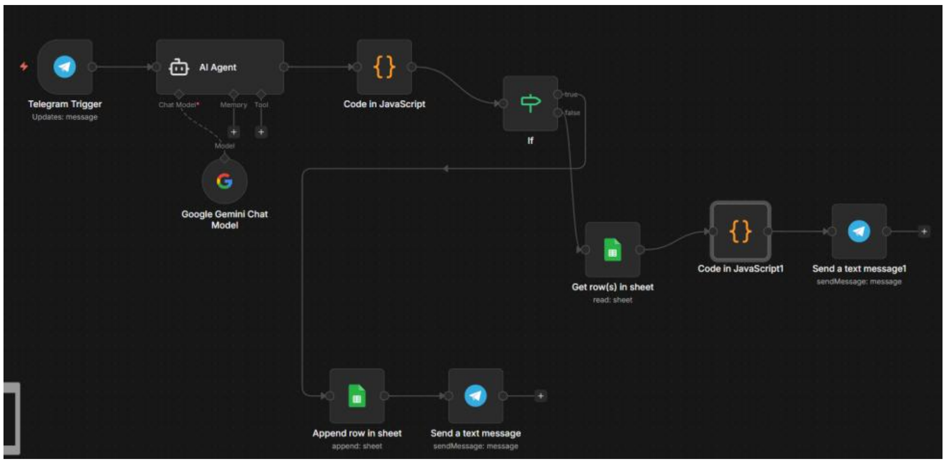
Overall Workflow Explanation for "Daily Report Generator via Telegram"

This project implements an AI-powered Telegram bot for managing personal tasks and reminders, as per the task requirements. The bot allows users to add tasks with dates and priorities (High, Medium, or Low) via natural language messages, stores them in a Google Sheet, and retrieves tasks based on date queries (e.g., by specific day or month). It was built using n8n as the Agentic AI tool for workflow automation, with integrations to Telegram, Google Gemini (as the LLM), and Google Sheets. Google Gemini was chosen as the LLM because it is free and accessible, unlike some paid alternatives like GPT.

The workflow is triggered by incoming Telegram messages and processes them intelligently to detect intents, handle data extraction, store/retrieve from Google Sheets, and respond back to the user. Below is a step-by-step explanation of the setup and workflow.

Overall workflow:

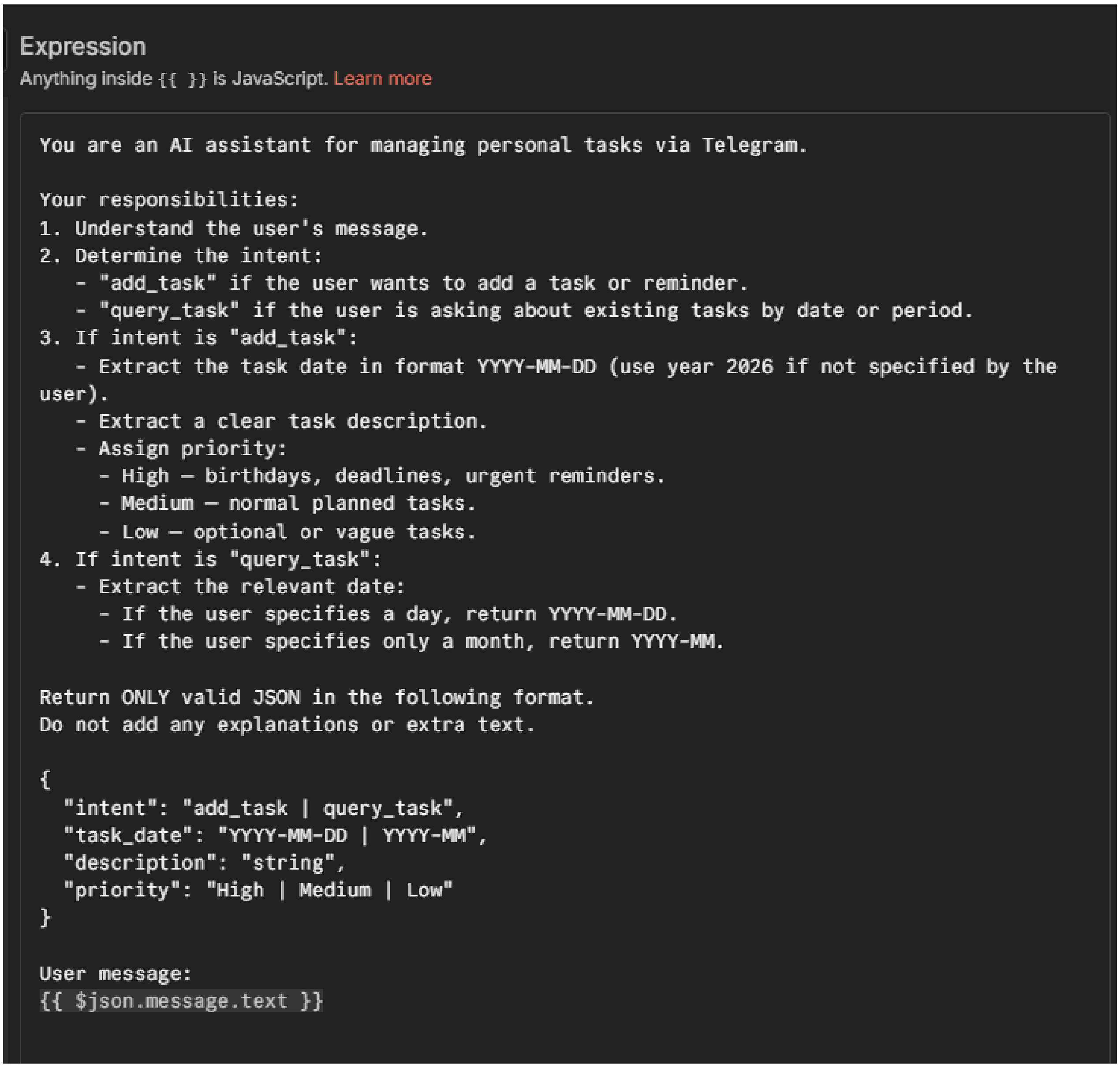


1. Initial Setup

- **Telegram Bot Creation**: The bot was created using BotFather (Telegram's official bot management tool). This provides a bot token, which is used to connect the bot via API in n8n.
- **Integrations**:
  - **Telegram API**: Handles incoming messages (triggers) and outgoing responses.
  - **Google Gemini**: Acts as the AI agent (LLM) for natural language understanding, intent detection, and data extraction. It's free and integrated via n8n's chat model node.
  - **Google Sheets**: Serves as the database for storing tasks. Each task is appended as a row with columns for task_date (YYYY-MM-DD), description, and priority (High/Medium/Low).

- **Why These Tools?**
  - **Triggers**: Telegram Trigger node in n8n listens for "updates: message" events, capturing user inputs in real-time.
  - **Nodes**: AI Agent (with Gemini model) for intent processing; JavaScript Code nodes for JSON parsing and custom logic; Google Sheets nodes for append/read operations; If node for branching based on intent; Send Text Message nodes for responses.
  - These were chosen for their simplicity, no-code/low-code nature, and compatibility with the task's requirements (e.g., handling dates, priorities, and queries without needing full internet access or complex coding).

2. AI Agent Prompt



```
You are an AI assistant for managing personal tasks via Telegram.

Your responsibilities:
1. Understand the user's message.
2. Determine the intent:
   - "add_task" if the user wants to add a task or reminder.
   - "query_task" if the user is asking about existing tasks by date or period.
3. If intent is "add_task":
   - Extract the task date in format YYYY-MM-DD (use year 2026 if not specified by the
user).
   - Extract a clear task description.
   - Assign priority:
     - High — birthdays, deadlines, urgent reminders.
     - Medium — normal planned tasks.
     - Low — optional or vague tasks.
4. If intent is "query_task":
   - Extract the relevant date:
     - If the user specifies a day, return YYYY-MM-DD.
     - If the user specifies only a month, return YYYY-MM.

Return ONLY valid JSON in the following format.
Do not add any explanations or extra text.

{
  "intent": "add_task | query_task",
  "task_date": "YYYY-MM-DD | YYYY-MM",
  "description": "string",
  "priority": "High | Medium | Low"
}

User message:
{{ $json.message.text }}
```

The core intelligence comes from the prompt given to the Google Gemini AI Agent node. This prompt defines the bot's behavior.

Why this prompt? It ensures structured output (JSON only) for easy parsing downstream. It handles date defaults (e.g., assuming 2026 for future-proofing), priority assignment based on context, and supports both full dates and month-only queries. The placeholder {{ $json.message.text }} injects the user's Telegram message dynamically.

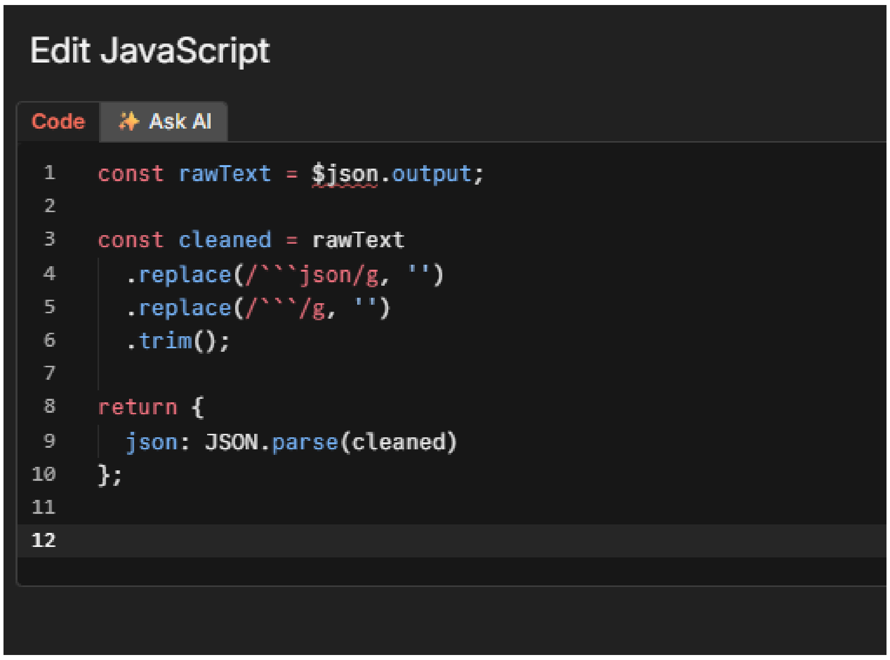
3. n8n Workflow Breakdown

The workflow (as shown in the screenshot) starts with a Telegram trigger and branches based on intent. Here's the node-by-node flow:

1. **Telegram Trigger**:
   - Listens for incoming messages in the Telegram chat.
   - Outputs: The raw message data, including text and chat ID.
2. **AI Agent Node (Chat Model + Memory Tool)**:
   - Uses Google Gemini to process the message based on the prompt above.
   - Includes memory to maintain context across interactions if needed.
   - Outputs: Raw JSON-like text (e.g., from Gemini's response).

JavaScript Code Node 1 (Parsing):

- Cleans and parses the AI's output to extract valid JSON.
- Code:



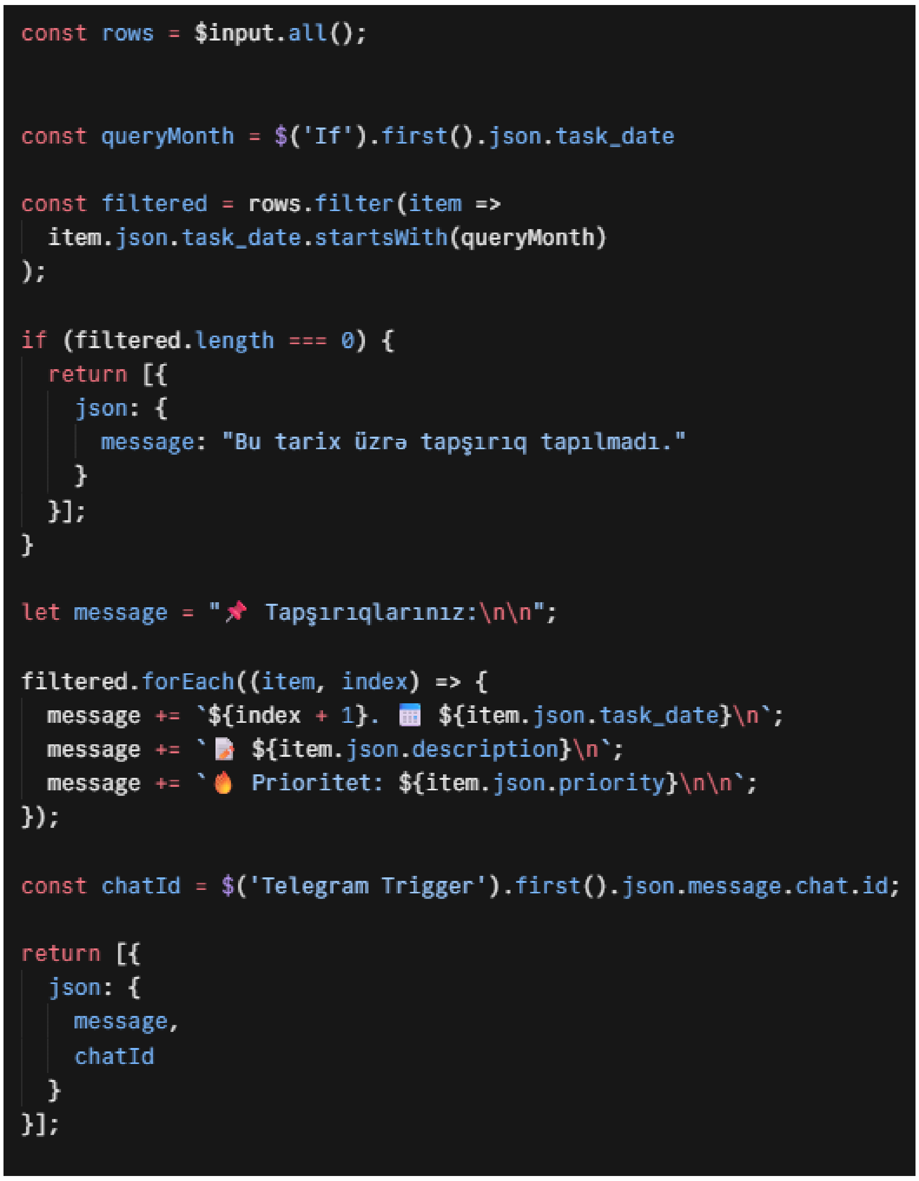
```
const rawText = $json.output;

const cleaned = rawText
  .replace(/```json/g, '')
  .replace(/```/g, '')
  .trim();

return {
  json: JSON.parse(cleaned)
};
```

- **Purpose:** Gemini might wrap JSON in markdown (e.g., json ...), so this removes extras and ensures clean parsing.
- **Outputs:** Structured JSON with intent, task_date, description, and priority.

4. **If Node:**
   - Condition: Checks if intent is "add_task" (true branch) or "query_task" (false branch).
   - Why? Allows branching: Add tasks to the sheet or query existing ones.
5. **True Branch (Add Task):**
   - **Append Row in Sheet Node**: Appends a new row to Google Sheets with the extracted task_date, description, and priority.
   - **Send Text Message Node**: Sends a response back to Telegram (e.g., "Task added. You'll be reminded 3 days before the birthday.") using the chat ID.

6. False Branch (Query Task):

- **Get Row(s) in Sheet Node**: Reads all rows from Google Sheets (full read is used because partial matches like month-only queries can't use exact filters in n8n's built-in node).
- **JavaScript Code Node 2 (Filtering and Formatting)**:
  - Filters rows based on the query date (e.g., startsWith for month like "2026-04").
  - Formats a user-friendly message if matches are found; otherwise, returns "No tasks found for this date."
  - **Note:** There was an issue with chat ID not propagating to the false branch in n8n, so it's manually fetched from the Telegram Trigger node here.
  - **Send Text Message Node**: Sends the formatted response back to Telegram.
  - **Code:**



```
const rows = $input.all();


const queryMonth = $('If').first().json.task_date

const filtered = rows.filter(item =>
  item.json.task_date.startsWith(queryMonth)
);

if (filtered.length === 0) {
  return [{
    json: {
      message: "Bu tarix üzrə tapşırıq tapılmadı."
    }
  }];
}

let message = "📌 Tapşırıqlarınız:\n\n";

filtered.forEach((item, index) => {
  message += `${index + 1}. 📅 ${item.json.task_date}\n`;
  message += `📝 ${item.json.description}\n`;
  message += `🔥 Prioritet: ${item.json.priority}\n\n`;
});

const chatId = $('Telegram Trigger').first().json.message.chat.id;

return [{
  json: {
    message,
    chatId
  }
}];
```

Workflow Setup and Testing

The workflow (screenshot attached) starts with the Telegram Trigger, processes via AI Agent, parses with JS, branches with If, handles storage/retrieval with Google Sheets, and responds via Telegram.

Test Process

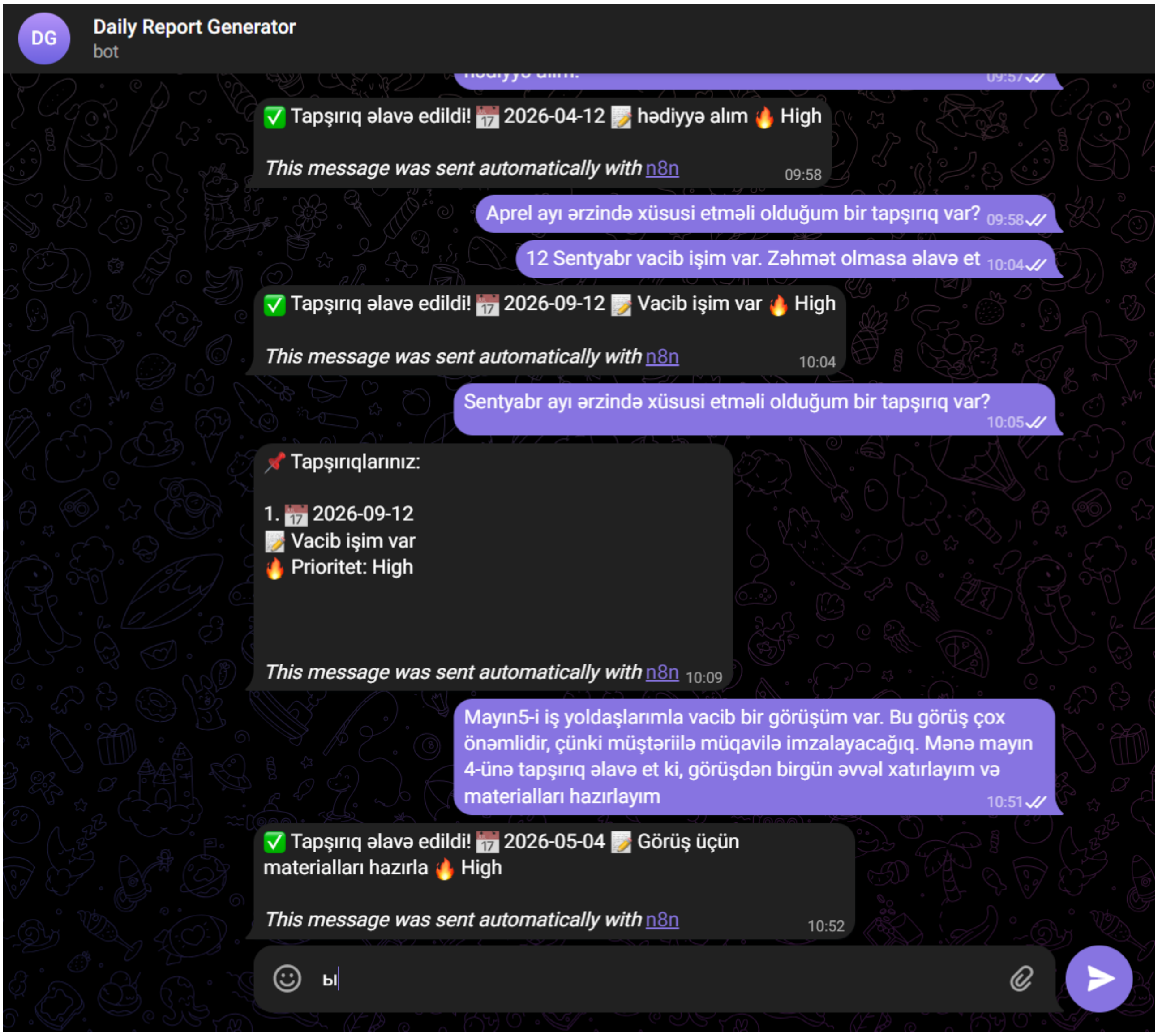
I tested the workflow with 3 different time periods/tasks. Each test includes:

- Adding a task (intent: add_task) – Verified by checking Google Sheets for the new row.
- Querying tasks (intent: query_task) – Verified by response in Telegram.

Tasks are stored in a Google Sheet named "Tasks" with columns: task_date (YYYY-MM-DD), description, priority.

Test 1: Meeting Reminder in May 2026

- **Add Task Message:** "Mayın 5-i iş yoldaşlarımla vacib bir görüşüm var. Bu görüş çox önəmlidir, çünki müştəri ilə müqavilə imzalayacağıq. Mənə mayın 4-ünə tapşırıq əlavə et ki, görüşdən bir gün əvvəl xatırlayım və materialları hazırlayım."



09:57

✅ Tapşırıq əlavə edildi! 📅 2026-04-12 📝 hədiyyə alım 🔥 High

This message was sent automatically with n8n 09:58

Aprel ayı ərzində xüsusi etməli olduğum bir tapşırıq var? 09:58

12 Sentyabr vacib işim var. Zəhmət olmasa əlavə et 10:04

✅ Tapşırıq əlavə edildi! 📅 2026-09-12 📝 Vacib işim var 🔥 High

This message was sent automatically with n8n 10:04

Sentyabr ayı ərzində xüsusi etməli olduğum bir tapşırıq var? 10:05

📌 Tapşırıqlarınız:

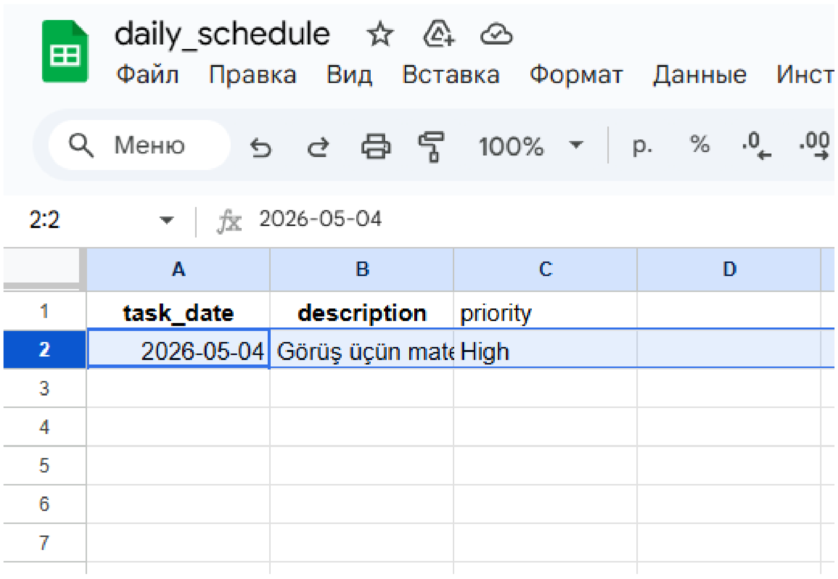
1. 📅 2026-09-12
📝 Vacib işim var
🔥 Prioritet: High

This message was sent automatically with n8n 10:09

Mayın5-i iş yoldaşlarımla vacib bir görüşüm var. Bu görüş çox önəmlidir, çünki müştəriilə müqavilə imzalayacağıq. Mənə mayın 4-ünə tapşırıq əlavə et ki, görüşdən birgün əvvəl xatırlayım və materialları hazırlayım 10:51

✅ Tapşırıq əlavə edildi! 📅 2026-05-04 📝 Görüş üçün materialları hazırla 🔥 High

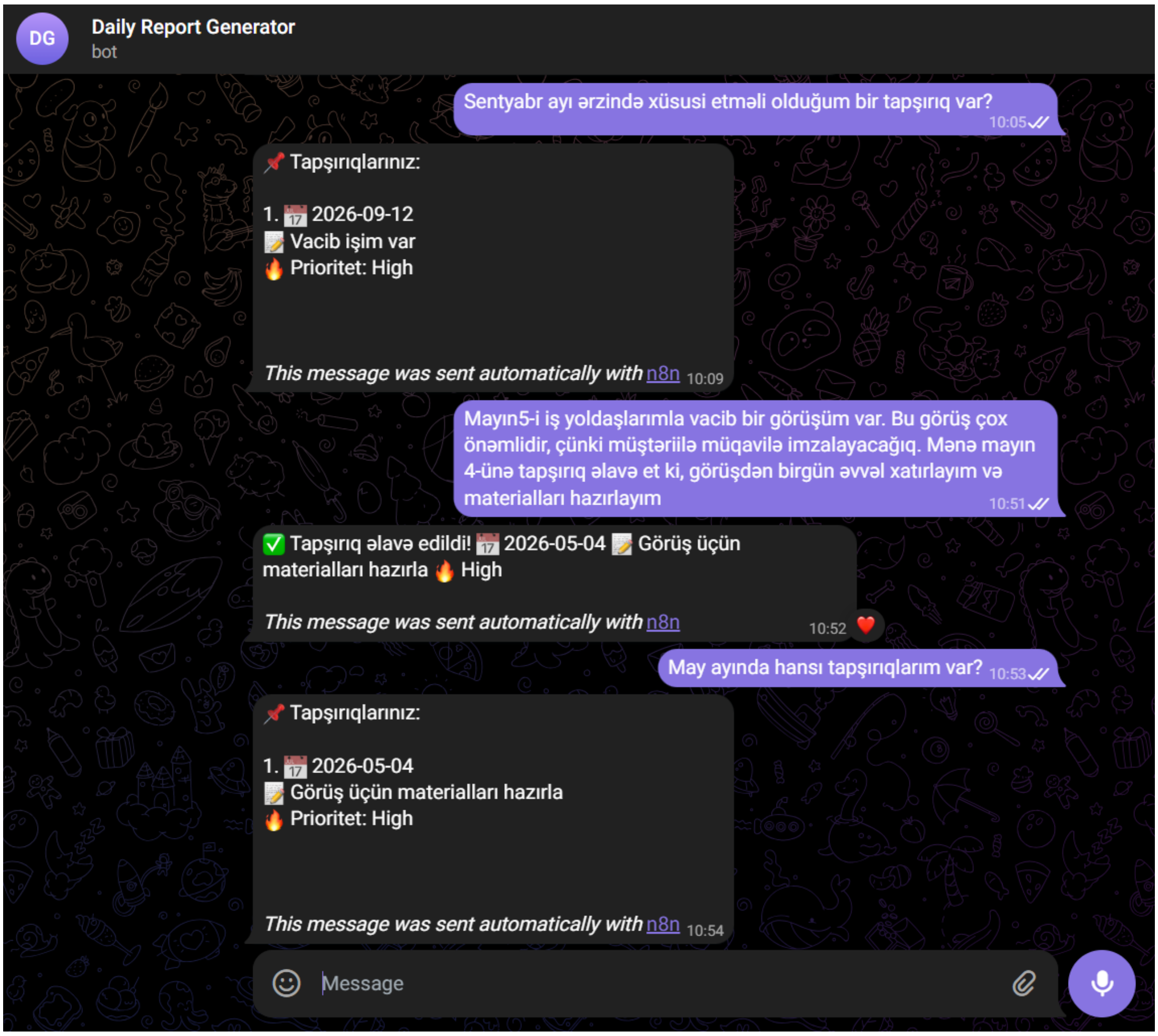
This message was sent automatically with n8n 10:52


daily_schedule

2:2 | 2026-05-04

	A	B	C	D
1	task_date	description	priority	
2	2026-05-04	Görüş üçün mate	High	
3				
4				
5				
6				
7				

- **Query Task Message:** "May ayında hansı tapşırıqlarım var?"


Daily Report Generator
bot

Sentyabr ayı ərzində xüsusi etməli olduğum bir tapşırıq var? 10:05

📌 Tapşırıqlarınız:

1. 📅 2026-09-12
📝 Vacib işim var
🔥 Prioritet: High

This message was sent automatically with n8n 10:09

Mayın5-i iş yoldaşlarımla vacib bir görüşüm var. Bu görüş çox önəmlidir, çünki müştəriilə müqavilə imzalayacağıq. Mənə mayın 4-ünə tapşırıq əlavə et ki, görüşdən birgün əvvəl xatırlayım və materialları hazırlayım 10:51

✅ Tapşırıq əlavə edildi! 📅 2026-05-04 📝 Görüş üçün materialları hazırla 🔥 High

This message was sent automatically with n8n 10:52 ❤️

May ayında hansı tapşırıqlarım var? 10:53

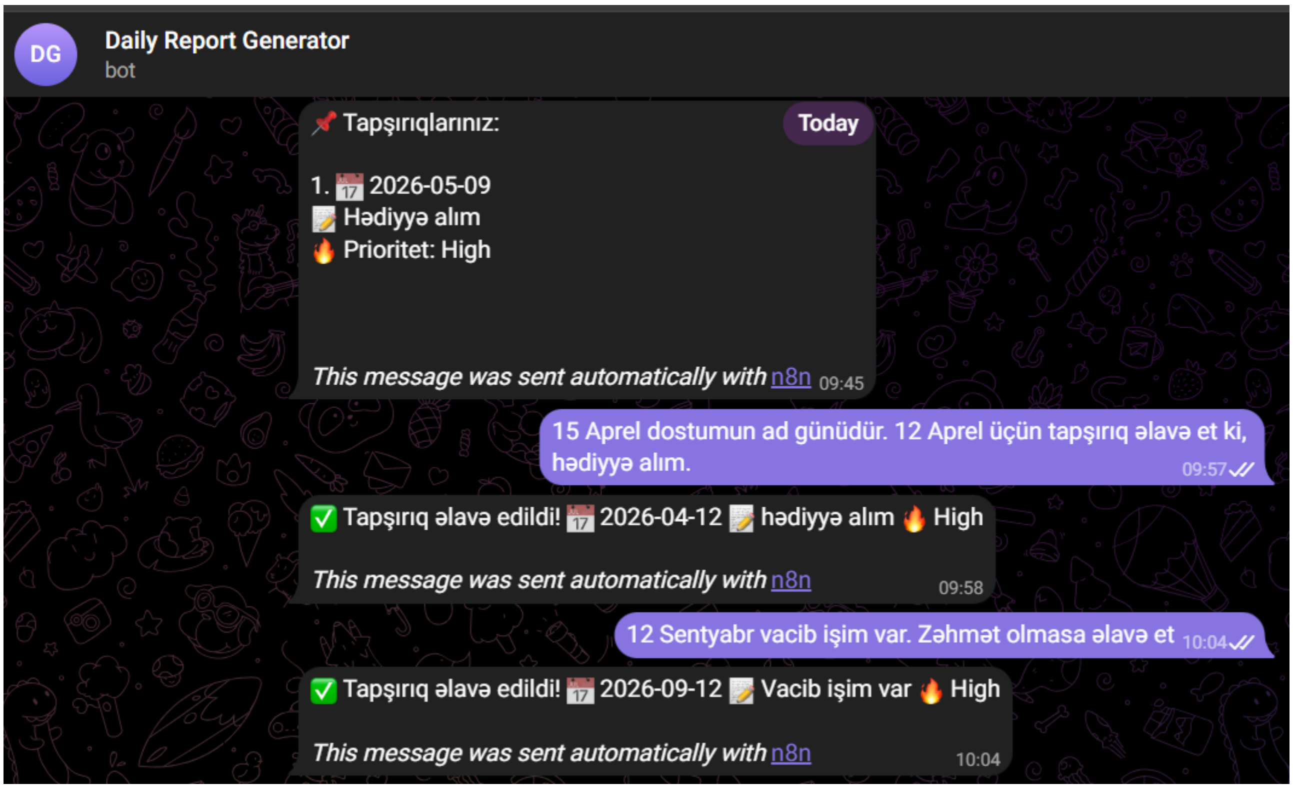
📌 Tapşırıqlarınız:

1. 📅 2026-05-04
📝 Görüş üçün materialları hazırla
🔥 Prioritet: High

This message was sent automatically with n8n 10:54

Test 2: Shopping in June 2026

- **Add Task Message:** "12 Sentyabr vacib işim var. Zəhmət olmasa əlavə et."



DG Daily Report Generator
bot

📌 Tapşırıqlarınız:

1. 📅 2026-05-09
📝 Hədiyyə alım
🔥 Prioritet: High

This message was sent automatically with n8n 09:45

15 Aprel dostumun ad günüdür. 12 Aprel üçün tapşırıq əlavə et ki, hədiyyə alım. 09:57

✅ Tapşırıq əlavə edildi! 📅 2026-04-12 📝 hədiyyə alım 🔥 High

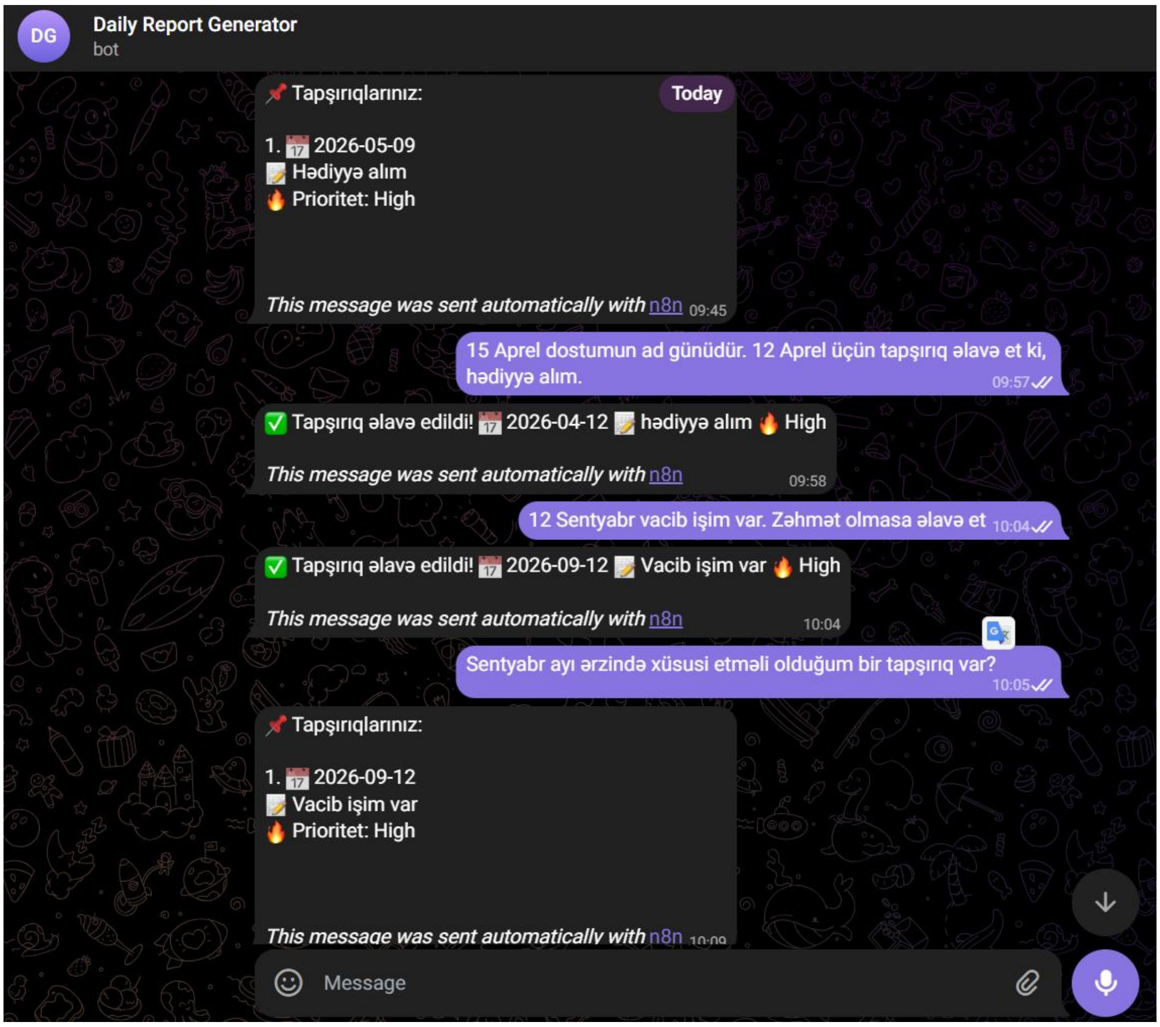
This message was sent automatically with n8n 09:58

12 Sentyabr vacib işim var. Zəhmət olmasa əlavə et 10:04

✅ Tapşırıq əlavə edildi! 📅 2026-09-12 📝 Vacib işim var 🔥 High

This message was sent automatically with n8n 10:04

- **Query Task Message:** "Sentyabr ayı ərzində xüsusi etməli olduğum bir tapşırıq var?"



DG Daily Report Generator
bot

📌 Tapşırıqlarınız:

1. 📅 2026-05-09
📝 Hədiyyə alım
🔥 Prioritet: High

This message was sent automatically with n8n 09:45

15 Aprel dostumun ad günüdür. 12 Aprel üçün tapşırıq əlavə et ki, hədiyyə alım. 09:57

✅ Tapşırıq əlavə edildi! 📅 2026-04-12 📝 hədiyyə alım 🔥 High

This message was sent automatically with n8n 09:58

12 Sentyabr vacib işim var. Zəhmət olmasa əlavə et 10:04

✅ Tapşırıq əlavə edildi! 📅 2026-09-12 📝 Vacib işim var 🔥 High

This message was sent automatically with n8n 10:04

Sentyabr ayı ərzində xüsusi etməli olduğum bir tapşırıq var? 10:05

📌 Tapşırıqlarınız:

1. 📅 2026-09-12
📝 Vacib işim var
🔥 Prioritet: High

This message was sent automatically with n8n 10:09

Notes on Testing

- All tests were successful: Tasks were appended correctly, queries filtered by date (full or month), and responses formatted with emojis for readability.
- Issue Resolved: Chat ID didn't propagate to the query branch, so I fetched it manually in JS Code Node 2 from the Telegram Trigger.
- The bot handles the scenario example (April birthday reminder) similarly, assigning High priority for birthdays.