KoreFi Assignment

Area picked: Operations, and why

I picked **Operations** (recovering money on Swiggy and Zomato) because in all three restaurants the final payout could not be tied back to individual orders, and it is examined in detail only when something already looks wrong.

Together the three restaurants take about 240 orders a day, worth ₹78,900 a day or ₹23.67 lakh a month in gross order value. Of these, 145 daily orders (60.4%) come through Swiggy and Zomato. That is ₹14.67 lakh a month of gross order value that reaches the bank only after commissions, discounts, taxes, refunds and adjustments. In every restaurant the step from orders to the final credit is reviewed at summary level by an owner or manager and examined in detail only by exception, with no fixed routine recorded.

I chose Operations over Growth and Finance for three reasons:

- **It is a money-back problem, not a money-forward one.** Recovered deductions go straight to profit and need no change in customer behaviour.
- **It showed up in all three restaurants** despite different cuisines, order volumes and who ran the area.
- **It can be measured.** Every platform settlement can be recomputed order by order, so a coworker's value can be proven in rupees.

Step 1: Restaurant profiles

Three single-outlet restaurants with 60 to 110 orders a day: Restaurant A (Mughlai / North Indian), Restaurant B (snacks, sweets, Chinese, veg main course) and Restaurant C (snacks, Chinese, Italian, non-veg). Restaurant C earns the most per order and depends most on the apps (75% of its orders).

	Restaurant A	Restaurant B	Restaurant C	Total
Type and cuisine	Mughlai / North Indian	Snacks, sweets, Chinese, veg main course	Snacks, Chinese, Italian, non-veg	
Outlets	1	1	1	3
Orders per day	70	110	60	240
Average order value, AOV (₹)	270	300	450	328.75
Revenue per day (₹)	18,900	33,000	27,000	78,900
Revenue per month (₹, 30 days)	5,67,000	9,90,000	8,10,000	23,67,000
Who handles Operations	Owner	Manager	Manager	

The total AOV is revenue divided by orders (78,900 / 240), not the average of the three AOVs.

Channel mix (orders per day)

	Restaurant A	Restaurant B	Restaurant C	Total
Swiggy	25	30	25	80
Zomato	20	25	20	65
Other channels (in-person, direct, takeaway)	25	55	15	95
Swiggy + Zomato share of orders	64.3%	50.0%	75.0%	60.4%

In all three, an owner or manager runs the delivery-app side day to day: the owner at Restaurant A, a manager at B and C. My notes record no outside agency or CA in this area.

Step 2: How they handle Operations today

All three run the same manual, exception-driven process: orders are fulfilled in the platform apps, settlements are reviewed at summary level, and discrepancies are chased only when a credit looks off.

Current process (common to all three)

1. **Orders:** staff watch incoming orders in the Swiggy and Zomato partner apps and cook from there. Order status, cancellations and customer complaints are handled inside each app.
2. **Settlement:** the owner (Restaurant A) or manager (B, C) periodically looks at the platform settlement summaries and bank credits. No fixed daily or weekly routine was recorded.
3. **Discrepancies:** a deduction, refund or short payment is chased only when the amount received looks inconsistent with expected sales, and then order by order through platform support.
4. **Menu and prices:** changed separately in each app. Direct, takeaway and in-person sales are tracked apart from the apps.

Tools: the two partner apps and their settlement reports, bank statements, and manual comparison. My notes record no tool that links an order to the rupees finally credited for it.

Cost in time and money: not quantified in my notes for any of the three, so the loss is unmeasured. I size it in Step 3.

Key problems, ranked by how much they matter

Ranked by (1) how many of the three restaurants raised it and (2) how directly it costs money. Problems are paraphrased from my visit notes (Appendix A).

Rank	Problem	Restaurant A	Restaurant B	Restaurant C
1	**No order-to-payout visibility.** Cannot connect daily orders to the final credit after commissions, discounts, taxes and adjustments; true net revenue per channel is unknown	Yes	Yes	Yes
2	**Discrepancies found by exception and chased by hand.** Small order-level differences go unnoticed; disputes take time to locate, document and follow up, and repeated deductions are treated as one-offs	Yes	Yes	Yes
3	**Cancellation and refund losses.** Orders cancelled after food is made, or for items that were unavailable, cost ingredients and labour with no tracking of the total	No	Yes	Yes
4	**Menu, price and availability drift across platforms.** Updates are repeated by hand in each app and go out of sync	Yes	Yes	No
5	**Dependence on one person remembering to check.** Reviews happen when the owner or manager has time, and slip at peak hours	Yes	Yes	No

Ranks 1 and 2 are one chain: with no order-level view (1), small deductions cannot be seen, and with no evidence trail (2), the ones that are seen take time to dispute. That chain is where the coworker in Step 3 sits.

Step 3: Hypothesis

I would build a **Payout Auditor**: an AI coworker that works out what every Swiggy and Zomato order should have paid, flags every shortfall, and prepares the dispute with evidence. It **reduces cost** by recovering money already earned.

What it does

1. **Collects** the daily order data, the platform settlement report and the bank credit the owner or manager already has.
2. **Recomputes** the expected net for every order from the agreed commission, restaurant-funded discounts, taxes and charges.
3. **Flags** each gap: unexplained deductions, refunds or cancellations charged to the restaurant, and orders that are missing from a payout.
4. **Prepares the dispute:** order ID, expected versus paid, and a ready-to-send message for platform support. It tracks each one until it is resolved.
5. **Reports weekly on one page:** gross to net by channel, money recovered, disputes still open.

It replaces the habit of chasing a payout only when it looks wrong with a routine that does not depend on the owner remembering. Menu and availability sync (problem 4) is the natural second coworker once the payout data is flowing.

Why this problem over the others

Candidate	Raised by	Money at stake	Can the value be proven?	Verdict
Payout reconciliation and disputes	3 of 3	Recurring, on ₹14.67 lakh a month of aggregator order value across the three	Yes: each recovered rupee is traceable to an order	**Build first**
Cancellation and refund tracking	2 of 3	Real but smaller; partly caused by kitchen availability	Partly: needs the cause of each cancellation	Fold into the auditor as a report
Menu, price and availability sync	2 of 3	Indirect: lost orders and refunds, hard to size	Weakly	Second coworker
Single-person dependency	2 of 3	A symptom of the manual process	No	Solved as a side effect

Evidence from the visits

The sizing below applies an assumed leakage rate to the aggregator order value I measured. The order value is observed; **the leakage rate is not, and is the first thing to test.**

Restaurant	Share of orders on apps	Aggregator order value per month (₹)	Leakage at 1% (₹)	Leakage at 2% (₹)	Leakage at 3% (₹)
Restaurant A	64.3%	3,64,500	3,645	7,290	10,935
Restaurant B	50.0%	4,95,000	4,950	9,900	14,850
Restaurant C	75.0%	6,07,500	6,075	12,150	18,225
Total	60.4%	14,67,000	14,670	29,340	44,010

Aggregator order value = Swiggy and Zomato orders per day × AOV × 30. Across the three restaurants it is 62.0% of all order value, slightly above the 60.4% order share because Restaurant C has both the highest AOV and the highest app share.

- At a 2% leakage the three restaurants lose about ₹29,340 a month, or ₹3,52,080 a year, which is 1.2% of their total monthly revenue.
- Across the 1% to 3% range that is ₹1.76 lakh to ₹5.28 lakh a year.
- Restaurant C carries 41.4% of the exposure (₹6,07,500 of ₹14,67,000) on 25% of the orders, because 75% of its ₹27,000 daily revenue runs through the apps at a ₹450 AOV.
- In all three, the visit showed the same gap: no order-level link between what was sold and what was credited, so none of this leakage can currently be seen.

Assumptions and how I would test them

Assumption	How I would test it
Real leakage is between 1% and 3% of aggregator order value	Take one month of redacted settlement reports and order exports from the same three restaurants, recompute expected net, and measure the unexplained variance. If it is under 1%, the problem is too small.
Flagged items are winnable disputes	File the first 20 flagged items and track win rate and days to resolution
Order-level data can be exported in a usable form	Ask each manager which reports they already download and whether they carry order IDs
Owners and managers will act on a weekly list of recoveries	Check whether the first batch of flagged items is actually filed within a week
Swiggy and Zomato payout leakage is a fair stand-in for DoorDash and Uber Eats in the US, as the brief assumes	Repeat the exercise with a handful of US operators' statements

How I did it

- **Preparation:** I picked Operations, and I knew which questions would show where it breaks, because following money from an order to the bank is close to my own work. Before each visit I wrote down a few plain questions (what happens when a payout looks short, who checks it, how often, what do you do about it) and kept the list short so a conversation would fit in 20 to 30 minutes.
- **Notes:** I logged each visit in a working spreadsheet (profile, channel mix, current process, key problems). Revenue and channel shares are spreadsheet formulas, not typed numbers.
- **AI use:** I used Claude to structure the findings, rank the problems, check the arithmetic and draft this document. The visit inputs are mine; the leakage rates in Step 3 are labelled assumptions, not findings.
- **Data handling:** I kept no bank, GST, PAN, login, staff or customer details.

Appendix

A. Notes from each visit

Notes are my paraphrased record of each visit, not transcripts.

Restaurant A: Mughlai / North Indian, 1 outlet. Operations handled by the owner.

- Scale: about 70 orders a day at an AOV of ₹270. Swiggy 25, Zomato 20, other channels 25.
- Process: the owner or manager watches orders in both partner apps. Kitchen staff prepare them. Order status, cancellations and complaints are handled in each app.
- Settlement: the owner or manager periodically reviews settlement summaries and bank credits, using platform reports and bank statements. Discrepancies are checked by hand when the amount received looks inconsistent with expected sales.
- Menu: prices and availability are updated separately on each app when something changes.
- Problems recorded: (1) hard to connect daily orders to the final credit after commissions, discounts, taxes and adjustments; (2) small order-level differences go unnoticed unless someone compares statements with deposits; (3) disputes are slow, since the order, the supporting information and the follow-up with support all have to be found; (4) price or availability changes do not reach both apps and in-house channels at the same time; (5) checks depend on the owner remembering to do them.

Restaurant B: snacks, sweets, Chinese, vegetarian main course, 1 outlet. Operations handled by a manager.

- Scale: about 110 orders a day at an AOV of ₹300, the highest volume of the three. Swiggy 30, Zomato 25, other channels 55 (half of all orders).
- Process: staff monitor orders and manage preparation through the apps. The owner or manager coordinates menu changes, pricing, availability and promotions across both apps.
- Settlement: reports are reviewed periodically and bank credits are checked against what the apps were expected to pay. Refunds, cancellations and deduction queries are handled by hand once an issue is spotted.
- Menu: sweets, snacks, Chinese, vegetarian and main-course items mean many categories, prices and availability rules to maintain.
- Problems recorded: (1) high order volume makes settlements complex to reconcile; (2) restaurant-funded discounts, platform charges and refunds are hard to separate when looking at net receipts; (3) items sold out in the kitchen stay orderable online, leading to cancellations, substitutions or refunds; (4) repetitive menu updates across apps; (5) at peak times staff put fulfilment ahead of updating availability or documenting disputes.

Restaurant C: snacks, Chinese, Italian, non-veg, 1 outlet. Operations handled by a manager.

- Scale: about 60 orders a day at an AOV of ₹450, the highest of the three. Swiggy 25, Zomato 20, other channels 15 (a quarter of orders).
- Process: staff run orders and kitchen preparation through the apps. The owner or manager watches platform sales and reviews settlement summaries when payouts arrive.
- Issues: complaints, cancellations, refund adjustments and payment discrepancies are taken up one at a time through platform support. Prices, availability and promotions are managed in each app.
- Channels: direct, takeaway and in-person sales are recorded separately, so comparing channels means a manual review.
- Problems recorded: (1) three of every four orders come from the two apps, so platform issues hit hardest here; (2) a high AOV does not show profitability when commissions, promotions and refunds

are not tracked per order; (3) food already prepared for a cancelled order costs ingredients and labour with no matching revenue, and the total is not tracked; (4) repeated deductions and complaints are treated as one-offs, not patterns; (5) separate platform and in-house records make net proceeds by channel hard to compare.

B. Calculations

All figures follow from the working sheet's inputs. Formulas: revenue per day = orders per day × AOV; revenue per month = revenue per day × 30; other channels = orders per day − Swiggy − Zomato; aggregator order value per month = (Swiggy + Zomato orders) × AOV × 30.

	Restaurant A	Restaurant B	Restaurant C	Total
Revenue per day (₹)	70 × 270 = 18,900	110 × 300 = 33,000	60 × 450 = 27,000	78,900
Revenue per month (₹)	5,67,000	9,90,000	8,10,000	23,67,000
Swiggy + Zomato orders per day	45	55	45	145
Share of orders on apps	45 / 70 = 64.3%	55 / 110 = 50.0%	45 / 60 = 75.0%	145 / 240 = 60.4%
Swiggy order value per month (₹)	25 × 270 × 30 = 2,02,500	30 × 300 × 30 = 2,70,000	25 × 450 × 30 = 3,37,500	8,10,000
Zomato order value per month (₹)	20 × 270 × 30 = 1,62,000	25 × 300 × 30 = 2,25,000	20 × 450 × 30 = 2,70,000	6,57,000
Aggregator order value per month (₹)	3,64,500	4,95,000	6,07,500	14,67,000

Cross-checks: 8,10,000 + 6,57,000 = 14,67,000. Aggregator value of ₹14,67,000 is 62.0% of ₹23,67,000. Swiggy carries 80 of the 240 daily orders (33.3%) and Zomato 65 (27.1%).

C. Assumptions

1. **Reported figures:** orders per day, AOV and channel counts are the owner's or manager's estimates from the visit. I did not check them against platform dashboards.
2. **Orders per day** is total orders across all channels, so "other channels" includes in-person, takeaway and direct orders. The sheet's column header says "ex in-person", but its formula subtracts only Swiggy and Zomato from total orders, so I treat the remainder as every non-app order.
3. **AOV is the same across channels** within a restaurant, because I have one AOV per restaurant. In practice dine-in and delivery AOVs differ, so the aggregator values are indicative.
4. **Revenue is gross order value,** before commissions, discounts, taxes and refunds. It is not net income.
5. **A month has 30 days,** every day is a typical day, and there is no seasonality or weekday effect.
6. **Settlement review cadence** was described as "periodic". I did not record a fixed schedule for any restaurant.
7. **Leakage rates of 1%, 2% and 3%** are scenarios chosen to size the opportunity. They are not observed. The first test in Step 3 replaces them with a measured figure.
8. **Leakage is applied to gross aggregator order value,** since commission rates and charge structures were not collected.
9. **Three restaurants are a small, informal sample** of single-outlet restaurants with 60 to 110 orders a day. They cannot be generalised to chains or to the US.