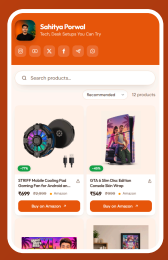
FEATURED CREATOR PRODUCT

Create Your Bio Store with "Store Card"

Use Your Bio Potential Properly — showcase your products, links, and affiliate picks in one clean, high-converting bio storefront.

Launch Your Store at StoreCard.in →



COMPLETE STEP-BY-STEP AUTOMATION PLAYBOOK • ENGLISH EDITION

n8n Gym Manager Automation: Complete Setup Guide

Google Sheets, automated WhatsApp renewal reminders, instant payment status updates, and daily owner reports — everything you need to build and sell a complete n8n gym workflow system.

Blog: BLOGGINGIFY.COM | YouTube: @sahitya.porwal | Instagram: @sahityaporwal | Bio Store: StoreCard.in

Looking For Website? Build a High-Converting Site with IconicBits

Contact: +91 8290133885 | **Portfolio:** iconicbits.in/projects/

Table of Contents

00 Introduction, Target Audience & How to Use This Guide 2

01 Chapter 1: The Promise & Complete System Overview 3

02 Chapter 2: Required Tools, Accounts & Cost Breakdown 4

03 Chapter 3: Designing the Google Sheet Database & Dashboard 6

04 Chapter 4: Workflow 1 — Automatic New Member Entry 8

05 Chapter 5: Workflow 2 — Pre-Expiry WhatsApp Reminders 10

06 Chapter 6: Workflow 3 — Automatic Payment Status Updates 12

07 Chapter 7: Workflow 4 — Daily Owner Summary Report 14

08 Chapter 8: Testing, Error Handling & Data Privacy 16

09 Chapter 9: Pitching, Pricing & Selling to Gym Owners 17

10 Implementation Checklist, Quick Reference, Mistakes, FAQs & Sources 19

START HERE

Introduction

This guide has one focused goal: helping you build a working **Gym Manager Automation** in n8n for small and medium gyms—and package it as a paid setup service for local gym owners. You will build four practical automations step by step: saving new members directly to Google Sheets, sending pre-expiry WhatsApp reminders, updating payment status automatically, and delivering a daily summary report to the gym owner.

There is no fluff or abstract theory here. Every chapter explains what each component does, why it matters, and how it works, followed by exact setup steps and ready-to-use copy-paste code and formulas.

Who This Guide Is For

- **Beginners:** Anyone who has never built an automation before but knows how to open and edit a Google Sheet.
- **Freelancers & Agencies:** Creators and consultants looking to offer practical automation setups to local businesses such as gyms, coaching centers, and salons.
- **Gym Owners & Managers:** Business owners who want to set up their own automated member tracking and renewal system in-house.

Prior coding experience is not required. The small JavaScript snippets and Google Sheets formulas included in this guide are ready to copy and paste directly.

What You Will Learn

- Designing a clean, automation-ready Google Sheet for gym members, fees, and expiry tracking.
- Building 4 production n8n workflows: New Member Entry, Expiry Reminder, Payment Update, and Daily Report.
- Understanding WhatsApp messaging options, official pricing trade-offs, and template approval rules.
- Testing edge cases, setting up error alerts, and protecting member data privacy.
- Running a 10-minute live demo, drafting a clear proposal, and structuring a realistic setup fee (for example, ₹10,000–₹15,000).

How to Use This Guide

Follow the chapters in sequence, because every workflow builds on the Google Sheet created in Chapter 3. Start by creating a dummy gym with 5 to 10 test members, run every workflow end to end, and only demo the system to a real gym owner once everything passes cleanly. Whenever pricing, plan limits, or platform policies are mentioned, check the linked official source page before quoting a client, as platform terms change periodically.

Important Disclaimer

This guide does not guarantee income, clients, or specific business results. The ₹10,000 to ₹15,000 setup fee is a realistic example benchmark that varies depending on your city, skill level, demo quality, and the size of the gym.

CHAPTER 1 The Promise & Complete System Overview

In one sentence, Gym Manager Automation works like this: store member records in Google Sheets, let n8n check the sheet automatically every day, and automate the repetitive tasks the owner currently does by hand—reminders, payment status updates, and daily reporting.

What This System Delivers

Core Feature / Promise	Where to Build It
New member details save automatically to Google Sheets upon joining	**Chapter 4** (Workflow 1)
Automated WhatsApp renewal reminders sent before membership expiry	**Chapter 5** (Workflow 2)
Payment status and next expiry date update automatically upon payment	**Chapter 6** (Workflow 3)
Daily owner report showing active, expired, and pending payment counts	**Chapter 7** (Workflow 4)
Positioning, demoing, and charging ₹10,000–₹15,000 for the setup	**Chapter 9** (Sales Guide)

The Real Problem Gym Owners Face

In a typical gym with 100 to 150 members, every member has a different joining date, membership plan, fee amount, and renewal date. Tracking all of this in a paper register or manual spreadsheet creates two costly problems: it consumes valuable time every day, and missed renewals directly cost the owner revenue. Automation solves this exact bottleneck, making its business value immediately obvious.

How the System Works

The architecture uses three core components: **Google Sheets** (the database), **n8n** (the automation engine), and **WhatsApp** (the member communication channel). Together, they power four independent workflows:

1. **Workflow 1 — New Member Entry:** Captures a new member via an online registration form or webhook and appends a formatted row to Google Sheets.
2. **Workflow 2 — Expiry Reminder:** Reads the sheet every morning, identifies members whose expiry is approaching, and sends an approved WhatsApp template message.

3. **Workflow 3 — Payment Update:** Listens for payment confirmation from Razorpay (or a manual Cash/UPI form), marks the member as "Paid," and extends their expiry date.
4. **Workflow 4 — Daily Report:** Runs at a scheduled time daily, counts active, expired, and pending-payment members, and sends a concise summary to the owner via Email or WhatsApp.

Verified vs. Recommended vs. Subject to Change

- **Verified (Official Docs):** n8n includes native Schedule Trigger, Google Sheets Trigger, and WhatsApp Business Cloud nodes (with the "Send Template" operation). Razorpay Payment Links supports the `payment_link.paid` webhook event.
- **Recommended (Suggested Architecture):** The 12-column sheet layout, status naming conventions, reminder wording, and 4-workflow split can all be customized to fit each gym.
- **Subject to Change:** n8n subscription tiers, WhatsApp per-message rates, and Meta policies evolve over time—always verify current pricing on the official pages linked in the Sources section.

CHAPTER 2 Required Tools, Accounts & Cost Breakdown

To run the complete system, you need four components: an n8n instance, a Google account, a WhatsApp sending route, and (optionally) an online payment platform. For a small gym, the primary recurring costs are n8n hosting and WhatsApp message charges.

1. n8n: Cloud vs. Self-Hosted

What it is: n8n is a visual workflow automation platform where you connect nodes (action blocks) to automate tasks across apps. **Why use it:** It provides ready-made Google Sheets, Schedule, Webhook, and WhatsApp nodes, minimizing custom code.

Option	Best For	Approximate Cost (Verify Officially)
n8n Cloud Starter	Beginners who do not want to manage a server	~€20/month (annual billing) or €24/month; 2,500 executions/month
n8n Cloud Pro	Higher execution volume or managing multiple clients	~€50/month (annual billing) or €60/month; 10,000 executions/month
Self-Hosted Community Edition	Users comfortable running and maintaining a VPS	Software is free; pay only for a small VPS (a few dollars/month)

Note that n8n Cloud does not offer a permanent free plan—only a 14-day free trial. The Self-Hosted Community Edition software is free, though server uptime, backups, and updates are your responsibility.

How executions are counted: On n8n Cloud, one complete run of a workflow counts as 1 execution regardless of how many nodes it contains. Workflows 2 and 4 run once per day (~60 executions/month), while Workflows 1 and 3 trigger only on member signups or payments. For a single

gym, 2,500 executions/month is more than enough. If you manage 5 to 10 gyms, consider n8n Pro or self-hosting.

2. Google Account

You need a Google account to host the Google Sheet. Always create or use an account owned by the gym owner so they retain full ownership of their data if your engagement ends. Connect Google Sheets to n8n using standard OAuth credentials.

3. WhatsApp: Message Sending Options

Choosing the right WhatsApp route is the most important architectural decision because it governs both cost and compliance.

Option	Key Advantage	Important Trade-Offs
WhatsApp Business Platform (Cloud API) + n8n built-in node	Official Meta route; n8n node natively supports "Send Template"	Requires Meta Business verification and template approval; messages are paid
Business Solution Provider (BSP) e.g., AiSensy	Simpler onboarding and visual dashboard	BSP may charge additional monthly platform fees
Unofficial WhatsApp Web wrappers	Appears cheaper to set up initially	**Not recommended:** Violates WhatsApp terms and risks banning the gym's number

Approximate WhatsApp Cost (India): Meta bills per delivered template message. According to 2026 India pricing guides, Marketing templates averaged around ₹0.86/message and Utility templates around ₹0.115/message (excluding GST). Meta can revise rates quarterly, and service message rules also updated effective October 1, 2026—always calculate final estimates from Meta's official rate card.

Worked Cost Example: Suppose a gym has 130 members, 40 memberships expire in a given month, and each expiring member receives 2 reminders (80 messages total). At the Utility rate (~₹0.115), 80 messages cost roughly **₹9 + GST**. If Meta classifies the template as Marketing (~₹0.86), the same 80 messages cost roughly **₹70 + GST**. Either way, the monthly cost is very small, but knowing the template category upfront prevents billing surprises.

4. Payment Platform (Optional, Used in Workflow 3)

To update payment status automatically, use a payment gateway that dispatches webhooks on successful payments. **Razorpay Payment Links** provides the `payment_link.paid` event for this exact purpose. If a gym only accepts cash or UPI screenshots, Chapter 6 covers a clean manual-confirmation alternative using an n8n form.

Chapter 2 Action Checklist

- [ ] Start your 14-day n8n Cloud trial (or install Self-Hosted Community Edition on a VPS).
- [ ] Create a dedicated Google account and set up a new spreadsheet named "Gym Manager."

- [ ] Begin WhatsApp Business Platform setup (requires a Meta Business account and a dedicated phone number).
- [ ] Verify your Razorpay account status if the gym wants automated online payment tracking.

CHAPTER 3 Designing the Google Sheet Database & Dashboard

The Google Sheet serves as your entire database. Build it cleanly first using two separate tabs: **Members** (where n8n reads and writes records) and **Dashboard** (formula-driven summary metrics that n8n never modifies).

Tab 1: Members

Enter these exact 11 column headings in Row 1 with identical spelling and lowercase underscores. Because n8n maps fields by column name, avoid renaming headers after setup.

Column Name	Purpose & Format	Updated By
`member_id`	Unique identifier, e.g., `GYM-001`	n8n (Workflow 1)
`name`	Member's full name	n8n (Workflow 1)
`phone`	WhatsApp number with country code, no `+` or spaces (e.g., `919876543210`)	n8n (Workflow 1)
`plan`	Plan label: Monthly, Quarterly, Half-yearly, Yearly	n8n (Workflow 1)
`plan_months`	Numeric duration: `1`, `3`, `6`, or `12`	n8n (Workflow 1)
`fee`	Plan fee amount in INR	n8n (Workflow 1)
`start_date`	Membership start date in `YYYY-MM-DD` format	n8n (Workflow 1)
`expiry_date`	`start_date + plan_months` in `YYYY-MM-DD`	n8n (Workflows 1 & 3)
`payment_status`	`Pending` or `Paid`	n8n (Workflows 1 & 3)
`last_payment_date`	Date of the most recent confirmed payment	n8n (Workflow 3)
`last_reminder_date`	Date the last WhatsApp reminder was sent	n8n (Workflow 2)

Crucial Date Formatting Rule

Store all dates as plain text in `YYYY-MM-DD` format (for example, `2026-12-15`). Google Sheets reformats native dates differently across regional locales, which breaks date comparisons in n8n. Select all date columns and set **Format → Number → Plain text**.

Sample Test Rows (CSV Format)

SAMPLE DATA — PASTE INTO MEMBERS TAB FOR TESTING

```
member_id,name,phone,plan,plan_months,fee,start_date,expiry_date,payment_status,
last_payment_date,last_reminder_date
GYM-001,Rohit Kumar,919000000001,Monthly,1,1200,2026-11-01,2026-12-01,Paid,2026-
11-01,
GYM-002,Neha Singh,919000000002,Quarterly,3,3000,2026-09-10,2026-12-
10,Paid,2026-09-10,
GYM-003,Amit Verma,919000000003,Monthly,1,1200,2026-10-05,2026-11-05,Pending,,
```

Always replace the phone numbers above with your own number or a friend's number during testing. Never send test reminders to real gym members.

Tab 2: Dashboard (Formulas Only)

Create a second tab named **Dashboard** so the owner gets an instant overview whenever they open the spreadsheet. Assuming `expiry_date` is in Column H and `payment_status` is in Column I of the `Members` tab, add these four formulas:

Metric Label	Google Sheets Formula
Total members	`=COUNTA(Members!A2:A)`
Active members	`=SUMPRODUCT((Members!H2:H<>"")*(Members!H2:H>=TEXT(TODAY(),"yyyy-mm-dd")))`
Expired members	`=SUMPRODUCT((Members!H2:H<>"")*(Members!H2:H<TEXT(TODAY(),"yyyy-mm-dd")))`
Pending payments	`=COUNTIF(Members!I2:I,"Pending")`

The Active and Expired formulas use `SUMPRODUCT` because ISO text strings (`YYYY-MM-DD`) sort chronologically when compared alphabetically. If a count looks wrong, verify that the date columns are formatted as Plain text.

Chapter 3 Action Checklist

- [ ] Create the `Members` tab with all 11 exact column headers.
- [ ] Set all date columns to Plain text format.

- [ ] Add the 3 sample test rows using your own phone number.
- [ ] Create the `Dashboard` tab, enter all 4 formulas, and verify the counts match.

CHAPTER 4 Workflow 1: Automatic New Member Entry

This workflow captures new member details from an online registration form, calculates the exact expiry date, cleans the phone number, and saves the row into the `Members` tab. The gym front desk can keep this form bookmarked on a tablet or phone.

n8n Form Trigger → **Code (Format & Expiry)** → **Google Sheets (Append or Update Row)**

These three nodes are all you need. Optionally, you can attach a `WhatsApp (Send Template)` node at the end to deliver an automatic welcome message.

Step-by-Step Setup

1. Create a new workflow in n8n and name it **"1 - New Member Entry"**.
2. Add a **Form Trigger** node. Set the form title to `Gym Registration` and add these fields: **Name** (text), **Phone** (text), **Plan** (dropdown: Monthly, Quarterly, Half-yearly, Yearly), **Fee** (number), and **Start Date** (date).
3. Add a **Code** node, set mode to `Run Once for All Items`, and paste the JavaScript snippet below.
4. Add a **Google Sheets** node. Select operation `Append or Update Row`, choose your `Gym Manager` spreadsheet and `Members` tab, set mapping mode to `Map Each Column Manually`, and set **Column to match on** to `phone`. Matching on phone prevents duplicate rows if someone submits twice.
5. Map every Google Sheet column to its corresponding Code node output field (`member_id`, `name`, `phone`, `plan`, `plan_months`, `fee`, `start_date`, `expiry_date`, `payment_status`).
6. Test the workflow by submitting a test member through the form and confirming the new row appears in your Sheet.
7. Publish/activate the workflow. Copy the **Production URL** (separate from the Test URL) and bookmark it on the gym's front-desk device.

Code Node Snippet (Copy-Paste)

This script normalizes 10-digit Indian phone numbers to the `91XXXXXXXXXX` format, maps the plan name to months, and computes the expiry date accurately—including month-end edge cases (such as January 31 + 1 month = February 28).

JAVASCRIPT — WORKFLOW 1 CODE NODE

```
const planMonths = { 'Monthly': 1, 'Quarterly': 3, 'Half-yearly': 6, 'Yearly':
12 };
```

```
function addMonths(isoDate, months) {
  const [y, m, d] = isoDate.split('-').map(Number);
  const total = (m - 1) + months;
  const ny = y + Math.floor(total / 12);
  const nm = total % 12;
  const lastDay = new Date(Date.UTC(ny, nm + 1, 0)).getUTCDate();
  const nd = Math.min(d, lastDay);
  return ny + '-' + String(nm + 1).padStart(2, '0') + '-' +
String(nd).padStart(2, '0');
}

return $input.all().map(item => {
  const f = item.json;
  let phone = String(f['Phone']).replace(/\D/g, '');
  if (phone.length === 10) phone = '91' + phone;
  const months = planMonths[f['Plan']] || 1;
  const start = String(f['Start Date']).slice(0, 10);
  return {
    json: {
      member_id: 'GYM-' + String(Date.now()).slice(-6),
      name: f['Name'],
      phone: phone,
      plan: f['Plan'],
      plan_months: months,
      fee: f['Fee'],
      start_date: start,
      expiry_date: addMonths(start, months),
      payment_status: 'Pending'
    }
  };
});
```

Custom field labels: If you name a form field differently (for example, "Full Name"), update `f['Name']` in the code to match the exact form label.

Migrating existing members: Copy the gym's existing 100–150 members from Excel once and paste them directly into the `Members` tab using the same 11 columns and `YYYY-MM-DD` date format. Workflow 1 handles new signups going forward.

Alternative trigger: If the owner prefers typing rows directly into Google Sheets, use the `Google Sheets Trigger` node (events: Row added, Row updated, or Row added or updated) to trigger welcome messages or downstream steps.

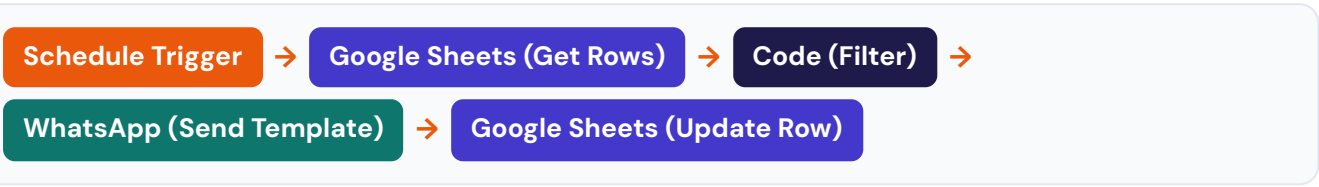
Chapter 4 Action Checklist

- [ ] Connect the Form Trigger, Code, and Google Sheets nodes in Workflow 1.
- [ ] Submit a test entry and verify that `expiry_date` calculates accurately.

- [ ] Submit the same phone number twice to confirm no duplicate row is created.
- [ ] Activate the workflow and save its Production Form URL.

CHAPTER 5 Workflow 2: Pre-Expiry WhatsApp Reminders

This workflow runs automatically every morning, reads the `Members` tab, identifies members whose membership expires in 3 days or today, sends an approved WhatsApp template message, and logs today's date in `last_reminder_date`.



Step 0: Create and Approve Your WhatsApp Template

Business-initiated WhatsApp messages (when a member has not messaged the business in the last 24 hours) require a pre-approved template in Meta's WhatsApp Manager. Example template copy:

> Hello {{1}}, your gym membership expires on {{2}}. Please visit the front desk or reply to us to renew your plan. Thank you!

Here, `{{1}}` is the member's name and `{{2}}` is their expiry date. Submit the template under the **Utility** category since it updates an existing membership. Note that Meta makes the final category decision and may reclassify templates as Marketing (higher rate)—avoid adding discounts or promotional offers inside the reminder text.

Opt-in compliance: Always collect member consent before sending WhatsApp updates. Add a checkbox to your Chapter 4 registration form (e.g., *"I agree to receive membership updates on WhatsApp"*) and message only opted-in members.

Step-by-Step Setup

1. Create a new workflow named **"2 - Expiry Reminder"**.
2. Add a **Schedule Trigger** node set to interval `Days` at `9:00 AM`. Set the workflow timezone to `Asia/Kolkata` in Workflow Settings so reminders fire at the exact local hour. Remember: scheduled workflows run automatically only when published/activated.
3. Add a **Google Sheets** node with operation `Get Row(s)` on the `Members` tab with no filters (fetch all rows).
4. Add a **Code** node and paste the Filter Code below.
5. Add a **WhatsApp** node: Resource `Message`, Operation `Send Template`. Select your sender number, set recipient to `{{ $json.phone }}`, pick your approved template, and pass body parameters in exact order: first `{{ $json.name }}`, then `{{ $json.expiry_date }}`. Parameter mismatches trigger an "invalid parameter" error.

6. Add a **Google Sheets (Update Row)** node: match on `phone` and set `last_reminder_date` to `{{ $now.setZone('Asia/Kolkata').toFormat('yyyy-MM-dd') }}`.
7. To test, set one test row's expiry date to 3 days from today and click **Execute workflow** manually.

Filter Code Snippet (Copy-Paste)

This script passes forward only members whose expiry is 3 days away or today and who have not already received a reminder today. If nobody matches, downstream nodes do not execute.

JAVASCRIPT — WORKFLOW 2 FILTER CODE NODE

```
const REMIND_DAYS = [3, 0]; // Days before expiry to send reminders
const today = new Date().toLocaleDateString('en-CA', { timeZone: 'Asia/Kolkata'
});

function toUtc(iso) {
  const [y, m, d] = iso.split('-').map(Number);
  return Date.UTC(y, m - 1, d);
}

return $input.all().filter(item => {
  const r = item.json;
  if (!r.expiry_date || !r.phone) return false;
  if (r.last_reminder_date === today) return false;
  const daysLeft = Math.round((toUtc(r.expiry_date) - toUtc(today)) / 86400000);
  return REMIND_DAYS.includes(daysLeft);
});
```

The `en-CA` locale is used intentionally because it outputs dates in strict `YYYY-MM-DD` format, matching your Google Sheet.

Choosing Reminder Timing

- **3 days before + expiry day (`[3, 0]`):** Ideal cadence for most gyms—gives members time to renew without feeling spammed.
- **Avoid over-messaging:** Sending more than 2 reminders per cycle annoys members and risks spam reports that degrade your WhatsApp phone number quality rating.

Chapter 5 Action Checklist

- [ ] Submit your WhatsApp reminder template in Meta WhatsApp Manager and wait for approval.
- [ ] Connect all 5 nodes in Workflow 2 and set timezone to `Asia/Kolkata`.
- [ ] Test with your own phone number and confirm the WhatsApp message arrives cleanly.
- [ ] Run the workflow a second time the same day to confirm no duplicate reminder is sent.

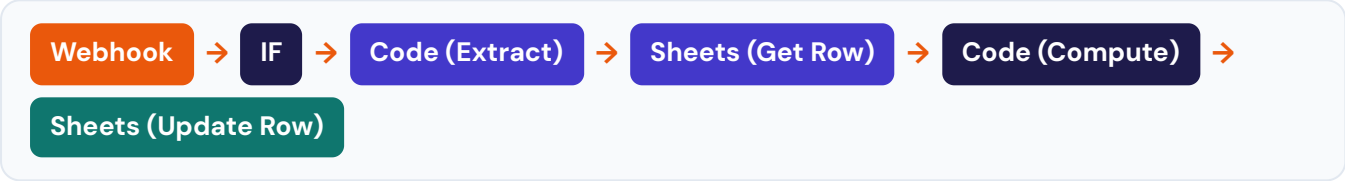
CHAPTER 6 Workflow 3: Automatic Payment Status Updates

Whenever a member pays their fee, this workflow marks `payment_status` as `Paid`, records `last_payment_date`, and extends `expiry_date` on renewals. You can implement either **Path A** (online Razorpay webhook) or **Path B** (manual Cash/UPI confirmation form)—both share the exact same update logic.

First: Add a 12th Column to Google Sheets

Add a 12th column header named `last_payment_ref` to the `Members` tab. Storing the payment reference ID makes the workflow idempotent—if a payment gateway retries the same webhook event twice, the member's expiry date will not be extended twice.

Path A: Razorpay Payment Link (Online Payments)



How it works: When generating a Razorpay Payment Link for a member, set its `reference_id` equal to the member's `member_id` from Google Sheets (e.g., `GYM-123456`). Upon payment, Razorpay sends a `payment_link.paid` webhook containing `reference_id`, allowing n8n to locate and update the exact row.

1. Create a new workflow named **"3 - Payment Update"**. Add a **Webhook** node (Method: `POST`) and copy its Production URL.
2. In the Razorpay Dashboard, go to **Account & Settings → Webhooks**, create a new webhook with your Production URL, and subscribe strictly to the `payment_link.paid` event.
3. **Set a Webhook Secret & Verify Signature:** Never go live without signature verification—otherwise anyone could POST a fake request to mark a member Paid. Enable `Raw Body` in the Webhook node options, compute the HMAC SHA256 hash using an n8n **Crypto** node against the raw request body (not parsed JSON), and compare it against Razorpay's signature header (see official Razorpay webhook docs in Sources).
4. Add an **IF** node checking that the event equals `payment_link.paid`.
5. Add a **Code** node named `Extract Payment` and paste the first snippet below.
6. Add a **Google Sheets (Get Row(s))** node filtered where column `member_id` equals `{{ $json.member_id }}`.
7. Add a second **Code** node (Compute) and paste the second snippet below. On a member's first payment, it keeps the initial expiry calculated by Workflow 1; on renewals, it extends from the existing expiry date (if still active) or from today (if already expired).
8. Add a **Google Sheets (Update Row)** node: match on `member_id` and map `payment_status`, `last_payment_date`, `expiry_date`, and `last_payment_ref`.
9. Test end-to-end using Razorpay Test Mode with a dedicated test member row (never test on a real member).

JAVASCRIPT — NODE 1: "EXTRACT PAYMENT"

```
const b = $input.first().json.body;
const e = b.payload.payment_link.entity;
return [{ json: {
  event: b.event,
  member_id: e.reference_id,
  link_id: e.id,
  amount_paid: e.amount_paid / 100 // Razorpay returns amounts in paise
}}];
```

JAVASCRIPT — NODE 2: COMPUTE NEW EXPIRY & DEDUPLICATE

```
function addMonths(isoDate, months) {
  const [y, m, d] = isoDate.split('-').map(Number);
  const total = (m - 1) + months;
  const ny = y + Math.floor(total / 12);
  const nm = total % 12;
  const lastDay = new Date(Date.UTC(ny, nm + 1, 0)).getUTCDate();
  const nd = Math.min(d, lastDay);
  return ny + '-' + String(nm + 1).padStart(2, '0') + '-' +
String(nd).padStart(2, '0');
}

const row = $input.first().json;
const pay = $('Extract Payment').first().json;
const today = new Date().toLocaleDateString('en-CA', { timeZone: 'Asia/Kolkata'
});

if (row.last_payment_ref === pay.link_id) return []; // Already processed

const firstPayment = !row.last_payment_date;
const base = row.expiry_date > today ? row.expiry_date : today;
const newExpiry = firstPayment ? row.expiry_date : addMonths(base,
Number(row.plan_months));

return [{ json: {
  member_id: row.member_id,
  payment_status: 'Paid',
  last_payment_date: today,
  expiry_date: newExpiry,
  last_payment_ref: pay.link_id
}}];
```

Note: Payment funds settle directly into the gym's own Razorpay account, never through yours.

Path B: Cash or UPI (Owner Manual Confirmation)

If the gym prefers cash or UPI screenshots, build a **Form Trigger** named `Record Payment` with fields for Member Phone, Amount, and Payment Mode (Cash / UPI). On submission, n8n looks up the member by phone and runs the same Compute and Update Row nodes (replacing `pay.link_id` with `'manual-' + Date.now()`). The owner never has to manually edit cells in the spreadsheet.

Chapter 6 Action Checklist

- [ ] Add the 12th column `last_payment_ref` to the `Members` tab.
- [ ] Implement either Path A (Razorpay webhook) or Path B (Record Payment form).
- [ ] Run a test payment and verify status turns `Paid` and expiry extends properly.
- [ ] Replay the same webhook event once to confirm the expiry date does not double-extend.

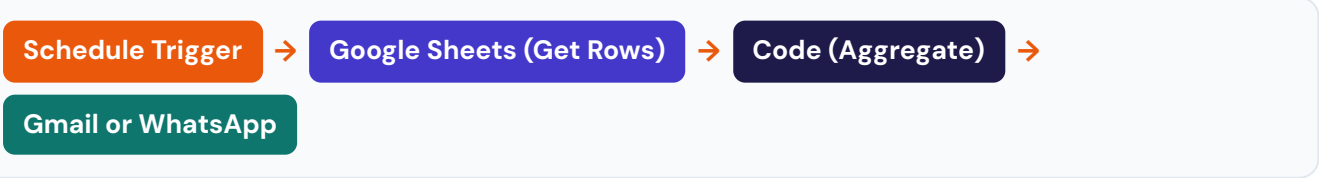
CHAPTER 7 Workflow 4: Daily Owner Summary Report

This workflow reads the `Members` tab at a fixed time every day, calculates active, expired, and pending-payment counts, and sends an executive summary to the gym owner—eliminating the need to open registers or spreadsheets.

Standardizing the Three Metric Definitions

Agree on these exact definitions with the gym owner upfront so manual spot-checks always match the automated report:

Metric	Business Meaning	Google Sheet Rule
Active	Membership is currently valid	`expiry_date` is today or later
Expired	Membership ended; renewal due	`expiry_date` is earlier than today
Pending payments	Fee not yet marked received	`payment_status` equals `Pending`



Step-by-Step Setup

1. Create a new workflow named **"4 - Daily Report"**. Add a **Schedule Trigger** for `8:00 AM` (or `9:00 PM`) daily in timezone `Asia/Kolkata`.
2. Add a **Google Sheets (Get Row(s))** node to fetch all rows from the `Members` tab.
3. Add a **Code** node and paste the aggregation snippet below.

4. Attach either the Email or WhatsApp delivery node (detailed below) and publish/activate the workflow.

JAVASCRIPT — WORKFLOW 4 DAILY REPORT AGGREGATOR

```
const today = new Date().toLocaleDateString('en-CA', { timeZone: 'Asia/Kolkata'
});
const rows = $input.all().map(i => i.json).filter(r => r.member_id);

const active  = rows.filter(r => r.expiry_date >= today);
const expired = rows.filter(r => r.expiry_date && r.expiry_date < today);
const pending = rows.filter(r => r.payment_status === 'Pending');

const names = list => list.length ? list.map(r => r.name).join(', ') : 'None';

const text =
  'Gym Report ' + today + '\n' +
  'Active members: ' + active.length + '\n' +
  'Expired memberships: ' + expired.length + ' (' + names(expired) + ')\n' +
  'Pending payments: ' + pending.length + ' (' + names(pending) + ')';

return [{ json: {
  date: today,
  active: active.length,
  expired: expired.length,
  pending: pending.length,
  text
}}];
```

Two Delivery Methods

- **Method 1 — Email (Recommended to Start):** Connect a `Gmail` (or Send Email) node, set recipient to the owner's email, subject to `Gym Report {{ $json.date }}`, and body to `{{ $json.text }}`. It includes full member name lists, costs nothing extra, and needs no WhatsApp template approval.
- **Method 2 — WhatsApp:** Requires an approved single-line template (WhatsApp template parameters disallow newline characters):

  > Gym Report {{1}}: Active {{2}}, Expired {{3}}, Pending payments {{4}}. View full member list in your Google Sheet Dashboard tab.

  Pass parameters in order: `{{ $json.date }}`, `{{ $json.active }}`, `{{ $json.expired }}`, `{{ $json.pending }}`.

Chapter 7 Action Checklist

- [ ] Confirm the three count definitions with the gym owner.

- [ ] Test Workflow 4 against 3 sample rows (active, expired, pending) and verify counts manually.
- [ ] Set the daily delivery schedule to the owner's preferred time.

CHAPTER 8 Testing, Error Handling & Data Privacy

Before handing the system to a paying client, deliberately try to break every workflow. A single wrong reminder or false Paid status quickly erodes client trust.

Proper Testing Protocol

1. **Use a dedicated test Sheet:** Run the entire system on a dummy spreadsheet containing only your and your friends' phone numbers.
2. **Test 5 critical edge cases:** (1) a January 31 join date, (2) submitting the same phone number twice, (3) an already-expired member making a renewal payment, (4) a duplicate webhook event firing twice, and (5) a row with a blank phone number.
3. **Manual Execute before scheduling:** Click "Execute workflow" manually on scheduled workflows and inspect the JSON output of every node before relying on the timer.
4. **Always deploy Production URLs:** Form and Webhook nodes have separate Test and Production URLs. Give the client only Production URLs and verify every workflow is switched Active/Published.

Automated Error Monitoring

Create a dedicated **Error Handler** workflow starting with an `Error Trigger` node connected to an Email or WhatsApp alert to yourself showing which workflow failed. Then open Settings on all 4 production workflows and assign this workflow as their **Error Workflow**.

Enable **"Continue on Fail"** selectively—specifically on the WhatsApp node in Workflow 2, so one member's invalid phone number never blocks reminders for the remaining members.

Common Errors & Quick Fixes

Problem	Root Cause	Solution
WhatsApp node returns "invalid parameter"	Template parameter order or count mismatch, or blank value	Match n8n parameters to template order and provide fallback values
Reminder never delivered	Phone contains `+` /spaces, or member never opted in	Use Chapter 4 phone-cleaning code and verify opt-in consent
Scheduled workflow did not run	Workflow not published/activated or wrong timezone	Activate workflow and set timezone to `Asia/Kolkata`
Dashboard counts are wrong	Date columns formatted as locale dates instead of plain text	Convert all date columns to Plain text in `YYYY-MM-DD`

Problem	Root Cause	Solution
Incorrect expiry date	Month-end rollover bug or mismatched plan spelling	Use Chapter 4 `addMonths()` helper and exact plan labels

Data Privacy & WhatsApp Compliance

- **Keep data in the owner's account:** Store the Google Sheet inside the gym owner's Google Drive and share only with authorized staff. Never enable public link sharing.
- **Collect explicit consent:** Obtain written or form-based opt-in for WhatsApp messages and immediately remove anyone who opts out.
- **Data minimization:** Collect only what the system needs (name, phone, plan, fee, dates)—never store sensitive health records or ID proofs in this sheet.
- **Legal compliance (India):** Personal data in India is governed by the **Digital Personal Data Protection Act, 2023 (DPDP Act)**. Consult a qualified lawyer for formal privacy notice and consent wording tailored to your client.
- **Follow Meta's policies:** Use accurate template categories and avoid spammy messaging frequencies that harm number quality.

Chapter 8 Action Checklist

- [ ] Run all 5 edge cases on your test Google Sheet and confirm clean behavior.
- [ ] Build the Error Trigger workflow and trigger one deliberate failure to verify alerts arrive.
- [ ] Audit Google Sheet sharing permissions (ensure public link sharing is OFF).
- [ ] Add the WhatsApp opt-in consent checkbox to the client's registration form.

CHAPTER 9 Pitching, Pricing & Selling to Gym Owners

The ₹10,000 to ₹15,000 setup fee is a practical benchmark—not a guaranteed rate. Your actual fee depends on your city, your demo presentation, the size of the gym, and the polish of your delivery. Here is how to present the offer clearly and professionally.

What You Are Actually Selling

A gym owner is not buying n8n or Google Sheets. They are buying four tangible business outcomes: **hours saved every week**, **recovered renewal revenue**, **crystal-clear pending payment visibility**, and **professional WhatsApp communication**. Lead every conversation with those outcomes, not technical jargon.

The 10-Minute Live Demo Script

1. Open your dummy Google Sheet in front of the owner (populated with 5 sample members).
2. Fill out the front-desk registration form live and show the new member row appearing automatically in the sheet.

3. Set one sample member's expiry date to 3 days from today, run Workflow 2, and show the WhatsApp reminder arriving on your phone in real time.
4. Trigger a test payment and show the status flip to `Paid` with the expiry date extending automatically.
5. Show the Daily Report email/message and ask: *"How long does it currently take you to pull these exact numbers every day?"*

Sample Pricing Structure

Component	Example Pricing	What to Clarify in Writing
One-time setup fee	₹10,000 – ₹15,000	Includes Sheet design, 4 workflows, WhatsApp template setup, testing, and 30 minutes of staff training
Monthly maintenance (optional)	Small fixed monthly retainer	Covers error monitoring, minor tweaks, and bug fixes; document separately
Running costs	Billed directly to client	n8n hosting/VPS, Meta WhatsApp message fees, and payment gateway charges

Keep running accounts in the client's name: Always register n8n, Google, and WhatsApp Business accounts under the gym owner's credentials or transfer full ownership. It builds trust and ensures you never carry recurring third-party billing liabilities.

Copy-Paste Proposal Template

Gym Manager Automation — Project Proposal

Client: [Gym Name]
Included Scope:
1. Automated new member registration saved to Google Sheets
2. Automated pre-expiry WhatsApp reminders (via approved Meta template)
3. Automated payment status and renewal expiry updates
4. Daily owner summary report (active, expired, and pending payments)

Not Included: Website, mobile app, biometric/attendance hardware integration, or items outside the list above.
One-Time Setup Fee: ₹[Amount]
Delivery Timeline: [X] days after client accounts and WhatsApp template approval are ready.
Client Responsibilities: WhatsApp Business verification, existing member spreadsheet data, and a test phone number.
Running Costs: Paid directly via client's own accounts.
Post-Launch Support: Free bug fixes for [X] days after handover.

Honest Expectations

- WhatsApp Business verification and template approval can take several days and are outside your control—set this expectation upfront.
- Your first client build will take the longest; your second and third gym deployments will be dramatically faster because you can reuse the Sheet structure and n8n workflow templates.
- Always deliver a clean handover document listing all accounts, credentials, and API keys.
- Start by perfecting your process with 1 to 2 local gyms before scaling outreach.

Chapter 9 Action Checklist

- [ ] Practice your 10-minute live demo once in front of a friend.
- [ ] Customize the proposal template with your name, pricing, and support terms.
- [ ] List 5 local gyms in your city and request a 10-minute live demo with each owner.

TOOLKIT Complete Implementation Checklist & Quick Reference

Master Pre-Launch Checklist (12 Items)

- [ ] n8n Cloud trial or Self-Hosted Community Edition is live and accessible.
- [ ] Google Sheet has the `Members` tab (12 columns) and formula-driven `Dashboard` tab.
- [ ] All date columns are formatted as Plain text in `YYYY-MM-DD`.
- [ ] Workflow 1 (New Member Entry) is tested, including duplicate phone number handling.
- [ ] WhatsApp template is approved and a live reminder was received on a test number.
- [ ] WhatsApp opt-in consent checkbox is included on the registration form.
- [ ] Workflow 3 (Payment Update) passed a test payment and verified duplicate events do not double-extend expiry.
- [ ] Razorpay webhook signature verification is active (if using Path A).
- [ ] Workflow 4 daily report counts match manual counts on the spreadsheet.
- [ ] Error Trigger workflow is configured and linked in every production workflow's Settings.
- [ ] All 4 workflows are published/activated with timezone set to `Asia/Kolkata`.
- [ ] Client has received account ownership and the complete handover document.

Quick-Reference Node Guide

Task	n8n Node	Essential Setting
Capture member signup	Form Trigger	Bookmark Production URL on front-desk device

Task	n8n Node	Essential Setting
Run on daily schedule	Schedule Trigger	Set timezone to `Asia/Kolkata` and activate workflow
Read / update database	Google Sheets	Match on `phone` or `member_id` (Append or Update Row)
Send renewal reminder	WhatsApp (Send Template)	Use approved template with exact parameter order
Receive payment event	Webhook	Subscribe to `payment_link.paid` + verify HMAC signature
Date & expiry math	Code	Keep all dates in `YYYY-MM-DD` plain text
Failure alerts	Error Trigger	Assign in Settings of every production workflow

8 Common Mistakes to Avoid

1. **Testing on real members:** Always use your own phone number during development.
2. **Leaving default Sheet date formatting:** Always switch date columns to Plain text.
3. **Sending free-form WhatsApp reminders:** Business-initiated messages outside the 24-hour customer window require approved templates.
4. **Putting promotional offers in reminder templates:** Triggers Marketing category pricing instead of Utility rates.
5. **Skipping webhook signature verification:** Leaves your payment endpoint vulnerable to spoofed requests.
6. **Forgetting to activate/publish workflows:** Draft workflows never run on schedule or production URLs.
7. **Keeping client accounts under your own email:** Always give clients ownership of their Google, n8n, and Meta accounts.
8. **Skipping a written scope:** Prevents scope creep (like unplanned websites, apps, or biometric hardware).

Frequently Asked Questions (FAQs)

Do I need to know coding? No. All JavaScript snippets in this guide are copy-paste ready—just ensure form field names match your labels.

Can reminders be sent without the WhatsApp API? Yes, via Email or SMS. However, WhatsApp reminders require the official WhatsApp Business Platform to avoid account bans.

How many n8n executions does one gym need? Workflows 2 and 4 use ~60 executions/month combined; Workflows 1 and 3 depend on signups and payments. For most single gyms, the Starter plan (2,500 executions/month) is plenty.

What if the gym owner insists on Excel? Show them Google Sheets works just like Excel in any browser and on their mobile phone, and paste their existing Excel data straight into the `Members` tab.

Is ₹10,000–₹15,000 guaranteed? No—it is a realistic example benchmark. Your actual fee depends on your location, pitch, and client size.

4-Week Implementation Action Plan

Timeline	Weekly Focus
Week 1	Chapters 2 & 3: Set up accounts, design the Google Sheet, and submit WhatsApp Business verification
Week 2	Chapters 4 & 5: Build and test Workflow 1 (Member Entry) and Workflow 2 (Expiry Reminders)
Week 3	Chapters 6 & 7: Build and test Workflow 3 (Payment Updates) and Workflow 4 (Daily Report)
Week 4	Chapters 8 & 9: Complete edge-case testing, rehearse your 10-minute demo, and pitch 5 local gyms

Conclusion

Gym Manager Automation is a focused system of four practical workflows that automates a gym owner's most repetitive daily tasks: member onboarding, renewal reminders, payment tracking, and daily reporting. Build your working demo first, verify every edge case, and present it cleanly as a value-driven service. Always double-check official pricing and policy links before finalizing client quotes.

Official & Secondary Sources (As of November 13, 2026)

- n8n WhatsApp Business Cloud Node (Official Docs) — https://docs.n8n.io/integrations/builtin/app-nodes/n8n-nodes-base.whatsapp/
- n8n Google Sheets Trigger Node (Official Docs) — https://docs.n8n.io/integrations/builtin/trigger-nodes/n8n-nodes-base.googlesheetstrigger/
- Razorpay Payment Links Webhook Events (Official Docs) — https://razorpay.com/docs/payments/payment-links/subscribe-to-webhooks
- Razorpay Payment Links Webhook Payloads (Official Docs) — https://razorpay.com/docs/webhooks/payment-links
- Meta WhatsApp Business Platform Pricing (Official) — https://developers.facebook.com/docs/whatsapp/pricing
- n8n Pricing 2026 Summary (Secondary Source) — https://dupple.com/learn/n8n-pricing
- n8n Pricing Comparison (Secondary Source) — https://www.jetadmin.io/blog/n8n-pricing/
- WhatsApp Business API India Rates (Secondary Source) — https://montymobile.com/blogs/whatsapp-business-api-pricing-in-india-inr-rates-gst-and-the-2026-currency-migration-deadline

- WhatsApp Pricing Update Notice (Secondary Source) — https://m.aisensy.com/blog/whatsapp-per-message-pricing-update-effective-january-1-2026/

Looking For Website? Let's Build Your Online Presence

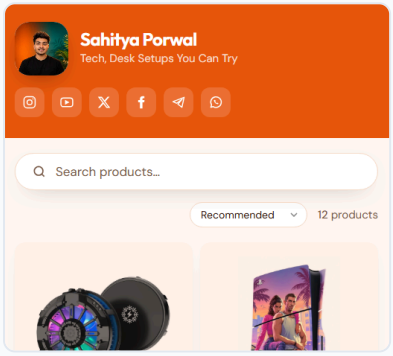
Need a fast, modern, high-converting website for your gym, agency, or business? Reach out directly or explore our recent client projects.

Contact: +91 8290133885

Portfolio: iconicbits.in/projects/

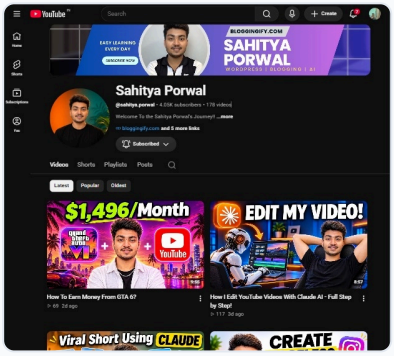
Bio Store: StoreCard.in

Connect with Sahitya Porwal & Explore More Guides



Store Card — Bio Store

storecard.in



YouTube & Blog

@sahitya.porwal • bloggingify.com



Instagram

@sahityaporwal