63LABS

Restauranteur Learn & Build

SECTION 02 OF 16

Business Planning and Money

Business plans, expense and payment records, cash timing, pricing, budgets and calculation practice.



SEPARATE TOPIC PDF / FULL RESTAURANT EDITION PRESERVED

Read this section

Two page-number systems

The section contents below use PDF page positions in this file. Original lesson footers keep their page numbers from the complete 619-page Restaurant Edition. "Master p/pp" still refers to the original 5,410-page Learn & Build source.

Same lessons, smaller files

Content has been separated, not rewritten. Nine shared-boundary pages were divided between their topics so every lesson keeps its ending. The complete Restaurant Edition and original master remain unchanged.

Start with the shared guide

The original scope guide is included below. Project sections also include the shared build starter and spreadsheet setup. These are learning and build specifications, not finished apps or certified operating instructions.

Moving between sections

Local contents and bookmarks open pages in this PDF. Cross-section links use the companion filenames and may require all files in one folder and a supporting PDF reader. If a link does not open, use the numbered section list and the printed edition-page label. Section 01 retains the original full-edition contents and source index.

The 16-section collection

01 Start Here and Source Index

02 Business Planning and Money

03 Stock Suppliers and Food Records

04 Team Operations and Procedures

05 Facilities Equipment and Resources

06 Catering Guests and Delivery

07 Marketing Publishing and Websites

08 Restaurant Products and Services

09 Careers and Team Learning

10 AI Tools and Task Guides

11 Operations Methods and Data

12 Evidence and Reasoning Cases

13 Integrated Restaurant Projects

14 Guest Clubs and Event Ideas

15 Supplier and Product Ideas

16 Concepts and Quality Checks

Contents for this PDF

Numbers on the right are PDF page positions in this section. Retained lesson footers show their complete-edition page labels.

How to use this edition (shared guide) 4

Shared build starter (shared guide) 5

Spreadsheet and document setup (shared guide) 6

Business direction and money 7

P001 Restaurant concept and operating plan 8

P002 Restaurant-industry side service 11

P003 Restaurant management overview 13

P008 Restaurant expense and receipt log 16

P009 Guest receipts, refunds and fees 18

P010 Catering and private-event quotes/invoices 21

P041 Restaurant cash timing forecast 23

P042 Menu-item contribution and price model 25

P043 Restaurant break-even scenario 27

P083 Restaurant project/department budget or owner cash-allocation worksheet 29

P084 Restaurant-owner borrowing statement and hypothetical payment-scenario record 31

P085 Restaurant equipment-replacement or planned-purchase reserve tracker 33

P086 Restaurant supplier, rent, utility, and subscription bill-occurrence calendar 36

Calculator discipline and reusable testing 39

Contribution and markup practice 40

Restaurant-industry service pricing 41

Whole-unit break-even practice 43

Billable-capacity rate planner 44

Catering and service project cost 45

Agreement-based sales commissions 46

Business budget and variance 47

Equipment reserve contribution plan 48

Loan-payment arithmetic practice 49

Restaurant project ROI comparison 51

How to use this edition

This is a restaurant-focused curation of the actual 5,410-page 63Labs Learn & Build combined master. It keeps useful original lessons, projects, worked reasoning and tool guidance, then adds clearly labeled restaurant applications. The original file remains unchanged.

A reference library rather than a course

Start with the problem you are solving. Operators can use the money, stock, supplier, team and guest workflows; restaurant-industry builders can use the service, software, content and commercial project material. You do not need to implement every project or buy every named tool.

Broad restaurant use, not just daily operations

The edition retains all 50 original business projects and 43 adaptable planning, facilities, food-record, event and learning projects. Optional supplier, catering, career, venue-event and creative applications are identified as such. Private health, medication, sleep, fitness, mood and veterinary-care projects are not converted into employee-management tools.

What was changed

Repeated setup instructions and repeated practice layouts are consolidated. Restaurant application paragraphs are new adaptations. Source lessons and project requirements are identified by their original master page numbers. Necessary source corrections are labeled rather than silently presented as original text.

What you receive

A learning and build-specification PDF. It does not include generated apps, connected POS systems, live forms, installed automations, completed spreadsheets, legal policies, or a restaurant-specific food-safety program. The projects explain what to build and how to test it.

Source references

Master p or pp always means the physical page position in 63Labs_Combined_Ready.pdf, not a page in this edition. P001-style labels preserve the original project IDs. The contents and PDF bookmarks navigate this edition; the final index traces its source material.

Keep consequential decisions reviewed

Use fictional data while building. Keep private guest, staff and payment data out of shared examples. Use approved restaurant procedures and current authoritative guidance for food safety, employment, taxes, accessibility, privacy and other regulated matters. This edition does not certify compliance or safe equipment work.

Shared build starter

Use this once per build, followed by either the spreadsheet setup or document setup, then the chosen project specification. These consolidated instructions preserve the common safeguards from the source rather than repeating them in every project.

Copy this starter

Act as a patient build partner. Ask the project intake questions in one message, then wait for my answers. I may answer not sure. Separate my facts, goals, assumptions and research still needed. Never invent business facts, sources, permissions, credentials, customers, results or regulatory requirements. Use fictional sample records. Produce the complete requested artifact, not merely an outline. Explain how to copy, set up, use, test and revise it. Do not claim you created or saved a file unless you actually did. If long, stop at a section boundary and resume when I say continue without omissions. End with a requirement-by-requirement checklist. Do not connect services, publish, send messages, spend money or change live records as a side effect of drafting this build.

Add my restaurant context

My restaurant or industry role: [role]. Intended user: [person or role]. Exact task: [task]. Scope and exclusions: [limits]. Approved sources: [documents and dates]. Tools and access available: [tools]. Currency, locale and units: [values]. Owner and reviewer: [roles]. Deadline or work budget: [limit]. Ask about missing information rather than assuming it.

If a response or formula fails

Continue from the exact point where you stopped; keep the same structure, tab names, column order and references. For a formula issue, inspect the actual formula, cell, expected result, source values, locale, dates, units and copied ranges. Explain the proposed correction and rerun the original test. A polished answer is not proof of correctness.

Spreadsheet and document setup

Spreadsheet setup

For Google Sheets, confirm locale and units. The source examples use English functions, comma separators, real dates displayed YYYY-MM-DD, headers in row 1 and data rows 2:1000. Preserve named tabs and column order. Supply actual tab-separated headers, fictional sample rows and exact paste cells. Leave computed sample cells empty. Give each formula with its exact cell, explanation, fill range and expected result. Keep unused rows blank and distinguish missing inputs from zero. Specify validation, dropdowns, formats, filters, frozen headers and formula protection. Keep raw data tables unmerged. Do not add scripts, paid add-ons or external connections by default. Explain how every related range extends beyond row 1000.

Document setup

Create copy-ready pages with clear headings, useful tables only where appropriate, fill-in fields and a completed fictional example. Keep the reusable blank version separate from the example. Supply the actual wording and templates requested, not instructions to write them later. Name owners, actions, dates and review points. Use approved policy content; for changing rules, cite current primary sources only when verified, otherwise list a precise verification task.

First use

Create a safe copy. Enter the source example and compare every expected value. Test blank inputs, explicit zero, duplicate IDs, refunds/reversals, period boundaries and a changed value. For dates and percentages, inspect the underlying values as well as the display. Save evidence before replacing fictional records with authorized live information.

Limits of the supplied specifications

The arithmetic examples have been reviewed, but this PDF is not an executed application test. Generated formulas, data validation, access controls, links, integrations and regulatory treatment still need testing in the actual implementation. Editorial clarifications identify known source gaps, not a claim that all possible edge cases are covered.

Business direction and money

Use the projects below independently. P numbers are the original master project IDs, retained for traceability. The short source lessons and worked reasoning are followed by project-specific build specifications.

P001 Restaurant concept and operating plan

P002 Restaurant-industry side service

P003 Restaurant management overview

P008 Restaurant expense and receipt log

P009 Guest receipts, refunds and fees

P010 Catering and private-event quotes/invoices

P041 Restaurant cash timing forecast

P042 Menu-item contribution and price model

P043 Restaurant break-even scenario

P083 Restaurant project/department budget or owner cash-allocation worksheet

P084 Restaurant-owner borrowing statement and hypothetical payment-scenario record

P085 Restaurant equipment-replacement or planned-purchase reserve tracker

P086 Restaurant supplier, rent, utility, and subscription bill-occurrence calendar

P001 Restaurant concept and operating plan

Business direction and money

Source project: Business Planning Project. Master pp 1012, 1119-1120, 2840. Restaurant application below is an editorial adaptation.

Restaurant application

Adapt examples and intake to the restaurant format, service area, menu offer, operating capacity and cash constraints.

Source based lesson

Turn a business idea into a written plan with a clear offer, customer, operating model, assumptions, and a 90-day action schedule.

Gather and build

Customer, problem, offer, delivery steps, startup costs. Write a one-page plan; mark each claim as evidence or assumption; choose one assumption to test this week.

Check before relying on it

Can a reader identify who pays, what they receive, and why they would choose it? Interview one prospective customer and revise the problem statement using their words.

Source based practice with fictional inputs

Five interviews: three describe the problem, one requests a quote, none has paid.

Worked result: Evidence supports investigating the problem and preparing one quote; it does not establish paying demand. Separate problem reports, quote requests, and completed purchases.

Try a changed condition: Two quoted jobs are accepted, but capacity supports only one this week.

Check your answer: Schedule one within capacity and agree a later date or narrower scope for the other; acceptance does not create capacity.

Restaurant boundary

Preserve the twelve-section plan, evidence/assumption separation, scenarios, cash reconciliation and 90-day schedule. No invented demand, registrations, permits or profitability.

P001 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake Questions

1. What is the business idea, name if chosen, and current stage?
2. What do you sell, to whom, and what specific problem does it solve?

3. Where will you operate or sell, and who will read this plan?
4. What evidence do you have: interviews, orders, sales, or experience?
5. What alternatives do customers currently use?
6. What price, payment terms, and sales channels are you considering?
7. How much money and weekly time can you commit?
8. What startup purchases, monthly expenses, and delivery costs do you know?
9. Who will do the work, and what equipment or partners are needed?
10. What are your 90-day and 12-month goals and biggest uncertainties?

First summarize my answers and list unresolved assumptions. Then build the plan using the instructions that follow.

Required Plan

1. One-page executive summary: offer, customer, problem, advantage, stage, next milestone, and resources needed. Write this after developing the other sections, but place it first.
2. Business overview: owner responsibilities, delivery area, scope, and goals. Mark any ownership or entity decision as unconfirmed unless I supply it.
3. Customer and problem: one primary segment, buying situation, decision-maker, reason to buy, and evidence gaps.
4. Offer: deliverables, exclusions, delivery time, proposed price, payment timing, and a simple first version to test.
5. Market and alternatives: compare three real alternatives I name. If unknown, give a research worksheet instead of fabricated competitors or market sizes.
6. Customer acquisition: two initial channels, the outreach or content tasks, responsible person, weekly activity targets, and how results will be recorded.
7. Sales process: inquiry through qualification, proposal, payment, delivery, and follow-up. Include a short discovery-question list.
8. Operations: supplies, workflow, capacity, quality checks, customer support, responsibilities, and dependencies. Identify local registration, insurance, and permission questions for me to verify without asserting legal requirements.
9. Financial assumptions: separate one-time startup uses, recurring fixed expenses, variable costs, owner compensation, and working cash. Build a 12-month planning table using my inputs or clearly labeled assumptions. Separate sales from cash received, and expenses from payment timing. Show beginning cash, receipts, payments, financing flows if supplied, and ending cash. State exclusions such as taxes or debt service when not modeled. Do not call a cash balance profit.
10. Scenarios: conservative, base, and optimistic assumptions for volume, price, and costs. Tie sales volume to delivery capacity. Explain the most sensitive assumption; avoid false precision.
11. Risks and validation: five relevant risks, early warning signs, response actions, and the next evidence to collect.
12. A 90-day action schedule: task, owner, due date, cost, dependency, and definition of done. Finish with a one-page decision summary and a weekly review checklist.

Quality Check

Reconcile the numbers and assumptions across sections. Revenue estimates must equal units sold times price for each offer. Cash must reconcile from one month to the next. Separate aspirational targets from forecasts. List any unanswered question that prevents a reliable decision. End with a completeness checklist for all twelve sections.

Source extensions and troubleshooting

Master p1121. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Make it fit my industry: Adapt my plan for [industry] serving [customer type] in [location]. Keep my confirmed facts. Revise the offer, operating workflow, acquisition channels, and risks. Mark new assumptions and explain what I should verify.

Make a shorter version: Create a two-page working version of my completed plan. Preserve the customer, offer, assumptions, cash constraints, risks, and 90-day actions. Keep the full plan intact.

Revise after feedback: Update the plan using this new evidence: [paste feedback or results]. Show which assumptions changed, which sections need revision, and what action I should take next.

The plan is generic. Supply one customer type, one offer, one location or channel, and one spending limit. Ask for tasks you can complete this week.

Numbers appear without evidence. Ask for an assumption register with input, source, confidence, and validation task. Replace unsupported statistics with research questions.

Cash and sales do not match. Check when customers pay and when bills are due. A sale on credit can occur before cash is received. Reconcile beginning cash plus receipts minus payments to ending cash.

Revenue example. Illustration only: 10 units sold at $100 each gives $1,000 in sales. If only $600 is collected this month, cash receipts are $600. Neither figure alone is profit.

P002 Restaurant-industry side service

Business direction and money

Source project: Side Business Project. Master pp 1013, 1123-1124, 2852. Restaurant application below is an editorial adaptation.

Restaurant application

Test a menu-update, food-photo, catering-preparation or restaurant website service using one offer, available hours, a spending limit and evidence of demand.

Source based lesson

Create a focused 30-day test for a side business that fits your available time, skills, and spending limit.

Gather and build

Available hours, usable skills, spending cap, minimum acceptable earnings. Select one service; reserve two work blocks; track every hour and expense in a small pilot.

Check before relying on it

Count travel, messages, preparation, and unpaid waiting, not just delivery time. Compare the pilot with your time target before buying equipment.

Source based practice with fictional inputs

Four jobs bring in $240. Direct costs are $40. Delivery takes six hours and travel/admin takes four.

Worked result: Included contribution is $200 over ten total hours, or $20/hour before excluded costs. Use the same complete time basis for both options being compared.

Try a changed condition: One job is cancelled: receipts fall to $180, costs to $30, and total time to eight hours.

Check your answer: Included contribution is $150; $150/8 = $18.75/hour.

Restaurant boundary

Use actual capability and capacity. A restaurant side business still needs the applicable operating permissions; the prompt does not establish them.

P002 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake Questions

1. Do I already have a side-hustle idea? If yes, what is it?
2. What skills, experience, equipment, or audience can I use?
3. How many hours can I work each week, and on which days?

4. What is my maximum test spend and acceptable loss?
5. Do I prefer online, local, physical, digital, or service work?
6. Which people or businesses could I realistically reach?
7. What is my first goal, target date, and reason for choosing it?
8. What commitments, employer restrictions to check, travel limits, or tasks should the plan avoid?

If I have no idea, propose three suitable options. Compare customer, deliverable, required skills, first test, cash required, weekly hours, and uncertainties. Explain your recommendation, then WAIT for me to choose. If I already chose an idea, proceed with that idea.

BUILD MY 30-DAY PLAN 1. A one-paragraph concept: a specific customer, problem, deliverable, and reason the offer fits my constraints. 2. A first offer: what the customer receives, exclusions, delivery time, proposed test price, payment timing, revision limits if relevant, and how I will know the work is complete. 3. A demand test: one way to reach likely buyers, the question or offer to present, what evidence to record, and a minimum test size appropriate to my time. Clearly label all success thresholds as proposed decision rules, not industry benchmarks. 4. A setup checklist: only the tools, materials, sample, payment method, and basic tracking needed for the first test. Flag local permissions or employer restrictions for verification. Do not assume I need a website, logo, or paid subscription to test the idea. 5. A spending plan: one-time test costs, per-order costs, optional later purchases, and remaining test budget. Keep total planned spend within my cap. State excluded costs and any tools that may require a separate subscription. 6. A capacity plan: allocate weekly time to prospecting, administration, and delivery. Estimate orders I could fulfill using time per order; keep this separate from expected demand. Show a worked example and the assumptions. 7. A four-week schedule: weekly objective, concrete tasks, time per task, estimated spend, output, and review question. Week 1 prepares and tests the offer, week 2 follows up and attempts first delivery, week 3 improves using evidence, and week 4 reviews whether to continue. Adapt this order if delivery needs a different timeline. 8. Two short messages I can adapt: one first-contact message and one follow-up. Use honest language and no invented testimonials, credentials, or scarcity. 9. A results log template: date, action, channel, time spent, cash spent, responses, orders, cash collected, refunds, delivery hours, and lesson. Explain how to avoid counting a follow-up as a new lead. 10. A decision review: proposed thresholds for continue, change, or stop; remaining budget; evidence collected; and a next-week action. A small test can be inconclusive; distinguish that from proof of success or failure.

Keep the plan practical for my available hours. Do not imply that first-month income is assured. If the desired income cannot fit the proposed price and delivery capacity, show the mismatch and offer a smaller test or revised goal. End with a first-day checklist and a list of the ten completed sections.

Source extensions and troubleshooting

Master p1125. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Fit a smaller schedule: Revise my plan for [hours] hours per week on [days]. Keep setup, prospecting, delivery, and administration inside that total. Identify what to postpone.

Lower the starting cost: Rebuild my first test with a maximum spend of [amount and currency]. Separate necessary purchases from optional ones. Use tools I already own where practical.

Use my first results: Review these test results: [actions, responses, orders, money, hours]. Compare them with my original assumptions. Recommend one change for the next week and explain what evidence would support it.

Too many ideas. Choose one customer and one offer for the first 30 days. Put unrelated ideas on a later list.

The goal exceeds capacity. Subtract admin and selling time before estimating delivery capacity. Capacity is a ceiling, not a sales forecast.

Capacity example. With 6 hours available, 2 reserved for sales and admin, and 1 hour per order, delivery capacity is 4 orders per week. Actual sales may be lower.

Costs keep increasing. Compare actual spending with the original test cap. Pause optional purchases and ask for a smaller test using the remaining amount.

P003 Restaurant management overview

Business direction and money

Source project: Business Overview Project. Master pp 1014, 1127-1128, 2864. Restaurant application below is an editorial adaptation.

Restaurant application

Adapt categories to restaurant receipts, supplier spending and management tasks; retain leads only if catering/private-event sales matter.

Source based lesson

Build a monthly overview of cash collected, spending, leads, and tasks, with simple source logs that feed the dashboard.

Gather and build

Receipts, expense log, lead log, task log, reporting dates. Define each metric; calculate it from source rows; add a link from every total to its underlying records.

Check before relying on it

A refund and a duplicate lead must change only the appropriate totals. Reconcile one week manually before adding charts.

Source based practice with fictional inputs

Five distinct inquiries produce two paying customers. A sixth form row duplicates one inquiry.

Worked result: Distinct-inquiry conversion is 2/5 = 40%, not 2/6. Deduplicate confirmed events and define the cohort before calculating the rate.

Try a changed condition: One additional distinct inquiry arrives with no purchase.

Check your answer: The rate becomes 2/6 = 33.33% for the expanded cohort.

Restaurant boundary

Tracked cash surplus is not bank cash, accounting profit or a complete restaurant P&L. Do not sum transaction-level POS records and processor payouts together.

P003 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

Ask for business type; locale and currency; reporting month; revenue and expense categories; lead statuses; task owners and desired measures.

Base Tabs And Columns

Revenue: A Receipt ID, B Date Received, C Customer, D Category, E Amount, F Notes. Expenses: A Transaction ID, B Date Paid, C Vendor, D Category, E Amount, F Notes. Leads: A Lead ID, B Date Added, C Company or Person, D Source, E Status, F Owner, G Notes. Tasks: A Task ID, B Task, C Owner, D Due Date, E Status, F Notes, G Due Flag. Lists: separate columns for revenue categories, expense categories, lead statuses, task statuses, and owners. Dashboard: put Month Start in A2 and a real first-of-month date in B2. Use A4:B11 for metric labels and values. Reserve A15:E27 for a 12-month trend table.

Data Rules

Revenue records cash received from customers, not unpaid invoices. Enter a customer refund as a negative amount on its actual refund date. Expenses records payments as positive and vendor refunds as negative. Exclude transfers between your own accounts, owner contributions, loans, and card bill payments when the underlying purchases are already logged. This dashboard tracks the entered operating transactions, not total bank cash, accounting profit, or tax results. Do not sum multiple currencies.

Use lead statuses New, Contacted, Qualified, Won, Lost and task statuses Not Started, In Progress, Done. One lead ID represents one prospect; update its existing row rather than adding each contact attempt. One task ID represents one task. For Tasks G2, return blank without an ID, Done for completed tasks, No date for missing due dates, Overdue before today, Today for today, and Upcoming after today.

Dashboard

B4: cash collected in the selected month. B5: net expenses paid in that month. B6: tracked cash surplus, B4 minus B5. B7: leads added during that month. B8: leads added that month currently marked Won. Label this as a current cohort count; it is not the number of deals closed during the month. B9: all tasks overdue as of today, excluding Done. B10: all open tasks due today. B11: all open tasks with no due date. Clearly separate monthly metrics from today's task snapshot.

Use inclusive month start and exclusive next-month start. B4 must be: =SUMIFS(Revenue!$E$2:$E$1000,Revenue!$B$2:$B$1000,">="&$B$2,Revenue!$B$2:$B$1000,"<"&EDATE($B$2,1)) Build B5 with the matching Expenses columns. B6 is =B4-B5. Give the remaining formulas with their exact cells.

In A15:E27 create Month, Collections, Expenses, Cash Surplus, and Leads Added for the calendar year containing B2. Generate twelve real month-start dates, even when a month has zero records. Provide a collections-versus-expenses chart from the trend table and a lead-source count table with a bar chart. State which figures are all-time and which use the selected month.

Checks And Handoff

Use two 500 receipts and one 200 expense in the selected month: B4=1000, B5=200, B6=800. Move one receipt to next month and expect B4=500. Add a completed overdue task and confirm it is excluded. Explain that chart and SUMIFS totals do not automatically change when raw rows are hidden by a filter. End with setup, daily-entry, and monthly-review checklists.

Source extensions and troubleshooting

Master p1129. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Add a useful measure: Add [metric] to my dashboard. Define the metric, source fields, date basis, formula cell, and a worked test. Tell me if a new source column is required.

Change the reporting period: Add a weekly view while preserving the monthly dashboard. Use a week-start input and dates from that day up to but excluding seven days later. Provide all cell changes.

Improve its appearance: Style the dashboard with [colors]. Keep input tables unmerged, use readable labels, format money consistently, and distinguish editable cells from formulas.

The monthly total is zero. Check that B2 and transaction dates are real dates, that B2 is the first of the intended month, and that amounts are numbers rather than text.

The total is too high. Look for duplicate transaction IDs, counted transfers, or both a card purchase and its later card payment. Record each underlying event once.

The total is not bank cash. The dashboard omits financing, opening balances, and other unlogged movements. Its cash surplus is collections minus tracked expenses for the period.

A filter did not change totals. These summaries use their stated date and status criteria. Ask for a separately defined filter-aware measure if that is what you need.

P008 Restaurant expense and receipt log

Business direction and money

Source project: Expense Management Project. Master pp 1019, 1147-1148, 2924. Restaurant application below is an editorial adaptation.

Restaurant application

Use vendor, ingredient, packaging, utilities and other user-approved categories without changing the cash sign rules.

Source based lesson (restaurant wording adapted)

Record payments, vendor refunds, categories, receipts, and a user-entered business-use allocation in a simple cash expense log.

Gather and build

Payment date, vendor, amount, category, receipt, allocation. Enter one row per payment or refund; link evidence; reconcile against a statement.

Check before relying on it

Use one sign convention and check that refunds reduce spending. Resolve uncategorized entries before treating the monthly total as complete.

Source based practice with fictional inputs

A $150 payment, a $30 received refund, and a duplicate copy of the $150 payment appear in the log.

Worked result: After removing the confirmed duplicate, net cash outflow is $120. Deduplicate by verified transaction identity and apply a stated refund sign rule.

Try a changed condition: The $30 refund is only approved, not received.

Check your answer: Current cash outflow remains $150; record the pending refund separately.

Restaurant boundary

Business-use allocation is not a tax conclusion; exclude duplicate card payments and account transfers.

P008 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask for the restaurant business scope; locale and currency; expense categories; payment account labels; reporting month; and whether any documented business-use allocation is needed. Do not add household spending.

Base Tabs And Columns

Expenses: A Transaction ID, B Date, C Vendor, D Description, E Category, F Type, G Amount, H Business Use Percent, I Payment Account, J Receipt Link, K Related Transaction ID, L Notes, M Net Expense, N Business Portion, O Duplicate Flag. Lists: categories, types, and account labels. Types are Expense and Refund. Summary: Month Start in A2 with a real first-of-month date in B2; totals, category breakdown, and a 12-month trend.

Use one row per payment or refund. ID is unique and permanent. A refund has its own ID; Related Transaction ID points to the original payment when known.

Entry Rules

Enter G as a nonnegative amount for both expenses and refunds. F supplies the sign. H is a numeric percentage between 0% and 100%, entered by the user; a blank percentage means allocation is unknown. Do not assume deductibility or calculate tax owed. Explain that the Business Portion is an organizational allocation, not a tax conclusion.

Date is when money was paid or the refund was received. Use the original category for a related refund unless the user has a reason to classify it differently. Exclude transfers between owned accounts and card bill payments when the purchases were already recorded. One currency per workbook; do not combine different currencies without a separately documented conversion process.

Formulas

Expenses M2, filled through M1000: =IF(A2="","",IF(OR(F2="",G2=""),"",IF(F2="Expense",G2,IF(F2="Refund",-G2,"")))) Expenses N2, filled through N1000: =IF(A2="","",IF(OR(M2="",H2=""),"",M2*H2)) Expenses O2, filled through O1000: =IF(A2="","",IF(COUNTIF($A$2:$A$1000,A2)>1,"Duplicate ID","")) Flag missing date, category, type, amount, and allocation where relevant. Zero is a valid amount or percentage; blank is missing. Flag refund references that do not match an existing ID, but allow a documented unknown reference.

Summary

Show selected-month gross expense payments, refunds, net expenses, recorded Business Portion, and the count of rows missing allocation. Label allocated totals as partial when any included transaction lacks H. Sum by dates greater than or equal to B2 and less than EDATE(B2,1). Provide net expense by category and account. Keep refunds signed in the category totals. Add twelve calendar-month rows and a net-expense trend chart.

The Net Expense monthly total uses: =SUMIFS(Expenses!$M$2:$M$1000,Expenses!$B$2:$B$1000,">="&$B$2,Expenses!$B$2:$B$1000,"<"&EDATE($B$2,1))

Tests

Enter Expense 100 at 80% and Refund 25 at 80% in the selected month: net expense=75, allocated portion=60. Change allocation to 0% and expect zero, not blank. Leave another allocation empty and confirm the missing-allocation count rises. Add a next-month payment and confirm it is excluded. Duplicate IDs must be flagged. End with entry and monthly reconciliation checklists.

Source extensions and troubleshooting

Master p1149. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Add categories: Replace my categories with [list]. Map existing entries before changing dropdowns and revise the category summary.

Review an import: Map these bank-export headers: [headers]. Explain how to exclude transfers and avoid duplicate purchases. Use fictional sample rows only.

Refunds increase spending. Enter a positive Amount and choose Refund. The Net Expense formula applies the negative sign.

Business Portion is unexpectedly blank. Check Business Use Percent. Enter 0% for no allocated business use; leave blank only when unknown.

Totals are duplicated. Look for repeated imports and both purchases and their card bill payments. Duplicate IDs catch reused IDs, not every possible duplicate purchase.

Receipt links do not work. Confirm that the link points to the intended file and that access is limited to the people who need it.

P009 Guest receipts, refunds and fees

Business direction and money

Source project: Income Management Project. Master pp 1020, 1151-1152, 2936. Restaurant application below is an editorial adaptation.

Restaurant application

Use restaurant sales-channel references, actual payment/refund dates and verified processor fee records.

Source based lesson

Track customer receipts, refunds, transaction fees, sources, and net cash from recorded sales without treating unpaid invoices as cash.

Gather and build

Receipt date, customer, amount, fees, refund, order reference. Record cash received; separate fees and refunds; reconcile payouts with sales records.

Check before relying on it

An unpaid invoice must not increase cash received. Explain the difference between gross receipts and the amount deposited.

Source based practice with fictional inputs

Gross receipts are $250, fees $10, and paid refunds $40.

Worked result: Net payout is $200 before other costs. Keep billed amounts separate from received funds.

Try a changed condition: Another $50 is actually collected with a $2 fee.

Check your answer: Total net payout becomes $248, assuming no other changes.

Restaurant boundary

Not a full ledger; separately collected sales tax is excluded. Delivery settlements need a documented mapping. Never double-count payouts and constituent sales.

P009 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

Ask for income sources; locale and currency; categories; payment methods; reporting month; and whether fees or partial payments occur.

Base Tabs And Columns

Receipts: A Receipt ID, B Date, C Customer or Source, D Description, E Type, F Category, G Gross Amount, H Fees, I Net Amount, J Payment Method, K Invoice Reference, L Related Receipt ID, M Notes, N Duplicate Flag. Lists: types, categories, and payment methods. Types are Sale and Refund. Summary: Month Start in A2 and a real first-of-month date in B2; selected-month totals, category totals, and a 12-month trend.

One row represents one actual customer payment or refund. Receipt IDs are permanent and unique. Repeated Invoice References are allowed for partial payments; do not treat those references as duplicate transaction IDs.

Entry Rules

Gross Amount is nonnegative for both Sale and Refund. Use Type to set the sign. Enter Fees as positive for a fee charged, zero when none, and negative only for a fee returned to you. A refund's fee field contains only the fee charged or returned with that refund; do not repeat the original sale fee. Blank fees mean unknown, not zero.

Gross Amount excludes separately collected sales tax. Do not calculate tax rates or liability. Exclude owner contributions, loan proceeds, transfers, and unpaid invoices. This is a log of recorded customer cash, not a full accounting ledger, profit calculation, or tax return. Maintain one currency. Invoice Reference helps trace a receipt; it does not create an accounts-receivable balance.

Formulas

Receipts I2, filled through I1000:
`=IF(A2="","",IF(OR(E2="",G2="",H2=""),"",IF(E2="Sale",G2-H2,IF(E2="Refund",-G2-H2,""))))`
Receipts N2, filled through N1000: `=IF(A2="","",IF(COUNTIF($A$2:$A$1000,A2)>1,"Duplicate ID",""))` Flag missing date, type, gross amount, category, and fees. A missing-fee row has a blank Net Amount and must be counted as incomplete. Never present a partial total as complete.

Summary

Show selected-month gross Sale receipts, gross Refunds, known Fees, Net Amount, and incomplete-row count. Give exact SUMIFS formulas using Date greater than or equal to B2 and less than EDATE(B2,1). With complete rows, verify Net Amount equals Sales minus Refunds minus Fees. Label the net total as partial if rows are incomplete.

Provide a category breakdown and twelve months of gross sales, refunds, fees, and net receipts. Create a net-receipts trend chart and a category bar chart. Define the date basis as

actual receipt or refund date, including when it differs from an invoice date. Do not count a processor's bank payout again if its constituent customer payments were already entered.

Tests

Sale 100 with fee 3 gives net 97. Refund 20 with extra fee 1 gives net -21. Together they total 76. If the refund instead returns a fee of 0.60, enter H=-0.60 and expect -19.40. A 0 fee is valid; a blank fee is incomplete. Two payments of 40 and 60 with one Invoice Reference are two legitimate receipts totaling 100 before fees. End with entry and monthly reconciliation checklists.

Source extensions and troubleshooting

Master p1153. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Separate income sources: Add a summary by [source or category]. Use the same selected-month date range and preserve separate gross sales, refunds, fees, and net receipts.

Handle partial payments: Show a fictional example of a 300 invoice paid in three 100 installments. Give separate receipt IDs and payment dates with the same invoice reference.

Prepare an import: Map these payment-export headers: [headers]. Explain whether the export is transaction-level or payout-level and how to avoid counting both.

Net receipts look too high. Check missing fees, duplicate sales, and a bank payout entered in addition to its underlying customer payments.

A refund became positive. Choose Refund and enter Gross Amount as a positive number. The formula supplies the negative sign.

Fees were returned. Enter only the returned fee as a negative number in Fees. For a 20 refund with 0.60 fee returned, net is -19.40.

An unpaid invoice is missing. That is expected in a cash-receipts tracker. Use a separate invoice ledger if you need amounts still owed.

P010 Catering and private-event quotes/invoices

Business direction and money

Source project: Billing Project. Master pp 1021, 1155-1156, 2948. Restaurant application below is an editorial adaptation.

Restaurant application

Useful for invoiced catering or events, not a replacement for a dining-room POS.

Source based lesson

Create numbered quotes and invoices, reconcile partial payments, and separate estimated sales from collected cash.

Gather and build

Quote ID, invoice ID, line items, due date, payment references. Define draft, sent, accepted, and paid statuses; link payments to invoices; show remaining balances.

Check before relying on it

An accepted quote is neither a payment nor a duplicate invoice. Test two partial payments and one corrected invoice.

Source based practice with fictional inputs

Invoice I-01 is $720. Payments are $200 and $180.

Worked result: Outstanding balance is $340. Calculate the balance from invoice, credits, and confirmed payments.

Try a changed condition: A valid $40 credit is applied once.

Check your answer: Outstanding balance becomes $300; the credit is not cash received.

Restaurant boundary

Keep quotes distinct from receivables; retain partial payments, credit balances, tax rounding and missing-rate tests. Deposit adaptations must not double-charge.

P010 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

1. What do you sell, in which currency and locale?
2. What quote expiry, invoice terms, numbering pattern, and payment methods do you use?
3. Are discounts and taxes applicable, and which rates have you verified?
4. Do you need deposits or multiple payments per invoice?

5. Which sender and customer fields should appear on the printable forms?

Structure

Tabs: Documents A:L = DocID, Type, Client, IssueDate, DueOrExpiry, Status, Currency, Subtotal, Tax, Total, Paid, Balance. Lines A:H = LineID, DocID, Description, Quantity, UnitPrice, DiscountAmount, TaxRate, Net. Payments A:E = PaymentID, DocID, Date, Amount, Reference. Add a Print tab with a selected DocID in B2.

Build Requirements

Use Type Quote/Invoice; Status Draft/Sent/Accepted/Declined/Void. A quote is never a receivable. Convert a quote by copying its lines to a new permanent invoice ID and recording the source quote in a separate Notes tab; do not change the quote record. Payments can reference only invoices. Positive payment amounts reduce balances; a negative payment reverses a prior payment and needs a reference.

Lines H2: =IF(A2="","",IF(OR(D2="",E2="",F2="",G2=""),"",D2*E2-F2)) Require nonnegative quantities/prices, discount between 0 and quantity*price, and a verified decimal tax rate (zero must be explicit). Round each line's tax to two decimals in an added I column named TaxAmount. Documents H sums line Net by DocID; I sums TaxAmount; J=H+I. Give complete guarded SUMIFS formulas, flag documents with no lines or incomplete lines, and prevent an incomplete total from looking final. K sums Payments amounts for the invoice; L=J-K. Show negative balances as credit due, not zero.

Create a printable quote and invoice with customer, item detail, issue/due date, totals, terms, payment instructions, and source ID. Exclude Void and Quotes from overdue/open-invoice summaries. Do not invent tax rules, bank details, or legal wording. Add duplicate-ID, unknown-DocID, overpayment, and past-due flags. Summary: issued invoice value, collected payments by payment date, outstanding invoices, and credits, each with a stated date basis.

Test: two units at 100, discount 20, tax 10% gives net 180, tax 18, total 198. Payments 50 and 48 leave 100. A quote with the same figures contributes zero to invoice receivables.

Acceptance Check

The 198 invoice shows 98 paid and 100 due; quotes and voids are excluded; an empty line tax rate is flagged. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

D06: State that the workbook credit-note scenario needs an explicit Credits ledger: CreditID, InvoiceID, Date, Amount, Reason, ApprovalRef. If implementing it, Balance=InvoiceTotal-confirmed payments-approved credits. Credits must remain distinct from cash received, and a duplicate credit ID must be rejected/flagged. Alternatively label this as a concept exercise beyond the base tool.

Source extensions and troubleshooting

Master p1157. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Deposits: Add a deposit-request workflow for [percentage], with separate deposit invoices and a final invoice that avoids charging twice.

Branding: Restyle the print sheet with [brand colors and logo placement], preserving every computed value and the DocID selector.

Different rules: Adapt discounts, rounding, and tax display to these verified requirements: [paste]. Identify every formula that must change.

Paid total is wrong. Check Payment DocID, signs, and dates; match each receipt to one invoice.

Tax differs. Check whether rounding occurs per line or per document and follow your verified requirement consistently.

Quote appears overdue. Require Type=Invoice and exclude Draft/Void before calculating receivables.

Printout is incomplete. Check selected ID, line count, print area, and page breaks before sharing.

P041 Restaurant cash timing forecast

Business direction and money

Source project: Cash Flow Project. Master pp 1052, 1279-1280, 3320. Restaurant application below is an editorial adaptation.

Restaurant application

Use dated supplier payments, rent and other actual commitments with expected receipts and clearly named scenarios.

Source based lesson

Forecast the timing of business cash receipts and payments with explicit assumptions and a rolling balance.

Gather and build

Opening cash, dated receipts, dated payments, timing assumptions. Forecast by week; separate expected from confirmed items; update balances when timing changes.

Check before relying on it

A sale with payment next month must not fund this week's bills. Delay the largest expected receipt and observe the lowest cash balance.

Source based practice with fictional inputs

Opening cash $600, confirmed receipts $300, payments $500; a $400 invoice remains unpaid.

Worked result: Ending cash is $400. Separate invoice value from cash receipts and use actual timing.

Try a changed condition: The customer pays $150 this period.

Check your answer: Ending cash becomes $550; the invoice has $250 outstanding.

Restaurant boundary

Cash forecast is not profit or a financial statement. Delayed receipts must move periods; owner funds/borrowing are financing, not sales.

P041 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

What opening cash is verified? Is the forecast weekly or monthly? What receipts and payments are expected and when? Which are committed versus uncertain? What minimum cash threshold do you want to monitor?

Structure

Inputs: FlowID, ExpectedDate, Type, Description, Amount, Confidence, Scenario. Forecast: PeriodStart, OpeningCash, Receipts, Payments, NetFlow, ClosingCash. Add Assumptions and Actuals.

Build Requirements

Use one currency and nonnegative amounts with Type Receipt/Payment. Base the forecast on expected cash dates, not invoice issue dates or accounting revenue. Label expected, committed, and actual items distinctly. Owner contributions and borrowing are financing cash flows, not sales; include only when supplied.

For monthly periods, Receipts and Payments use SUMIFS with date>=PeriodStart and date<EDATE(PeriodStart,1), filtered by the selected scenario. NetFlow=Receipts-Payments; ClosingCash=OpeningCash+NetFlow; next OpeningCash equals previous ClosingCash. Provide exact guarded formulas and distinguish a blank opening balance from zero.

Create Base/Cautious/Upside scenarios using explicit user-controlled assumptions about timing and amounts. Never present scenarios as predictions or assurance of solvency. Flag balances below the selected threshold and show the first shortfall period. Do not mix currencies without a documented conversion input.

Test opening cash 1,000, receipts 800, payments 1,200 yields closing 600. Next period receipts 500 and payments 700 gives closing 400. Delaying the 800 receipt to the second period changes the first closing to -200. Add a weekly reconciliation routine against actual transactions and an assumptions-change log. This is a planning model, not a financial statement or tax calculation.

Acceptance Check

The base sequence closes at 600 then 400; the delayed receipt produces -200 in the first period. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

D11: For a weekly selection, explicitly use dates >=PeriodStart and <PeriodStart+7 and step starts by7 days. Keep monthly EDATE logic only for monthly periods; preserve scenario filtering.

Source extensions and troubleshooting

Master p1281. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the forecast to [weekly/monthly] periods and [horizon].

Change the workflow: Add a cautious scenario using these explicit timing and amount changes: [list].

Add a useful view: Show forecast versus actual cash with timing differences separated from amount differences.

Check the inputs. Use expected cash dates rather than invoice issue dates.

Check the workflow. Keep transfers between tracked accounts from being counted as new business cash.

Check the result. Reconcile the opening balance and avoid mixing forecast with actual duplicates.

P042 Menu-item contribution and price model

Business direction and money

Source project: Pricing Project. Master pp 1053, 1283-1284, 3332. Restaurant application below is an editorial adaptation.

Restaurant application

Define one sale unit and verified ingredient, packaging, per-sale fee and labor assumptions before comparing prices.

Source based lesson

Compare prices, variable costs, fees, contribution, and markup for one product or service at a time.

Gather and build

Selling price, direct cost, fees, labor assumptions, target contribution. Define the cost basis; calculate contribution and margin; test sensitivity to one cost change.

Check before relying on it

Do not call contribution net profit when fixed costs are excluded. Increase labor time by 25% and see whether the price still works.

Source based practice with fictional inputs

Price $120; included variable cost $72.

Worked result: Contribution is $48; margin on price is 40%; markup on cost is 66.67%. Label both denominators and state excluded costs.

Try a changed condition: Included cost rises to $90 with price unchanged.

Check your answer: Contribution is $30 and margin is 25%; markup is 33.33%.

Restaurant boundary

The source does not cost recipes or establish yields. Contribution is not net profit; markup is on direct cost. Preserve denominator checks for target pricing.

P042 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

What is one unit of sale? Which costs vary per unit? Which fees are fixed per sale or percentage of price? What costs are fixed overhead? Are taxes included in the entered price, and what verified treatment should apply?

Structure

Calculator inputs: UnitPrice, DirectCost, FixedFeePerSale, PercentFee, OtherVariableCost. Outputs: VariableCost, Contribution, ContributionMargin, Markup. Add Scenarios and Assumptions.

Build Requirements

Use a clearly defined pre-tax net selling price unless the user supplies a verified alternate treatment. VariableCost=DirectCost+FixedFeePerSale+UnitPrice*PercentFee+OtherVariableCost. Contribution=UnitPrice-VariableCost. ContributionMargin=Contribution/UnitPrice; Markup=(UnitPrice-DirectCost)/DirectCost, clearly labeled as markup on direct cost. Return N/A for zero denominators and flag missing inputs.

Separate contribution from net profit; fixed overhead, income taxes, and financing are excluded unless modeled explicitly. Percentage fees are decimals between 0 and 1; a zero fee is valid. Do not pull unverified current platform fees from memory.

Build a table for three user-chosen prices with units, costs, and assumptions unchanged. Include a target-contribution-margin price formula: (DirectCost+FixedFeePerSale+OtherVariableCost)/(1-PercentFee-TargetMargin), valid only when the denominator is positive. Explain the assumptions and round the proposed price only after calculation.

Test price 100, direct cost 40, fixed fee 2, percent fee 3%, and other variable cost 5: variable cost 50, contribution 50, margin 50%, markup on direct cost 150%. A target margin of 50% yields price 100. Add a cost-review checklist and sensitivity to one cost changing.

Acceptance Check

The example returns contribution 50, margin 50%, markup 150%, and target price 100. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

D01: Keep both original examples, explicitly label MarkupOnDirectCost and MarkupOnIncludedCost and their formulas. Do not compare their percentages as if they were the same metric.

D07: Add an explicit TargetMargin input, definition, and valid domain; retain positive-denominator guard 1-PercentFee-TargetMargin>0. Define whether target is decimal percentage and distinguish blank from zero.

Source extensions and troubleshooting

Master p1285. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the unit-cost model to [service/product] without mixing fixed overhead into per-sale costs silently.

Change the workflow: Add a labor-cost assumption with hours per unit and a stated hourly cost.

Add a useful view: Create a price sensitivity table for these three price points: [list].

Check the inputs. Margin divides by price; markup here divides by direct cost.

Check the workflow. Validate the target-price denominator before division.

Check the result. Use verified fee inputs and explicit tax treatment.

P043 Restaurant break-even scenario

Business direction and money

Source project: Financial Viability Project. Master pp 1054, 1287-1288, 3344. Restaurant application below is an editorial adaptation.

Restaurant application

Use a specific item, fixed menu bundle or explicitly documented stable sales mix; compare required volume with capacity.

Source based lesson

Estimate the sales volume needed to cover fixed costs under a stated single-product cost model.

Gather and build

Fixed costs, price per unit, variable cost per unit, period. Calculate contribution per unit; divide fixed costs by contribution; round required units upward.

Check before relying on it

If contribution is zero or negative, ordinary positive break-even volume does not exist. Recalculate after a price reduction and explain the volume change.

Source based practice with fictional inputs

Fixed costs $640; price $45; variable cost $20.

Worked result: Contribution is $25/unit. 25.6 rounds up to 26 whole units. Use an upward whole-unit target for positive-contribution break-even.

Try a changed condition: Variable cost rises to the $45 price.

Check your answer: Contribution is zero, so this model has no finite sales-volume break-even for positive fixed costs.

Restaurant boundary

This is a single-product model. Applying it to a mixed restaurant menu without a sales-mix model is unsupported; zero/negative contribution gives no finite break-even.

P043 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Intake

What product and period are modeled? What fixed costs belong to that period? What unit price and variable costs are supported? What capacity limit applies? What scenarios would be useful?

Structure

Inputs: Period, FixedCosts, UnitPrice, VariableCostPerUnit, CapacityUnits. Outputs: ContributionPerUnit, BreakEvenUnitsExact, WholeUnitsNeeded, BreakEvenRevenue, CapacityGap. Add three Scenarios.

Build Requirements

ContributionPerUnit=UnitPrice-VariableCostPerUnit. If fixed costs are positive and contribution<=0, display No finite nonnegative sales-volume break-even under these assumptions and do not divide. If fixed costs are zero, zero units already cover fixed costs; with zero contribution every modeled quantity has zero operating result, while with negative contribution any positive quantity loses money. Do not divide by zero. Exact break-even units=FixedCosts/ContributionPerUnit; whole units needed=ROUNDUP(exact,0). Whole-unit revenue=WholeUnitsNeeded*UnitPrice. Label the distinction between exact theoretical and feasible whole-unit quantities.

Require nonnegative fixed costs and a positive unit price. Missing inputs remain incomplete. Fixed costs and sales must use the same period; do not mix annual fixed cost with monthly volume. Keep taxes, financing, and step changes in capacity excluded unless explicitly modeled. This is a simplified planning calculation, not a guarantee of profitability.

Create a volume table from zero to a user-chosen capacity, showing revenue, variable cost, total cost, and operating result. If charting, use the same unit scale and clearly label the crossing. Show a capacity warning if whole units needed exceed feasible volume.

Test fixed costs 1,000, price 50, variable cost 30: contribution 20, break-even 50 units, revenue 2,500. At 40 units, operating result is -200. At variable cost 50, no finite break-even exists. For multiple products, explain that a stable sales mix is an additional assumption; do not apply the single-product result indiscriminately.

Acceptance Check

The example returns 50 units, 2,500 revenue, and -200 at 40 units; zero contribution blocks division. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

D02: State no finite sales-volume break-even when FixedCosts>0 and Contribution<=0. For FixedCosts=0: zero units already cover fixed costs; with zero contribution every modeled quantity has zero operating result; with negative contribution zero sales break even but positive sales lose money. Do not divide by zero. The retained positive-fixed-cost example remains unchanged.

Source extensions and troubleshooting

Master p1289. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the period to [month/quarter] and reconcile every cost to it.

Change the workflow: Add a capacity constraint of [units] and show the shortfall.

Add a useful view: Compare three cost/price scenarios while stating all changed assumptions.

Check the inputs. Contribution must be positive for a finite break-even result.

Check the workflow. Use matching cost and sales periods.

Check the result. Do not assume a mixed-product business behaves like one product without a stated mix.

P083 Restaurant project/department budget or owner cash-allocation worksheet

Business direction and money

Source project: Budgeting Project. Master pp 1094, 1447-1448, 3824. Restaurant application below is an editorial adaptation.

Restaurant application

Map income/expenses to a bounded restaurant event, startup activity, or operating plan; preserve external inflows versus internal transfers and planned/actual separation.

Source based lesson (restaurant wording adapted)

Plan allocations for a bounded restaurant project or operating budget, compare actual spending, and identify unassigned amounts for a defined period.

Gather and build

Period, confirmed funds available, cost categories, commitments, reserve allocations, actual transactions. Choose one period and scope; allocate confirmed available funds; compare actuals and revise remaining allocations. Label uncertainty rather than treating forecast receipts as cash.

Check before relying on it

Transfers between your own accounts must not be counted as new income. Reconcile the plan to actual balances before making spending decisions.

Source based practice with fictional inputs

The recorded opening total across two restaurant accounts is $1,000; $200 moves from account A to account B.

Worked result: Combined total remains $1,000 absent other events. Record the movement between accounts without increasing total receipts from outside.

Try a changed condition: A genuine $100 outside payment arrives.

Check your answer: Combined total becomes $1,100, with the external receipt recorded separately.

Restaurant boundary

Not full accounting, payroll, taxes, profit, financing advice, or bank reconciliation unless separately specified. Personal staff use can remain an optional private annex.

P083 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which bounded restaurant event, department, startup activity, or operating budget is modeled; period/currency; confirmed versus uncertain inflows; expense categories; planned reserve allocations; tracked accounts; and actual-record classification rules.

Structure

Plan: Category, Type, PlannedAmount. Actuals: TransactionID, Date, Category, Type, Amount, AccountRef. Summary: planned/actual income, spending, savings allocations, and unassigned amount.

Build Requirements

Use Type Income/Expense/Transfer. Enter nonnegative amounts and separate direction/type rather than mixing inconsistent signs. A transfer between tracked accounts is not new income or spending. Savings allocations in the plan are assignments of income; actual transfers to savings must not also become an expense in spending reports.

Define planned unassigned=planned income-planned expenses-planned savings allocations. Actual cash surplus=actual income-actual expenses; label it before savings transfers. Use SUMIFS by date>=period start and date<next period start with category/type filters. Provide exact formulas, dropdowns, and unknown-category flags. Do not assume uncertain income will arrive.

Create a budget review with fixed commitments, flexible categories, irregular costs, and user-chosen priorities. Show overspending as information for adjustment rather than moral failure. Avoid prescriptive percentages or claims about what the user must spend.

Test planned income 3,000, expenses 2,400, and savings allocations 300 gives 300 unassigned. Actual income 3,000 and expenses 2,500 gives cash surplus 500 before savings transfers. A 300 transfer to savings does not reduce the expense-based surplus again. Add a month-close reconciliation and a clean template for the next period.

Acceptance Check

Planned unassigned is 300; actual surplus is 500 before transfers; internal transfers do not inflate totals. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1449. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt categories to [restaurant project priorities] without prescribing spending ratios.

Change the workflow: Add irregular income with confirmed and possible receipts kept separate.

Add a useful view: Create a planned-versus-actual view that excludes internal transfers from income and expenses.

Check the inputs. Do not double count savings transfers as expenses.

Check the workflow. Keep planned amounts separate from actual transactions.

Check the result. Use one currency and a clearly defined period.

P084 Restaurant-owner borrowing statement and hypothetical payment-scenario record

Business direction and money

Source project: Debt Management Project. Master pp 1095, 1451-1452, 3836. Restaurant application below is an editorial adaptation.

Restaurant application

Use verified balances/rates/minimums/fees to organize existing obligations and compare explicitly chosen simplified scenarios.

Source based lesson (restaurant wording adapted)

Record existing restaurant-business debt statements and, only when requested, show user-specified hypothetical payments with explicit approximation assumptions.

Gather and build

Verified statement dates, balances, rates, minimum payments, due dates, fees, actual terms, requested hypothetical payment amounts. Enter current statement data; document unknowns and model assumptions; compare a requested scenario with statements without advising repayment priority or claiming a lender payoff quote.

Check before relying on it

Fees, daily accrual, variable rates, and payment timing can change actual results. Compare the model with the next statement and correct the assumptions.

Source based practice with fictional inputs

An explicitly simplified hypothetical model of existing restaurant borrowing starts at $1,000, nominal annual rate 12%, with a $100 payment after interest.

Worked result: Modeled interest is $10; modeled ending balance is $910. Label the simplified accrual and timing assumptions and compare with actual terms.

Try a changed condition: The actual statement includes a $15 fee not modeled.

Check your answer: Add the fee where applicable; under otherwise identical assumptions, modeled balance is $925.

Restaurant boundary

APR/12 is an approximation, not payoff quote or recommendation to borrow/refinance. No transactions, lender promises, or inferred rights; keep actual financial details private.

P084 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which existing restaurant-business borrowing statements are being recorded; verified statement dates/balances/rates/minimums/fees; actual terms; scheduled payments; and any user-requested hypothetical scenario. Do not recommend taking, refinancing, or prioritizing debt; do not execute payments.

Structure

Debt register; payment calendar; simplified monthly projection; user-requested hypothetical payment scenarios; assumptions; reconciliation.

Build Requirements

Use fictional sample data until the user is ready. Register fields: DebtID, BalanceDate, Balance, APR, MinimumPayment, DueDate, RateType, PromoEnd, TermsSource. Do not invent rates or assume a promotional rate lasts indefinitely. Explain that actual lenders may calculate interest and minimums differently.

For a simplified monthly model, interest=opening balance*APR/12; payment is limited to opening balance+interest+known fees; closing balance=MAX(0,opening+interest+fees-payment). State that APR/12 is a planning approximation, not a payoff quote. Keep rates as decimals, missing terms as unknown, and never silently assume zero interest.

If the user requests a scenario, record their proposed payment amounts and check the verified required minimums; do not recommend a repayment priority or new financial product. Do not recommend skipping payments. If the available amount is below total required minimums, flag the shortfall rather than produce an impossible schedule. Verify any current rights, assistance options, or lender-specific rules through official sources before describing them.

Test opening 1,000, APR 12%, no fees, payment 110: approximate interest 10 and closing 900. At zero APR, the same payment leaves 890. Include a statement reconciliation log and explain that real payoff dates change with daily accrual, variable rates, fees, and payment timing.

Acceptance Check

The sample closes at 900 at 12% and 890 at 0%; shortfalls and unknown terms are visible. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Source extensions and troubleshooting

Master p1453. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Compare these user-selected extra payments: [amounts], with all minimums preserved.

Change the workflow: Adapt the model to these verified lender terms: [paste], identifying approximation limits.

Add a useful view: Add a statement-versus-projection reconciliation with interest and fee differences.

Check the inputs. Missing rates are unknown, not zero.

Check the workflow. Do not present approximate projections as lender payoff quotes.

Check the result. Flag any payment budget that cannot cover stated minimums.

P085 Restaurant equipment-replacement or planned-purchase reserve tracker

Business direction and money

Source project: Saving Project. Master pp 1096, 1455-1456, 3848. Restaurant application below is an editorial adaptation.

Restaurant application

Map savings goals to user-selected equipment, fit-out, or other reserve purposes; preserve opening allocation, paired reallocation, contributions, withdrawals, and actual-cash reconciliation.

Source based lesson (restaurant wording adapted)

Record restaurant equipment or project reserve goals while keeping allocated amounts distinct from actual cash accounts.

Gather and build

Reserve purpose, target amount/date, currently allocated cash, confirmed contribution calendar. Allocate contributions by goal; reconcile total earmarks with actual saved cash.

Check before relying on it

The same $200 cannot be assigned in full to two goals. Review one missed contribution and revise the target date or amount.

Source based practice with fictional inputs

Two restaurant reserve purposes are allocated $300 and $250, but actual available reserve cash is $500.

Worked result: Earmarks exceed available cash by $50. Reduce or reallocate earmarks so the total is supported by actual cash.

Try a changed condition: An additional $100 contribution is received.

Check your answer: Saved cash becomes $600; the $550 earmarks leave $50 unassigned.

Restaurant boundary

No investment recommendation or return promise. A reserve allocation is not a separate bank balance; do not count the same cash twice.

P085 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant equipment, renovation, event, or other reserve purposes are planned; target amounts/dates; actual allocated cash; confirmed contribution calendar; whether funds share an account; and whether interest is excluded or recorded only when credited.

Structure

Goals: GoalID, Goal, TargetAmount, TargetDate, OpeningAllocated, Contributions, Withdrawals, CurrentAllocated, Remaining, Progress. Transactions: EntryID, Date, GoalID, Type, Amount, AccountRef, TransferOrReallocationRef.

Build Requirements

In the base ledger, Type is Contribution or Withdrawal. Record a reallocation as two equal entries: a Withdrawal from one goal and a Contribution to a different goal, sharing one TransferOrReallocationRef. Validate exactly one of each, equal amounts, and distinct goals. The pair changes neither total allocated money nor actual cash. Do not count an additional standalone Reallocation row. CurrentAllocated=Opening+Contributions-Withdrawals. Remaining=MAX(0,Target-Current); Progress=Current/Target when target>0, with overfunding shown separately.

A sinking fund is an allocation, not necessarily a separate bank account. Reconcile total allocated funds to actual cash available for those purposes; do not count the same cash toward multiple goals. Exclude hypothetical interest unless explicitly modeled and label it as an assumption.

Required contribution per remaining period=Remaining/number of contribution dates before the target, using a user-selected calendar. If no dates remain and money is still needed, show Target date reached, not a divide-by-zero result. Avoid promising returns or recommending investments.

Test target 1,200, opening allocation 300, contributions 200, withdrawals 50 gives current 450, remaining 750, progress 37.5%. With five remaining contributions, each would need 150 under the no-interest assumption. Provide a monthly reconcile-and-reprioritize checklist.

Acceptance Check

The sample shows 450 allocated, 750 remaining, 37.5% progress, and 150 per remaining contribution. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

E05: Append TransferOrReallocationRef and validate pair amounts and distinct goals. Net allocation/cash must stay unchanged across the pair.

Source extensions and troubleshooting

Master p1457. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt the contribution calendar to [pay schedule].

Change the workflow: Add a reallocation workflow that preserves the total cash allocation.

Add a useful view: Show goals with overdue target dates or allocated balances exceeding available cash.

Check the inputs. Do not assign the same cash to two goals.

Check the workflow. Keep actual interest separate from hypothetical returns.

Check the result. Handle zero remaining contribution dates explicitly.

P086 Restaurant supplier, rent, utility, and subscription bill-occurrence calendar

Business direction and money

Source project: Bill Management Project. Master pp 1097, 1459-1460, 3860. Restaurant application below is an editorial adaptation.

Restaurant application

Use real verified bills, distinct templates and due occurrences, partial payments, confirmations, owners, and exceptions.

Source based lesson (restaurant wording adapted)

Track actual restaurant bill occurrences, due dates, accountable owners, and payment confirmations without confusing scheduled autopay with paid.

Gather and build

Bill occurrence, due date, amount, payment method, confirmation. Create each due occurrence; separate scheduled and paid; record confirmation references.

Check before relying on it

Do not delete the recurring template when one occurrence is paid. Review upcoming bills and identify any missing amounts or confirmations.

Source based practice with fictional inputs

A restaurant bill occurrence is scheduled for payment, but no payment confirmation is recorded.

Worked result: Status is scheduled, not paid. Record each occurrence and its confirmation.

Try a changed condition: The payment attempt fails.

Check your answer: Keep the bill unpaid and record the failed event and responsible follow-up.

Restaurant boundary

Not payment execution, tax advice, or a promise autopay succeeded. Add explicit bill-owner and billing-period fields if needed for the requested summaries.

P086 Source based build specification

Copy the shared starter, the matching document or spreadsheet setup, your restaurant context, and this specification together. Project-specific requirements are retained and adapted for restaurant use; repeated setup is consolidated. Apply the labeled implementation clarifications as part of the brief. This is not finished software.

Adaptation: restaurant-only context or a non-restaurant example has been revised. See the source and editorial index.

Intake

Ask which restaurant bills recur; actual frequency, period, due date, confirmed/unknown amount, and owner; whether a saved autopay arrangement exists; how completion is verified; and the review horizon. No payment or account action is authorized by this planner.

Structure

BillTemplates: BillID, Payee, Frequency, ExpectedAmount, DueRule, Autopay, AccountRef, DefaultOwner. Occurrences: OccurrenceID, BillID, DueDate, AmountDue, Status, PaidDate, AmountPaid, ConfirmationRef, Owner, BillingPeriodKey. Add Calendar and Review.

Build Requirements

Generate a separate occurrence for each bill period. A template is not a payment record. Status Expected/Statement received/Scheduled/Partially paid/Paid/Disputed/Cancelled/Failed; Autopay is a method, not proof of payment. Paid requires actual confirmation.

Compute remaining amount=AmountDue-AmountPaid when amount due is known; allow a negative result to show a credit or overpayment for review. For unknown variable bills, display Amount pending instead of zero. Flag overdue unpaid occurrences and duplicate BillID plus BillingPeriodKey pairs. Define BillingPeriodKey from the actual billed service period or another documented unique cycle identifier; a due-date change must not create a new bill occurrence. Keep the original OccurrenceID and history. Use real dates and safe account labels, never full credentials.

Create a month view showing due date, payee, confirmed amount, payment method, owner, and status. Keep scheduled cash outflow separate from completed payments. Do not log into accounts, submit payments, or alter autopay.

Test a 90 bill with 30 confirmed paid leaves 60 outstanding. A scheduled autopay for the remaining 60 is unconfirmed; the bill remains Partially paid until the remaining balance is actually settled. Include a monthly reconciliation against statements and a process for changing a recurring template without rewriting old occurrences.

Acceptance Check

The partial-payment example leaves 60 and scheduled payments remain unconfirmed. Give expected results for the worked example and one missing-input case. Finish with a short first-use checklist and a repeat-use routine.

Editorial status clarification

Paid means confirmed full settlement of this occurrence. Record confirmed partial payments as Partially paid and keep the remaining balance active, including in overdue views when applicable. Scheduled, failed, unknown and unconfirmed payments are not Paid. Preserve payment evidence and compute the outstanding balance from confirmed payment records.

Editorial implementation clarifications

Apply these explicit additions or scope limits to the source brief before building. They resolve missing fields or boundaries; the original master has not been edited.

E06: Append Owner and a stable BillingPeriod/PeriodKey or explicitly define period identity. Keep scheduled, partially paid, failed, and confirmed paid states distinct.

Source extensions and troubleshooting

Master p1461. These are optional follow-on changes, not included features. Keep the restaurant scope and approval boundaries above.

Adapt the scope: Adapt recurrence rules to [monthly/quarterly/annual] bills with actual due-date rules.

Change the workflow: Add shared responsibility using safe account references and one owner per bill.

Add a useful view: Create an upcoming-cash view that distinguishes confirmed amounts from estimates.

Check the inputs. Autopay is not evidence that payment succeeded.

Check the workflow. Unknown variable amounts must remain pending.

Check the result. Do not overwrite past occurrences when a bill changes.

Calculator discipline and reusable testing

Master pp 1520,1552-1554; restaurant applications and clarifications are editorial additions. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Define the result before building

Source p1520: name the result precisely, list units and timing, work one case by hand, test blank/zero/negative/large values, change one input at a time, protect formulas and keep a version. A missing input and a genuine zero are different. Do not hide every calculation error as zero.

Source test record

Input set | Expected result | Actual result | Pass or fail | Explanation. For every result state the cost basis, period, units, input source and remaining assumptions.

Consolidated source build request

Create clearly separated inputs, assumptions and results. Give exact Google Sheets cells, labels, formulas, edit/protection instructions, formats, example inputs, expected results and test steps. Explain how to add rows without breaking references. Use the relevant complete project's stricter validation and blank-row rules. This instruction consolidates pp1532-1553 rather than duplicating setup for every calculator.

Source customization prompt

COPY AND PASTE: Review the calculator we just created. Ask me one question at a time about its purpose, my situation, available inputs, desired outputs, reporting period, multiple items/accounts, charts, dashboard, conditional formatting, dropdowns, and printable summary. Then rebuild the specification with exact cells, labels, formulas, formatting, input/formula-cell instructions, and testing instructions. Do not remove important calculations without approval.

Source troubleshooting prompt

COPY AND PASTE: Act as my Google Sheets troubleshooting assistant. Ask which calculator, problem cell, expected result, actual result, current formula, and referenced values. Check cell references, parentheses, percentage/currency formats, text-vs-number issues, blanks, division by zero, absolute/relative references, ranges, dates, circular references, copied formulas, and assumptions. Give the corrected formula, exact paste location, and explanation.

Source design and test instructions

COPY AND PASTE: Professionally format my calculator without changing calculations. Give step-by-step instructions for title, headings, input cells, formula cells, results, borders, number/currency/percentage/date formats, column widths, row heights, frozen rows/columns, conditional formatting, dropdowns, and useful charts. Make editable inputs visually distinct and keep the sheet easy on desktop and mobile. CALCULATOR TESTING CHECKLIST Test normal values, zero values, blank optional inputs, percentages, large values, negative values where relevant, copied formulas, and dates. Manually verify at least one simple example. For important decisions, compare results with an appropriate trusted statement, invoice, accounting record, lender information, or professional source.

Scope

These are educational spreadsheet specifications, not executed files, accounting/tax/lending advice or guaranteed outcomes. Actual statements, agreements and qualified advice take precedence over simplified models.

Contribution and markup practice

Master pp 1521, 1533, 1534, 1535, 1536; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Restaurant item or supplier product unit economics.

What this source adds

Use Project042 as the primary build; this source adds a compact hand-worked check, not another independent price model.

Original worked practice

Price 100; included variable costs 60; quantity 10. Revenue = 100 x 10 = 1,000. Included variable costs = 600. Contribution = 400. Contribution margin = 400 / 1,000 = 40%. Markup on included costs = 400 / 600 = 66.67%. Change one input and test again Change unit costs to 75. Contribution becomes 250 and margin 25%. Price zero should produce a clear undefined-margin result. State the limits Fixed overhead and taxes are excluded here. Do not label the result net profit.

Implementation clarification (editorial)

The older builder labels results gross profit/profit while mixing configurable costs. Name the included cost basis and use contribution or modeled surplus as appropriate; do not call it net profit.

Build cross-reference

Preserve the source worked test above, then use complete Project 042 Pricing for the full build, validation and edge-case requirements.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Restaurant-industry service pricing

Master pp 1522, 1536, 1537, 1538, 1539; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Menu-design, food-photo, restaurant consulting, installation or catering-service quote.

What this source adds

Adds labor-hour, travel, software, equipment, subcontract and overhead allocation for service packages beyond a simple per-item menu price.

Original worked practice

Four delivery hours at an internal cost of 30 per hour; materials 20; allocated overhead 20; target margin 20%. Included cost = 4 x 30 + 20 + 20 = 160. With no additional fees, price = 160 / (1 - 0.20) = 200. Included profit = 40; 40 / 200 = 20%. Change one input and test again Change delivery time to five hours. Cost becomes 190 and target price 237.50. State the limits A wage or internal hourly cost is not the same as the customer billing rate. Add any omitted costs explicitly.

Implementation clarification (editorial)

An internal labor cost is not the billing rate. Do not double-count overhead or fees; for target margin m the denominator must stay positive. Reuse Project042's stronger fee and input-validation safeguards.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Service Pricing Calculator in Google Sheets. INPUTS: Service name Estimated labor hours Hourly labor cost Material costs Travel costs Subcontractor costs Software/tool costs Equipment costs Other direct expenses Overhead allocation Desired profit margin percentage CALCULATE: Total labor cost Total direct costs Total overhead Total estimated cost Recommended selling price Estimated profit Estimated profit margin Effective hourly revenue Create these sections: INPUTS COST BREAKDOWN PRICING RESULTS Give me: Exact cell locations Exact row and column labels Exact Google Sheets formulas Editable input cells Protected/formula cells Currency formatting Percentage formatting Conditional formatting recommendations Five sample services Testing instructions Use IFERROR or appropriate safeguards. When calculating a selling price based on a target profit margin, use the correct margin calculation rather than simply adding the percentage to cost. Explain the difference. After creating the calculator, ask me what industry I want to customize it for.

Original customization and formula guide

Customize my Service Pricing Calculator. Ask me one question at a time about: My industry My services My average labor time Employee labor Materials Mileage Travel Equipment Software Subcontractors Minimum service charges Rush fees Recurring-service discounts Materials markup Project complexity Overhead Target profit margin After I answer, rebuild the calculator specification using my business model. Give me exact Google Sheets formulas and exact cells. LABOR COST: Labor Hours x Hourly Labor Cost TOTAL COST: Labor + Materials + Travel + Subcontractors + Equipment + Software + Other Costs + Overhead TARGET-MARGIN PRICE: If the desired margin is expressed as a percentage of the final selling price: Selling Price = Total Cost / (1 − Desired Margin) ESTIMATED PROFIT: Selling Price − Total Cost Always verify that percentages are entered consistently.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Whole-unit break-even practice

Master pp 1523, 1539, 1540, 1541; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

One restaurant product, fixed catering bundle or explicitly modeled stable mix.

What this source adds

Use Project043 as the primary build; retain the hand-worked whole-unit and nonpositive-contribution tests.

Original worked practice

Fixed costs 600 per month; price 50; variable cost 20 per unit. Contribution per unit = 50 - 20 = 30. Break-even = 600 / 30 = 20 units per month. At 19 units the modeled result is -30; at 21 it is +30. Change one input and test again Use fixed costs of 610: 20.33 units rounds up to 21 whole units. State the limits This is a single-product model with constant prices and costs. A zero or negative contribution does not yield a normal positive break-even target.

Implementation clarification (editorial)

Do not apply this one-product formula to an unmodeled mixed menu. Fixed costs and units must use the same period.

Build cross-reference

Preserve the source worked test above, then use complete Project 043 Financial Viability for the full build, validation and edge-case requirements.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Billable-capacity rate planner

Master pp 1524, 1541, 1542, 1543; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Restaurant-industry consultant, designer, photographer, trainer or software freelancer.

What this source adds

Unique annual working-time, billable-utilization and revenue-requirement logic.

Original worked practice

Annual compensation target 48,000; operating-cost allowance 12,000; 1,000 billable hours. Revenue requirement in this simplified model = 60,000. Rate floor = 60,000 / 1,000 = 60 per billable hour. Change one input and test again Reduce billable hours to 800: the modeled floor becomes 75. State the limits The example is not a tax or take-home-pay estimate. Add your actual obligations and nonbillable work assumptions.

Implementation clarification (editorial)

Tax reserves are user-supplied planning assumptions, not a tax calculation. Define whether percentages apply to revenue or another base. Do not count holidays again if already excluded as whole weeks; market willingness to pay is not established.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Freelance Rate Calculator. INPUTS: Desired annual personal income Annual business expenses Estimated amount or percentage set aside for taxes Desired annual profit or reserve Vacation weeks Holidays or non-working days Working hours per week Percentage of working time that is billable Average project hours CALCULATE: Available working weeks Estimated total working hours Estimated billable hours Required annual revenue Required monthly revenue Required weekly revenue Suggested hourly rate Suggested day rate Suggested project rate Clearly separate: INPUTS ASSUMPTIONS RESULTS Give me exact cells and exact Google Sheets formulas. Explain billable versus non-billable time. Include sample numbers. Use safeguards against zero billable hours. Explain that this is an estimate and does not determine what the market will pay.

Original customization and formula guide

Customize my Freelance Rate Calculator for my work. Ask me: What service I provide My income goal My expenses My expected taxes or tax reserve assumption My working schedule My vacation time My billable percentage My average project length Whether I charge hourly or by project Whether I want minimum project pricing Whether I want rush pricing After I answer, update the calculator. AVAILABLE WORKING WEEKS: 52 − Vacation Weeks − Other Full Weeks Off ESTIMATED BILLABLE HOURS: Available Working Hours x Billable Percentage BASIC REQUIRED HOURLY RATE: Required Annual Revenue / Billable Hours PROJECT RATE: Hourly Rate x Estimated Project Hours You may also need to account for project-specific expenses, revisions, subcontractors, or contingency.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Catering and service project cost

Master pp 1525, 1543, 1544, 1545; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Catering production/service project, restaurant launch support, photo shoot or fitout contractor estimate.

What this source adds

Unique multiple labor categories, line-item costs, explicit contingency base and estimate-versus-price comparison.

Original worked practice

Ten hours at 40 internal cost; materials 100; subcontract work 200; contingency 10% of those costs. Base cost = 400 + 100 + 200 = 700. Contingency = 70. Estimated cost = 770. Change one input and test again Add two hours: base becomes 780, contingency 78, total 858. State the limits A cost estimate is not automatically a customer price. State exactly which cost base receives contingency.

Implementation clarification (editorial)

A cost estimate is not a quote or customer price. Apply contingency only to the declared base; do not double-count allocated overhead, travel or materials. Technical/permit inputs come from qualified sources.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Project Cost Calculator. INPUTS: Project name Client Labor categories Labor hours Hourly labor costs Materials Equipment Software Travel Subcontractors Permits and fees Other expenses Contingency percentage Client price CALCULATE: Labor total Material total Equipment total Other direct costs Subtotal Contingency amount Estimated total project cost Client price Estimated project profit Estimated profit margin Create a detailed cost table and summary section. Allow multiple expense rows. Give me exact Google Sheets cells and formulas. Include sample project data. Show me how to add additional cost rows without breaking formulas.

Original customization and formula guide

Customize my Project Cost Calculator. Ask me about: My industry My projects Labor Materials Equipment Travel Subcontractors Permits Software Contingency Client pricing Desired profit After I answer, update the calculator for my projects. LABOR TOTAL: Sum of Labor Hours x Applicable Hourly Costs SUBTOTAL: Labor + Materials + Equipment + Travel + Subcontractors + Fees + Other Expenses CONTINGENCY: Subtotal x Contingency Percentage ESTIMATED PROJECT COST: Subtotal + Contingency ESTIMATED PROFIT: Client Price − Estimated Project Cost PROFIT MARGIN: Estimated Profit / Client Price

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Agreement-based sales commissions

Master pp 1526, 1545, 1546, 1547; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Restaurant-supplier sales, catering sales or agency commission agreements.

What this source adds

Unique eligibility, commission rates, bonus, adjustment, refund and payment-status calculations.

Original worked practice

Eligible collected sales 2,000; commission rate 5%; eligible refund 200. If the agreement nets refunds before commission: basis = 2,000 - 200 = 1,800; commission = 1,800 x 0.05 = 90. Change one input and test again Without the refund, commission is 100. Test an ineligible sale that should not enter the basis. State the limits Eligibility, timing, chargebacks, and tier rules must come from the actual agreement, not a generic formula.

Implementation clarification (editorial)

Not a tip pool or payroll-compliance system. Actual agreements determine eligibility, refund/clawback timing and tiers; payment status must not be inferred from accrual. Revenue remaining after commission is not profit.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Sales Commission Calculator. INPUTS: Salesperson Client/customer Sale date Eligible sale amount Commission percentage Bonus Adjustment Payment status CALCULATE: Base commission Bonus amount Adjustments Total commission Company revenue remaining after commission Monthly commissions Year-to-date commissions Create: SALES TABLE MONTHLY SUMMARY YEAR-TO-DATE SUMMARY Give me exact formulas and cell locations. Design it so additional sales rows can be added easily. Include sample sales. Explain how to handle percentages correctly.

Original customization and formula guide

Customize my Sales Commission Calculator. Ask whether I use: Flat commissions Tiered commissions Quotas Bonuses Recurring commissions Different rates by product Team commissions Manager overrides Chargebacks Refund adjustments Paid/unpaid status Then rebuild the calculator based on my commission structure. BASE COMMISSION: Eligible Sale Amount x Commission Rate TOTAL COMMISSION: Base Commission + Bonus + Adjustments COMPANY REVENUE AFTER COMMISSION: Eligible Sale Amount − Total Commission Actual commission agreements can include eligibility rules, payment timing, cancellations, refunds, clawbacks, or other terms. Configure the calculator to match the applicable agreement.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Business budget and variance

Master pp 1527, 1547, 1548, 1549; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Changed example notice

The source household budget intake/categories are replaced by a restaurant-business intake. Planned/actual variance and allocation arithmetic are retained; the original source is preserved in the audit.

Restaurant-industry application (adapted)

Restaurant startup/department budget, catering event budget or agency monthly spending plan.

What this source adds

Adds planned-versus-actual allocation variance, unlike actual transaction logs or a dated cash forecast.

Original worked practice

Available income 3,000; fixed allocations 1,800; variable allocations 700; savings allocation 300. Unassigned = 3,000 - 1,800 - 700 - 300 = 200. Change one input and test again Raise variable allocation to 950: unassigned becomes -50 and should be visible as overallocated. State the limits A savings transfer is an allocation here, not a new expense in every accounting context. Use consistent labels.

Implementation clarification (editorial)

Original categories are household-oriented. Label restaurant business categories as an adaptation. Transfers to reserves are allocations, not a second operating expense; define positive and negative variance consistently.

Restaurant-adapted calculator build prompt

Create a Google Sheets Monthly Business Budget Calculator for my restaurant or restaurant-industry business. Ask for currency, month, income/receipt sources, planning basis and approved categories. Use PLANNED and ACTUAL columns. Suggested categories to confirm: ingredients/products, packaging, labor/service delivery, rent, utilities, insurance, transport, marketing, equipment, software and other operating costs. Record debt payments, reserve allocations and financing separately with clear labels; do not count transfers twice. Calculate total available funds, operating allocations, separately labeled non-operating allocations, unassigned funds and planned-minus-actual variance by category. Add input/assumption/result sections, exact cells/formulas, formats, sample records and tests. Define positive variance as under budget and negative as over budget for expense categories; state any different income-variance convention. Make overspending visible, distinguish blank from zero and retain source reconciliation.

Restaurant-adapted customization and source formula logic

Ask about the restaurant business scope, income/receipts basis, operating categories, seasonal or annual costs, reserve goals, debt payments, reporting month and owners. Then customize the category list and exact cells/formulas. Source logic: money remaining equals included income minus included allocations; savings/reserve rate equals the selected reserve allocation divided by the stated income base; expense rate equals included expense divided by that base; budget difference equals planned minus actual. Mark zero-denominator rates not applicable. Reserve transfers are allocations in this model, not automatically new accounting expenses or profit deductions.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Equipment reserve contribution plan

Master pp 1528, 1549, 1550; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Restaurant equipment replacement or seasonal working-capital reserve planning.

What this source adds

Unique remaining-goal and contribution-period arithmetic.

Original worked practice

Target 1,200; current earmarked savings 300; six remaining monthly contributions; no interest. Remaining amount = 900. Required contribution = 900 / 6 = 150 per month. Change one input and test again With three contributions remaining, the amount becomes 300 per contribution. State the limits If the deadline has passed or no contribution periods remain, show a planning error rather than dividing by zero.

Implementation clarification (editorial)

This is a funding-allocation exercise, not an investment recommendation. Set growth blank/zero by default; any return is hypothetical. A passed target date or zero periods is a visible planning error, not zero contribution.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Savings Goal Calculator. INPUTS: Goal name Target amount Current savings Target date Expected contribution amount Contribution frequency Optional expected annual return CALCULATE: Amount remaining Time remaining Required monthly contribution Required weekly contribution Progress percentage Projected completion date Estimated future value if an optional annual return is included Create GOAL INPUTS, SAVINGS REQUIREMENTS, PROGRESS SUMMARY, and OPTIONAL GROWTH SCENARIO sections. Give exact cells, labels, Google Sheets formulas, editable cells, formula cells, currency/percentage/date formatting, conditional formatting, five sample goals, and testing instructions. Make the calculator work when expected return is blank or zero. Clearly label return-based results as estimates. CUSTOMIZATION INSTRUCTIONS

Original customization and formula guide

Customize my Savings Goal Calculator. Ask me one question at a time about my goal, target amount, current savings, target date, contribution frequency, affordable contribution, optional annual return, multiple goals, progress charts, and milestones. Then update the calculator with exact cells and formulas. TROUBLESHOOTING & FORMULA GUIDE Amount Remaining = Target Amount - Current Savings Progress Percentage = Current Savings / Target Amount Simple Required Contribution = Amount Remaining / Remaining Contribution Periods Warn me if the target date has passed. Avoid division by zero. Investment returns are assumptions, not guarantees.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Loan-payment arithmetic practice

Master pp 1529, 1550, 1551; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Understand a restaurant equipment-finance payment schedule using its actual terms, as arithmetic practice.

What this source adds

Unique amortization schedule and extra-payment comparison beyond cash forecasting.

Original worked practice

Opening balance 1,000; nominal APR 12%; monthly rate 1%; payment 100 at month-end; no fees. Month 1 interest = 10. Ending balance = 1,000 + 10 - 100 = 910. Month 2 interest = 9.10; ending balance = 819.10. Change one input and test again With a payment of 10, the first month has no principal reduction. Flag a payment below interest as balance growth under this model. State the limits Actual lenders may use daily accrual, fees, varying rates, and different timing. Reconcile against statements; this is arithmetic practice.

Implementation clarification (editorial)

This simplified model is not lender advice or a recommendation to borrow/repay. Daily accrual, fees, rates and timing can differ. Define base payment and extra principal separately so extra payment is deducted exactly once; cap the final payment.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional Debt Payoff Calculator. Allow inputs for debt name, current balance, annual interest rate, minimum payment, additional payment, payment frequency, and optional start date. Calculate total scheduled payment, periodic interest, principal, number of payments, estimated payoff date, total interest, total amount paid, and estimated interest saved from extra payments. Create DEBT INPUTS, PAYOFF SUMMARY, PAYOFF SCHEDULE, and EXTRA PAYMENT COMPARISON sections. The schedule must show payment number, date, beginning balance, payment, interest, principal, extra payment, and ending balance. Give exact cell locations, labels, Google Sheets formulas, editable/formula cells, currency/percentage/date formatting, conditional formatting, sample data, and testing instructions. Prevent the final payment from incorrectly creating a negative balance. State the interest and payment-frequency assumptions. Ask whether I want multiple debts, snowball, or avalanche comparisons.

Original customization and formula guide

Customize my Debt Payoff Calculator. Ask me one question at a time about my debts, balances, rates, minimum payments, payment frequencies, extra-payment budget, one-time payments, snowball/avalanche comparisons, payoff schedule, and charts. Then update it with exact cells and formulas. 4. TROUBLESHOOTING & FORMULA GUIDE Periodic Interest Rate = Annual Rate / Payment Periods Per Year Period Interest = Beginning Balance x Periodic Rate Principal = Payment - Interest Ending Balance = Beginning Balance - Principal - Extra Principal If payment is too small to reduce the balance, show a warning. Lender calculations may differ because of daily interest, fees, promotional rates, compounding, or other rules. Compare estimates with lender statements.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.

Restaurant project ROI comparison

Master pp 1530, 1551, 1552; original calculator lesson plus labeled restaurant adaptation. Restaurant applications and changed examples are editorial adaptations unless identified otherwise.

Restaurant-industry application (adapted)

Evaluate a restaurant marketing test, equipment/software project or training investment using checked records.

What this source adds

Unique attributable benefit, full implementation cost, comparison period and optional baseline analysis.

Original worked practice

Project cost 1,000; attributable benefit 1,300 for the selected period; no other included costs. Net benefit = 1,300 - 1,000 = 300. Simple ROI = 300 / 1,000 = 30%. Change one input and test again If attributable benefit is 800, ROI is -20%. Zero cost makes this formula undefined. State the limits Revenue is not necessarily net benefit. Attribution, ongoing costs, and timing must be defined; simple ROI ignores the time value of money.

Implementation clarification (editorial)

Source formula uses Revenue + Other Measurable Benefits. Clarify whether revenue is incremental gross receipts or net benefit: include associated costs, avoid counting benefits/costs twice, and do not claim causal attribution without evidence. Different periods are not comparable without adjustment; simple ROI omits time value and risk.

Original calculator build prompt

You are my Google Sheets calculator builder. Create a professional ROI Calculator. Inputs: investment/project name, initial investment, additional costs, revenue generated, other measurable financial benefits, time period, and optional baseline. Calculate total investment, total financial return, net return, ROI percentage, return multiple, and optional annualized ROI only when appropriate. Create INVESTMENT INPUTS, ROI RESULTS, COMPARISON TABLE, and OPTIONAL BEFORE-AND- AFTER ANALYSIS sections. Allow multiple investments to be compared. Give exact cells, labels, Google Sheets formulas, editable/formula cells, currency/percentage formatting, conditional formatting, five examples, and testing instructions. Explain revenue vs profit vs benefit vs net return vs ROI vs return multiple. Use safeguards when investment is zero. Do not treat revenue as profit.

Original customization and formula guide

Customize my ROI Calculator. Ask me one question at a time about the investment, initial and ongoing costs, implementation costs, revenue, cost savings, other benefits, measurement period, baseline, comparison options, reporting period, and charts. Then update it with exact cells and formulas. 4. TROUBLESHOOTING & FORMULA GUIDE Total Investment = Initial Investment + Additional Costs Total Financial Return = Revenue + Other Measurable Benefits Net Return = Total Financial Return - Total Investment Simple ROI = Net Return / Total Investment Return Multiple = Total Financial Return / Total Investment Simple ROI does not automatically account for time value of money, risk, taxes, financing costs, opportunity cost, or different measurement periods.

Acceptance check

Record normal, changed-input, blank and zero-denominator tests with hand-calculated expected values. Keep nonpositive contributions, insufficient payments, impossible dates or invalid margin assumptions visible instead of returning plausible-looking zeros.