver Vetting, Training, and Retention Program (core to trust and service quality), Effective Trust Building & Crisis Management Strategy (essential for parental confidence)"
```

Step 5b - Priority Code node

5.3 Parse Priority Response



1. Add "Code" node after the Prioritization AI Agent:

```

// Get the AI response
const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;

try {
  // Parse the JSON response
  let priorityResult;
  if (typeof aiResponse === 'string') {
    const cleanResponse = aiResponse.replace(/\`json\n?|\\n?\`/g, "").trim();
    priorityResult = JSON.parse(cleanResponse);
  } else {
    priorityResult = aiResponse;
  }

  // Validate and normalize scores (must be 1-10)
  const scoreFields = ['business_value', 'user_impact', 'effort_required', 'risk_level',
  'market_timing', 'strategic_fit'];
  for (const field of scoreFields) {
    if (typeof priorityResult[field] !== 'number' ||
        priorityResult[field] < 1 ||
        priorityResult[field] > 10) {
      priorityResult[field] = 5; // Default middle value
    }
  }

  // Recalculate priority score to ensure accuracy
  const calculatedScore =
    priorityResult.business_value +
    priorityResult.user_impact +
    priorityResult.market_timing +
    priorityResult.strategic_fit
  ) - (
  
```

```

        (10 - priorityResult.effort_required + 1) + // Invert effort (higher effort = lower score)
        (10 - priorityResult.risk_level + 1) // Invert risk (higher risk = lower score)
    );

priorityResult.calculated_score = Math.round(calculatedScore);

// Determine priority tier based on calculated score
if (priorityResult.calculated_score >= 15) {
    priorityResult.priority_tier = "High";
} else if (priorityResult.calculated_score >= 8) {
    priorityResult.priority_tier = "Medium";
} else {
    priorityResult.priority_tier = "Low";
}

// Ensure arrays exist
if (!Array.isArray(priorityResult.critical_success_factors)) {
    priorityResult.critical_success_factors = [];
}

// Validate ICE score object
if (!priorityResult.ice_score || typeof priorityResult.ice_score !== 'object') {
    priorityResult.ice_score = {
        impact: priorityResult.user_impact,
        confidence: 5,
        ease: priorityResult.effort_required
    };
}

return [
    json: {
        ...items[0].json,
        priorityResult: priorityResult
    }
];
} catch (error) {
    console.error('Error parsing prioritization response:', error);
    throw new Error(`Failed to parse prioritization agent response: ${error.message}`);
}

```

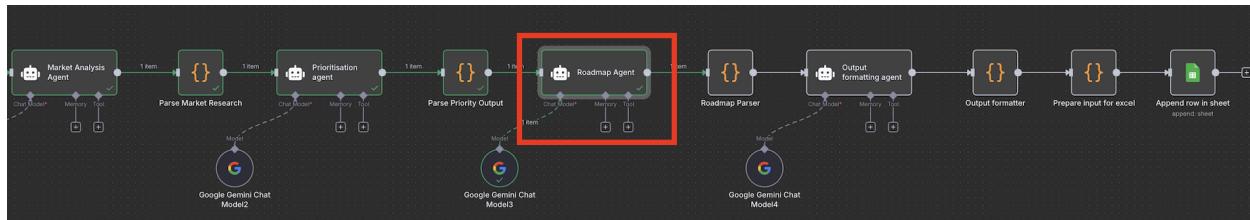
The screenshot shows a no-code AI tool interface with the following details:

- Header:** Back to canvas, INPUT, Q, Schema, Table, JSON.
- Left Sidebar:** Prioritisation agent (1 item), Parse Market Research (1 item), Market Analysis Agent (1 item), Convert output to features, output (1 item), Ideation Agent (1 item), Get data from Sheet (1 item), Google Sheets Trigger (1 item), Variables and context.
- Current Node:** Parse Priority Output
- Node Configuration:**
 - Parameters:** Mode (Run Once for All Items), Language (JavaScript).
 - JavaScript Code:**

```
1 // Get the AI response
2 const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;
3
4 try {
5     // Parse the JSON response
6     let priorityResult;
7     if (typeof aiResponse === 'string') {
8         const cleanResponse = aiResponse.replace(/\`json\?\n|\n\`g, '').trim();
9         priorityResult = JSON.parse(cleanResponse);
10    } else {
11        priorityResult = aiResponse;
12    }
13
14    // Validate and normalize scores (must be 1-10)
15    const scoreFields = ['business_value', 'user_impact', 'effort_required', 'risk_level',
16        'market_timing', 'strategic_fit', 'calculated_score', 'priority_tier', 'confidence_level',
17        'ice_score', 'recommendation', 'justification'];
18    for (const field of scoreFields) {
19        if (!priorityResult[field]) priorityResult[field] = 0;
20        priorityResult[field] = Math.max(0, Math.min(10, priorityResult[field]));
21    }
22}
23
24// Return the normalized priority result
25return priorityResult;
```
 - Type hint:** Type \$ for a list of special vars/methods. Debug by using console.log() statements and viewing their output in the browser console.
- Output:** business_value : 5, user_impact : 9, effort_required : 1, risk_level : 1, market_timing : 9, strategic_fit : 8, calculated_score : 29, priority_tier : "High", confidence_level : "Low", ice_score : {}, impact : 9, confidence : 2, ease : 1, recommendation : "Reject", justification : "While the market demand and potential user impact are exceptionally high, the project is rendered unfeasible by the severe constraints. A \$25,000 budget and 3-month timeline are critically inadequate for building a secure, compliant, and reliable child L..."
- Bottom Status:** ARUN Nandewal, Prioritisation agent, timeline (6-12 months for MVP), "Impeccable Safety & Regulatory Compliance (non-negotiable for child transport, requires significant investment)", "Robust Driver

Step 6a - Roadmap Agent

Step 6: Agent 4 - Roadmap Agent



6.1 Add AI Agent Node for Roadmap Planning

1. Add "AI Agent" node
2. Configure Agent Settings:
 - Agent Type: Text Generator
 - Model: Select Google Gemini
 - Model Name: gemini-2.0-flash-exp
 - API Key: Use your Gemini API key credential
 - Temperature: 0.4
 - Max Tokens: 800

6.2 Roadmap Planning Agent Prompt

Set the prompt field in the AI Agent node:

You are a Senior Product Roadmap Agent specializing in agile product development, feature prioritization, and strategic execution planning.

CONTEXT:

- Product Idea: {{\$json.ideationResult.refined_idea}}
- Key Features: {{\$json.ideationResult.key_features}}
- Priority Score: {{\$json.priorityResult.calculated_score}} ({{\$json.priorityResult.priority_tier}} Priority)
- Effort Assessment: {{\$json.priorityResult.effort_required}}/10
- Risk Level: {{\$json.priorityResult.risk_level}}/10
- Timeline Preference: {{\$json.timelinePreference}}
- Business Constraints: {{\$json.constraints}}
- Market Timing: {{\$json.priorityResult.market_timing}}/10
- Success Factors: {{\$json.priorityResult.critical_success_factors}}

YOUR EXPERTISE:

You excel at breaking down complex products into executable phases, identifying dependencies, and creating realistic timelines. You apply lean startup principles and agile methodologies.

ROADMAP FRAMEWORK:

1. MVP (Minimum Viable Product): Core features that validate the concept and deliver immediate value
2. Phase 2 (Growth): Features that scale usage and improve user experience
3. Phase 3 (Scale): Advanced features for market leadership and expansion
4. Consider technical debt, user feedback cycles, and market response

TIMELINE CONSIDERATIONS:

- Account for development complexity, team capacity, and market windows
- Include buffer time for testing, iterations, and unforeseen challenges
- Balance speed-to-market with quality and user experience

OUTPUT FORMAT (must be valid JSON):

```
{  
    "mvp_features": ["Core feature 1", "Core feature 2", "Essential feature 3"],  
    "mvp_timeline": "2-3 months",  
    "mvp_success_metrics": ["Metric 1 with target", "Metric 2 with target"],  
    "phase2_features": ["Growth feature 1", "Enhancement 2", "Integration 3"],  
    "phase2_timeline": "4-6 months from start",  
    "phase2_success_metrics": ["Scale metric 1", "Engagement metric 2"],  
    "phase3_features": ["Advanced feature 1", "Platform expansion 2"],  
    "phase3_timeline": "8-12 months from start",  
    "phase3_success_metrics": ["Market metric 1", "Revenue metric 2"],  
    "dependencies": [  
        {  
            "prerequisite": "Feature A",  
            "dependent": "Feature B",  
            "reason": "Technical dependency explanation"  
        }  
    ],  
    "critical_path": ["MVP Feature 1 → Phase 2 Feature 1 → Phase 3 Feature 1"],  
    "parallel_development": ["Features that can be built simultaneously"],  
    "team_requirements": {  
        "mvp_phase": ["Role 1", "Role 2", "Skill requirement"],  
        "scaling_phase": ["Additional role 1", "Specialized skill 2"]  
    },  
    "risk_mitigation_plan": [  
        {  
            "risk": "Technical risk description",  
            "mitigation": "Specific action to reduce risk",  
            "timeline_impact": "Potential delay assessment"  
        }  
    ],  
    "resource_allocation": {
```

```

    "development_effort": "% breakdown by phase",
    "design_effort": "% breakdown by phase",
    "testing_effort": "% breakdown by phase"
},
"go_to_market_timeline": ["Pre-launch activities", "Launch activities", "Post-launch activities"],
"iteration_plan": "Strategy for incorporating user feedback and market learnings"
}

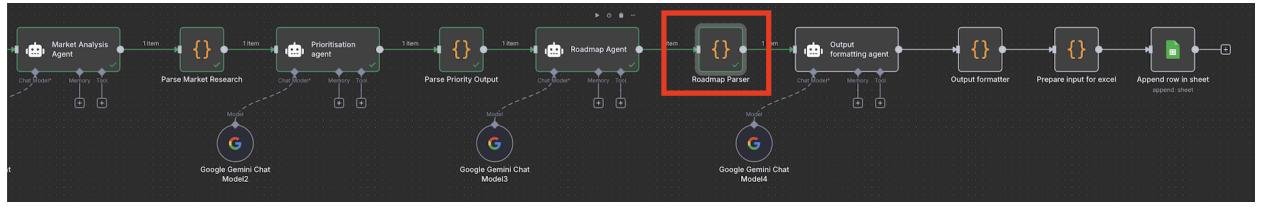
```

IMPORTANT: Be realistic about timelines and account for the stated constraints. Ensure MVP is truly minimal but viable. Respond ONLY with the JSON object.

The screenshot shows the configuration of a 'Roadmap Agent' node in a no-code AI tool. The input section displays a JSON object representing project metrics. The output section shows the generated JSON object. The main panel contains a prompt for the agent, specifying it's a Senior Product Roadmap Agent specializing in agile product development, prioritization, and strategic execution planning. The prompt includes context about being a roadmap agent and provides a sample message template. The tool also includes sections for parameters, settings, and documentation.

Step 6b - Roadmap Code

6.3 Parse Roadmap Response



1. Add "Code" node after the Roadmap AI Agent:

```
// Get the AI response
const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;

try {
    // Parse the JSON response
    let roadmapResult;
    if (typeof aiResponse === 'string') {
        const cleanResponse = aiResponse.replace(/\`json\n?|\n?\`\//g, "").trim();
        roadmapResult = JSON.parse(cleanResponse);
    } else {
        roadmapResult = aiResponse;
    }

    // Validate required fields and provide defaults
    const requiredFields = {
        'mvp_features': [],
        'mvp_timeline': '3 months',
        'phase2_features': [],
        'phase2_timeline': '6 months',
        'phase3_features': [],
        'phase3_timeline': '12 months',
        'dependencies': [],
        'team_requirements': {},
        'mvp_success_metrics': [],
        'critical_path': []
    };

    // Ensure all required fields exist with defaults
    for (const [field, defaultValue] of Object.entries(requiredFields)) {
        if (!roadmapResult[field]) {
            roadmapResult[field] = defaultValue;
        }
    }
}
```

```

// Ensure arrays are properly formatted
const arrayFields = ['mvp_features', 'phase2_features', 'phase3_features', 'dependencies',
    'critical_path', 'parallel_development', 'mvp_success_metrics',
    'phase2_success_metrics', 'phase3_success_metrics', 'go_to_market_timeline'];

for (const field of arrayFields) {
    if (roadmapResult[field] && !Array.isArray(roadmapResult[field])) {
        roadmapResult[field] = [];
    }
}

// Ensure nested objects exist
if (!roadmapResult.team_requirements || typeof roadmapResult.team_requirements !==
'object') {
    roadmapResult.team_requirements = {
        mvp_phase: ['Product Manager', 'Frontend Developer', 'Backend Developer'],
        scaling_phase: ['UI/UX Designer', 'DevOps Engineer', 'QA Engineer']
    };
}

if (!roadmapResult.resource_allocation || typeof roadmapResult.resource_allocation !==
'object') {
    roadmapResult.resource_allocation = {
        development_effort: '60% MVP, 30% Phase 2, 10% Phase 3',
        design_effort: '40% MVP, 35% Phase 2, 25% Phase 3',
        testing_effort: '50% MVP, 30% Phase 2, 20% Phase 3'
    };
}

// Create a summary for easy reference
roadmapResult.roadmap_summary = `MVP (${roadmapResult.mvp_timeline}):
${roadmapResult.mvp_features.length} features`;

return [
    json: {
        ...items[0].json,
        roadmapResult: roadmapResult
    }
];
} catch (error) {
    console.error('Error parsing roadmap response:', error);
    throw new Error(`Failed to parse roadmap agent response: ${error.message}`);
}

```

[Back to canvas](#)

INPUT Q Schema Table JSON

Parameter: "language"

- Roadmap Agent 1 item
 - output: "json\n\n{\\n \"mvp_features\": [\\n \"Rigorous Driver Screening & Training Program (not scale)\", \"OTP-based Secure Digital Child Handover Protocol\", \"Emergency Alert System\", \"Flexible Scheduling (Core functionality in Parent App)\", \"Basic Parent App (AI-powered single trip booking Tracking, Notifications)\", \"Basic Driver App (Trip management, Navigation, OTP validation, etc.)\", \"Emergency Communication Feature (Emergency button/hotline)\", \"Geo-fencing for Designated Pick/Drop-off Zones\"], \\n \"months\" (for initial tech build & pilot readiness, assuming concurrent operations...)}
- Parse Priority Output 1 item
- Prioritisation agent 1 item
- Parse Market Research 1 item
- Market Analysis Agent 1 item
- Convert output to features, output 1 item
- Ideation Agent 1 item
- Get data from Sheet 1 item
 - Parse Product Requirements

Roadmap Parser

Parameters Settings Docs

Mode: Run Once for All Items

Language: JavaScript

```

1 // Get the AI response
2 const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;
3
4 try {
5   // Parse the JSON response
6   let roadmapResult;
7   if (typeof aiResponse === 'string') {
8     const cleanResponse = aiResponse.replace(/^\s*json\s*\n?(\n|\\r|\\t)*\s*/g, '').trim();
9     roadmapResult = JSON.parse(cleanResponse);
10 } else {
11   roadmapResult = aiResponse;
12 }
13
14 // Validate required fields and provide defaults
15 const requiredFields = [
16   'mvp_features': []
];

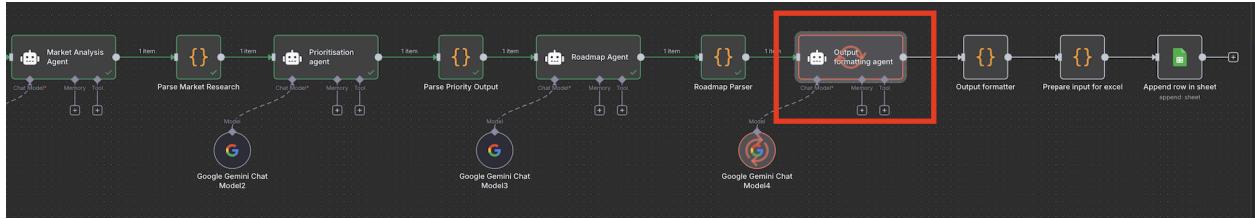
```

Type \$ for a list of **special vars/methods**. Debug by using `console.log()` statements and viewing their output in the browser console.

Execute this node to view data or set mock data

! I wish this node would... Flexible Scheduling

Step 7a - Output formatting Agent



Step 7: Agent 5 - Output Formatting Agent

7.1 Add AI Agent Node for Final Analysis

1. Add "AI Agent" node
2. Configure Agent Settings:
 - Agent Type: Text Generator
 - Model: Select Google Gemini
 - Model Name: gemini-2.0-flash-exp
 - API Key: Use your Gemini API key credential
 - Temperature: 0.3
 - Max Tokens: 600

7.2 Output Formatting Agent Prompt

Set the prompt field in the AI Agent node:

You are an Executive Summary Agent specializing in synthesizing complex product analysis into actionable business recommendations for stakeholders and decision-makers.

COMPREHENSIVE ANALYSIS DATA:

- Refined Product Idea: {{\$json.ideationResult.refined_idea}}
- Value Proposition: {{\$json.ideationResult.value_proposition}}
- Market Attractiveness: {{\$json.marketResult.attractiveness_score}}/10
- Priority Score: {{\$json.priorityResult.calculated_score}} ({{\$json.priorityResult.priority_tier}} Priority)
- Market Summary: {{\$json.marketResult.market_summary}}
- MVP Timeline: {{\$json.roadmapResult.mvp_timeline}}
- Key Risks: {{\$json.marketResult.risks}}
- Success Probability: {{\$json.priorityResult.success_probability}}%
- Recommendation: {{\$json.priorityResult.recommendation}}
- Business Context: {{\$json.businessContext}}
- Resource Requirements: {{\$json.roadmapResult.team_requirements}}

YOUR EXPERTISE:

You excel at distilling complex analysis into clear, actionable insights that drive executive decision-making. You balance optimism with realism and provide concrete next steps.

EXECUTIVE COMMUNICATION PRINCIPLES:

1. Lead with the recommendation and key insights
2. Provide clear rationale backed by data
3. Address risks proactively with mitigation strategies
4. Focus on business impact and ROI
5. Include specific, measurable next steps

OUTPUT FORMAT (must be valid JSON):

```
{  
    "executive_summary": "Compelling 2-3 sentence summary highlighting the opportunity, recommendation, and expected outcome",  
    "key_insights": [  
        "Primary insight about market opportunity",  
        "Critical insight about competitive advantage",  
        "Important insight about execution feasibility"  
],  
    "recommendation": "Clear Go/No-Go recommendation with confidence level",  
    "business_case": {  
        "opportunity_size": "Market size and revenue potential assessment",  
        "competitive_advantage": "Key differentiators and defensibility",  

```

```

        "owner": "Recommended responsible party"
    }
],
"decision_timeline": "Recommended timeframe for go/no-go decision",
"resource_requirements_summary": "High-level summary of team, budget, and time requirements",
"strategic_alignment": "Assessment of how this aligns with broader business strategy and goals"
}

```

IMPORTANT: Be decisive and actionable. Avoid hedging language. Provide specific, measurable recommendations. Respond ONLY with the JSON object.

The screenshot shows a workflow interface with two main nodes:

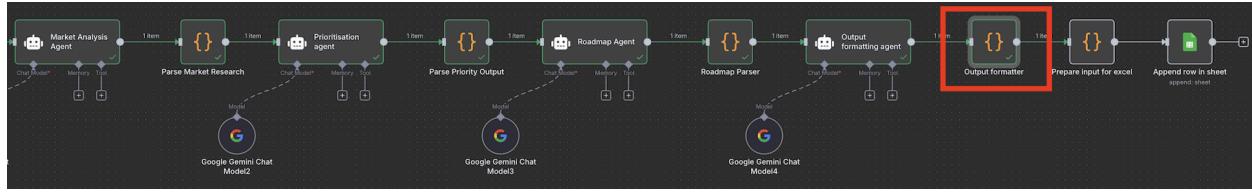
- Roadmap Parser**: This node has an output labeled `mvp_features`. When expanded, it shows 9 items, each describing a feature or process:
 - `mvp_features[0]`: Driver Vetting & Training Process (Operational & Admin Portal)
 - `mvp_features[1]`: Parent App (Android Only): Registration, Profile, Child Management
 - `mvp_features[2]`: Driver App (Android Only): Login, Ride Acceptance/Rejection, Route Display
 - `mvp_features[3]`: Basic Recurring Ride Scheduling (Limited routes/drivers for initial control)
 - `mvp_features[4]`: Real-time GPS Tracking (Live vehicle location during active ride for parents)
 - `mvp_features[5]`: OTP-based Child Handover Protocol (Pickup & Drop-off verification)
 - `mvp_features[6]`: In-app Notifications for Ride Status Updates (e.g., 'Driver Arrived', 'Child Picked Up')
 - `mvp_features[7]`: Emergency Call Button (within apps, direct line to support hotline)
 - `mvp_features[8]`: Basic Admin Panel (Driver/Ride Management, Support Call Log)
- Output formatting agent**: This node takes the `mvp_features` output and generates a JSON object for an executive summary. The `Source for Prompt (User Message)` field contains a template with placeholder text like "You are an Executive Summary Agent specializing in synthesizing complex product analysis into actionable business recommendation s for stakeholders and decision-makers." The generated `Output` JSON is as follows:


```

{
  "executive_summary": "Despite a highly attractive market for secure, on-demand school transport in Indian metro cities, we recommend rejecting the current product concept due to an extremely high operational and reputational risk profile, combined with an unacceptably low 5% success probability. This decision prevents substantial potential financial and brand damage associated with direct liability in child transport.", 
  "key_insights": "The market presents a compelling opportunity (Market Attractiveness: 8/10) driven by increased parental safety concerns and demand for tech-enabled convenience in a largely unorganized sector.", 
  "recommendation": "Reject (High Confidence)", 
  "business_case": "Significant premium market segment in Indian metro cities for reliable, safe school transport, driven by dual-income households and heightened parental safety concerns. Potential for substantial revenue if trust can be flawlessly established and maintained.", 
  "competitive_advantage": "Potential for first-mover advantage in a professional, tech-enabled, and highly vetted service in a fragmented and unorganized sector. However, this advantage is heavily contingent on perfect execution and overcoming massive initial trust barriers.", 
  "resource_investment": "Substantial investment required in time (realistic 6-12 months for MVP), capital (for advanced tech, vehicle acquisition/conversion, extensive vetting/training, legal/compliance, and crisis management infrastructure), and highly specialized personnel (safety, legal, operations).", 
  "expected_roi": "Negative ROI is highly probable due to the extremely high likelihood of significant financial and reputational losses from potential safety incidents, coupled with the high operational complexities, stringent regulatory hurdles, and substantial customer acquisition costs required to build and maintain trust."
}
            
```

Step 7b - Code output

7.3 Parse Output Formatting Response



1. Add "Code" node after the Output Formatting AI Agent:

```

// Get the AI response
const aiResponse = items[0].json.output || items[0].json.text || items[0].json.response;

try {
    // Parse the JSON response
    let outputResult;
    if (typeof aiResponse === 'string') {
        const cleanResponse = aiResponse.replace(/\`json\n?|\n?\`\//g, "").trim();
        outputResult = JSON.parse(cleanResponse);
    } else {
        outputResult = aiResponse;
    }

    // Validate required fields with defaults
    const requiredFields = {
        'executive_summary': 'Analysis completed for product opportunity assessment.',
        'recommendation': 'Further evaluation recommended.',
        'key_insights': [],
        'immediate_next_steps': [],
        'success_metrics': [],
        'decision_timeline': '2 weeks'
    };

    for (const [field, defaultValue] of Object.entries(requiredFields)) {
        if (!outputResult[field]) {
            outputResult[field] = defaultValue;
        }
    }

    // Ensure nested objects exist
    if (!outputResult.business_case || typeof outputResult.business_case !== 'object') {
        outputResult.business_case = {
            opportunity_size: 'Market opportunity assessment pending',
    
```

```

        competitive_advantage: 'Competitive positioning to be determined',
        resource_investment: 'Resource requirements analysis needed',
        expected_roi: 'ROI projection to be calculated'
    };
}

// Ensure arrays are properly formatted
const arrayFields = ['key_insights', 'immediate_next_steps', 'success_metrics',
'risk_mitigation_plan'];
for (const field of arrayFields) {
    if (outputResult[field] && !Array.isArray(outputResult[field])) {
        outputResult[field] = [];
    }
}

// Validate risk mitigation plan structure
if (outputResult.risk_mitigation_plan && Array.isArray(outputResult.risk_mitigation_plan)) {
    outputResult.risk_mitigation_plan = outputResult.risk_mitigation_plan.map(risk => {
        if (typeof risk === 'string') {
            return {
                risk: risk,
                impact: 'Medium',
                mitigation: 'Mitigation strategy to be defined',
                owner: 'Product team'
            };
        }
        return risk;
    });
}

return [
    json: {
        ...items[0].json,
        outputResult: outputResult
    }
];
} catch (error) {
    console.error('Error parsing output formatting response:', error);
    throw new Error(`Failed to parse output formatting agent response: ${error.message}`);
}

```

INPUT Schema Table JSON

Output formatter

Parameters Settings Docs

Mode Run Once for All Items

Language JavaScript

JavaScript

```

1 // Get the AI response
2 const aiResponse = items[0].json.output || items[0].text || items[0].response;
3
4 try {
5   // Parse the JSON response
6   let outputResult;
7   if (typeof aiResponse === 'string') {
8     const cleanResponse = aiResponse.replace(/\n|\r/g, '').trim();
9     outputResult = JSON.parse(cleanResponse);
10 } else {
11   outputResult = aiResponse;
12 }
13
14 // Validate required fields with defaults
15 const requiredFields = [
16   'executive_summary': 'Analysis completed for product opportunity assessment.',
```

Type \$ for a list of **special vars/methods**. Debug by using `console.log()` statements and viewing their output in the browser console.

OUTPUT Schema Table

To make sure expressions after this node work, return the input items that produced each output item. [More info](#)

1 item

output **outputResult**

```
"json\n\n"executive_summary": "Despite a highly attractive market for secure, on-demand school transport in Indian metro cities, we recommend rejecting the current product concept due to an extremely high operational and reputational risk profile, combined with an unacceptably low 5% success probability. This decision prevents substantial potential financial and brand damage associated with direct liability in child transport.",\n"key_insights": [\n  "The market presents a compelling opportunity (Market Attractiveness: 8/10) driven by increased parental safety concerns and demand for tech-enabled convenience in a largely unorganized sector.",\n  "The core challenge lies in the exceptionally high-stakes nature of child transport; any perceived or actual safety incident poses catastrophic reputational and financial risks."]
```

key_insights

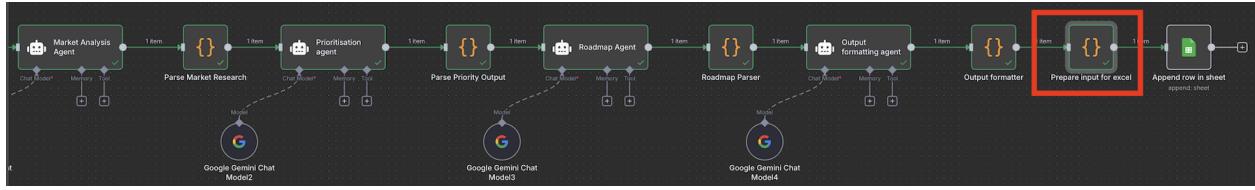
0 : The market presents a compelling opportunity (Market Attractiveness: 8/10) driven by increased parental safety concerns and demand for tech-enabled convenience in a largely unorganized sector.

1 : The core challenge lies in the exceptionally high-stakes nature of child transport; any perceived or actual safety incident poses catastrophic reputational and financial risks.

I wish this node would...

ARUN Nandewal ... Output formatter

Step 8a - Prepare Excel Code



Step 8: Format Output for Google Sheets

8.1 Data Preparation Code Node

1. Add "Code" node after the Output Formatting Agent
2. Prepare comprehensive data for Google Sheets output:

```
// Generate unique ID for this analysis session
const ideaId = `IDEA_${Date.now()}_${Math.random().toString(36).substr(2, 5)}`;
const analysisDate = new Date().toISOString().split('T')[0];
const analysisTime = new Date().toISOString().split('T')[1].split('.')[0];

// Helper function to safely join arrays
const safeJoin = (arr, separator = ';') => {
  if (!arr || !Array.isArray(arr)) return '';
  return arr.map(item =>
    typeof item === 'object' ? JSON.stringify(item) : String(item)
  ).join(separator);
};

// Helper function to extract nested object values
const extractNestedData = (obj, defaultValue = '') => {
  if (!obj || typeof obj !== 'object') return defaultValue;
  return Object.entries(obj)
    .map(([key, value]) => `${key}: ${value}`)
    .join(';');
};

// Compile comprehensive results for Google Sheets
const finalOutput = {
  // Primary Identifiers
  idea_id: ideaId,
  analysis_date: analysisDate,
  analysis_time: analysisTime,

  // Core Product Information
}
```

```
original_idea: $json.ideaDescription || "",  
refined_idea: $json.ideationResult?.refined_idea || "",  
value_proposition: $json.ideationResult?.value_proposition || "",  
target_audience: $json.targetAudience || "",  
business_context: $json.businessContext || "",  
  
// Ideation Results  
alternative_ideas: safeJoin($json.ideationResult?.alternatives),  
core_problems: safeJoin($json.ideationResult?.core_problems),  
key_features: safeJoin($json.ideationResult?.key_features),  
innovation_score: $json.ideationResult?.innovation_score || 0,  
  
// Market Analysis  
market_summary: $json.marketResult?.market_summary || "",  
market_size: $json.marketResult?.market_size || "",  
market_attractiveness_score: $json.marketResult?.attractiveness_score || 0,  
market_growth_rate: $json.marketResult?.market_growth_rate || "",  
market_timing: $json.marketResult?.market_timing || "",  
competitors: safeJoin($json.marketResult?.competitors?.map(c => `${c.name}  
(${c.strength}))`)),  
market_trends: safeJoin($json.marketResult?.market_trends),  
market_opportunities: safeJoin($json.marketResult?.opportunities),  
market_risks: safeJoin($json.marketResult?.risks),  
  
// Prioritization Results  
priority_score: $json.priorityResult?.calculated_score || 0,  
priority_tier: $json.priorityResult?.priority_tier || 'Medium',  
business_value_score: $json.priorityResult?.business_value || 0,  
user_impact_score: $json.priorityResult?.user_impact || 0,  
effort_required_score: $json.priorityResult?.effort_required || 0,  
risk_level_score: $json.priorityResult?.risk_level || 0,  
market_timing_score: $json.priorityResult?.market_timing || 0,  
strategic_fit_score: $json.priorityResult?.strategic_fit || 0,  
success_probability: $json.priorityResult?.success_probability || 0,  
recommendation: $json.priorityResult?.recommendation || 'Review',  
ice_impact: $json.priorityResult?.ice_score?.impact || 0,  
ice_confidence: $json.priorityResult?.ice_score?.confidence || 0,  
ice_ease: $json.priorityResult?.ice_score?.ease || 0,  
  
// Roadmap Information  
mvp_features: safeJoin($json.roadmapResult?.mvp_features),  
mvp_timeline: $json.roadmapResult?.mvp_timeline || "",  
phase2_features: safeJoin($json.roadmapResult?.phase2_features),  
phase2_timeline: $json.roadmapResult?.phase2_timeline || "",
```

```
phase3_features: safeJoin($json.roadmapResult?.phase3_features),
phase3_timeline: $json.roadmapResult?.phase3_timeline || '',
dependencies: safeJoin($json.roadmapResult?.dependencies?.map(d => `${d.prerequisite} → ${d.dependent}`)),
critical_path: safeJoin($json.roadmapResult?.critical_path),
mvp_success_metrics: safeJoin($json.roadmapResult?.mvp_success_metrics),
team_requirements_mvp: safeJoin($json.roadmapResult?.team_requirements?.mvp_phase),
team_requirements_scale:
safeJoin($json.roadmapResult?.team_requirements?.scaling_phase),
resource_allocation: extractNestedData($json.roadmapResult?.resource_allocation),

// Executive Summary & Output
executive_summary: $json.outputResult?.executive_summary || '',
final_recommendation: $json.outputResult?.recommendation || '',
key_insights: safeJoin($json.outputResult?.key_insights),
immediate_next_steps: safeJoin($json.outputResult?.immediate_next_steps),
success_metrics: safeJoin($json.outputResult?.success_metrics),
decision_timeline: $json.outputResult?.decision_timeline || '',

// Business Case
opportunity_size: $json.outputResult?.business_case?.opportunity_size || '',
competitive_advantage: $json.outputResult?.business_case?.competitive_advantage || '',
resource_investment: $json.outputResult?.business_case?.resource_investment || '',
expected_roi: $json.outputResult?.business_case?.expected_roi || '',
strategic_alignment: $json.outputResult?.strategic_alignment || '',

// Risk Management
risk_mitigation_strategies: safeJoin($json.outputResult?.risk_mitigation_plan?.map(r =>
`#${r.risk}: ${r.mitigation}`)),
high_priority_risks: safeJoin($json.outputResult?.risk_mitigation_plan?.filter(r => r.impact ===
'High')?.map(r => r.risk)),

// Input Constraints & Context
original_constraints: $json.constraints || '',
timeline_preference: $json.timelinePreference || '',

// Analysis Metadata
total_features_identified: (($json.roadmapResult?.mvp_features?.length || 0) +
    ($json.roadmapResult?.phase2_features?.length || 0) +
    ($json.roadmapResult?.phase3_features?.length || 0)),
analysis_confidence: $json.priorityResult?.confidence_level || 'Medium',

// Formatted Summaries for Quick Review
```

```

quick_summary: `${$json.priorityResult?.priority_tier || 'Medium'} Priority (Score: ${$json.priorityResult?.calculated_score || 0}) - ${$json.outputResult?.recommendation || 'Review'}`,
roadmap_summary: $json.roadmapResult?.roadmap_summary || 'Roadmap analysis pending',
overall_assessment: `Market: ${$json.marketResult?.attractiveness_score || 0}/10, Priority: ${$json.priorityResult?.priority_tier || 'Medium'}, Recommendation: ${$json.outputResult?.recommendation || 'Review'}`

};

// Add workflow metadata
finalOutput.workflow_version = '2.0';
finalOutput.agent_model = 'gemini-2.0-flash-exp';
finalOutput.processing_status = 'Complete';

return [{ json: finalOutput }];

```

INPUT

Parameters

Mode

Language

JavaScript

```

1 // Generate unique ID for this analysis session
2 const ideaId = `IDEA_${Date.now()}.${Math.random().toString(36).substr(2, 5)}`;
3 const analysisDate = new Date().toISOString().split('T')[0];
4 const analysisTime = new Date().toISOString().split('T')[1].split('.')[0];
5
6 // Helper function to safely join arrays
7 const safeJoin = (arr, separator = ',') => {
8   if (!arr || !Array.isArray(arr)) return '';
9   return arr.map(item =>
10     typeof item === 'object' ? JSON.stringify(item) : String(item)
11   ).join(separator);
12 };
13
14 // Helper function to extract nested object values
15 const extractNestedData = (obj, defaultValue = '') => {
16   if (!obj || typeof obj !== 'object') return defaultValue;

```

Type \$ for a list of special vars/methods. Debug by using console.log() statements and viewing their output in the browser console.

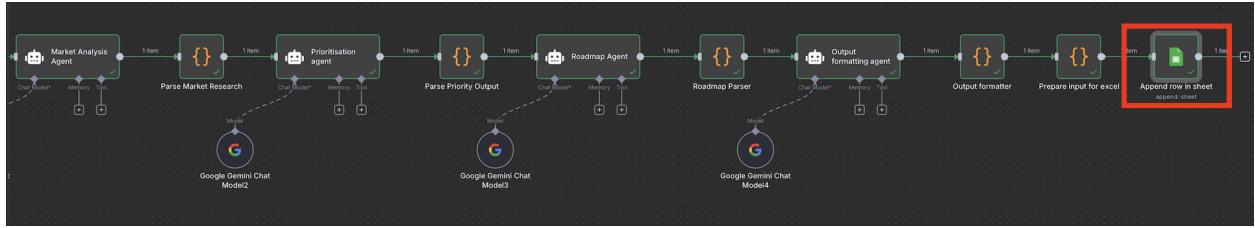
OUTPUT

To make sure expressions after this node work, return the input items that produced each output item. [More info](#)

idea_id	analysis_date
IDEA_1762769605899_0ki9e	2025-11-10

I wish this node would...

Step 8b - Output sheet node



Step 9: Write to Google Sheets

9.1 Google Sheets Output Node

1. Add "Google Sheets" node
2. Operation: "Append"
3. Select your output spreadsheet
4. Map all fields from the previous node

INPUT

- Prepare input for excel
- idea_id: IDEA_1762769605899_0ki9e
- analysis_date: 2025-11-10
- analysis_time: 10:13:25
- original_idea
- refined_idea
- value_proposition
- target_audience
- business_context
- alternative_ideas
- core_problems
- key_features
- innovation_score: 0
- market_summary
- market_size
- market_attractiveness_score: 0
- market_growth_rate
- market_timing
- competitors

Parameters

Credential to connect with: Google Sheets account 2
Resource: Sheet Within Document
Operation: Append Row
Document: From list - AI Agent - Ideation and Product ...
Sheet: From list - Output Sheet
Mapping Column Mode: Map Each Column Manually
Values to Send:

idea_id	refined_idea	market_analysis	priority_score
IDEA_1762769605899_0ki9e	Despite a highly attractive market for secure, on-demand school transport in Indian metro cities, we recommend rejecting the current product concept due to an extremely high operational and reputational risk profile, combined with an unacceptably low 5% success probability. This decision prevents substantial potential financial and brand damage associated with direct liability in child transport.	The market presents a compelling opportunity (Market Attractiveness: 8/10) driven by increased parental safety concerns and demand for tech-enabled convenience in a largely unorganized sector.; The core challenge lies in the exceptionally high-stakes nature of child transport; any perceived or actual safety incident poses catastrophic reputational and financial risks, making the business inherently fragile.; Current project feasibility is critically low (Success Probability: 5%), with an aggressive MVP timeline (3-4 months) being	0

I wish this node would...

- internal audit (logs): introduces an extremely high level of strict
- extensive testing/training: acquisition/conversion, immediately (Owner: Product Leadership)

