PRD - Swiggy's "Anjali/Ankush" - A Contextual GenAI Assistant

Introduction: The Bigger Indian Q-Comm Picture

According to CLSA, the GOV of quick commerce players is expected to reach an astounding $10 billion by FY26. The market has grown exponentially over the last few years, but so has the competition. As the race for quick commerce gets more cutthroat, standing out is becoming tougher but also more important than ever.

One strategy to beat the competition is the good old "Super-app" game plan. With single offerings, it's easier for customers to switch - a new app comes along with slightly better pricing, a little more variety, and poof! your customers are gone. But with super-apps - customers become entrenched into your ecosystem, which significantly increases their switching cost. **Swiggy** with its food delivery, instamart, scenes, dineout - has been trying to follow a similar path. I am a fan of this approach as I think it unlocks a lot of synergy. Global players such as Grab, Deliveroo, WeChat, etc. have all followed a similar approach and the resulting ecosystems have worked wonders.

While the super-app offers a host of advantages, it also comes with significantly tougher execution, which is also the same reason it's tougher to beat. It tends to be chaotic, often feels like 'jack of all trades, master of none' and as a user, it may not feel any more useful than a standalone app. Even for Swiggy in its current state, users treat it like four separate apps in one and open the app with a single intent. GenAI however, with its data synthesis, second order thinking, and scoped outputs - could actually be the key to solving this problem of lack of 'super-app synergy', which is what I want to focus on in this PRD.

Problem Detailing: What problem would this solve? Why should we build it?

- The key problem we are trying to solve is "siloed user intent". In my experience, users still treat Swiggy like four separate apps housed in one container and open the app with a single mission. E.g. I would open the app for either food delivery, Instamart or dineout, but there is very rarely an order from multiple Swiggy offerings at the same time.
- There is huge untapped potential for synergies, for example the below use cases:
    - Complex, situational planning. E.g. Suppose a user is planning a game night for 4 people. Swiggy has so much to offer in this case - Food Delivery order for the main meal, Instamart for urgent supplies (ice, mixers, chips), and maybe even relevant offers from **Dineout** or **Scenes** if the user wants to go for an outing instead.But currently, navigating and thinking of all of this on their own is tough for the user
    - Habit nudging in line with user priorities. E.g. Some users who frequently request cutlery with food delivery, could benefit from purchasing a small, sustainable set from Instamart, or a user who frequently buys sourdough items from restaurants might appreciate a larger, Instamart sourdough loaf. However, currently there is no mechanism in place to give such suggestions

- The idea is to help users navigate offerings based on their current intent and long-time preferences, and enable cross-selling from various sub-apps housed in Swiggy

Key Metrics:

Outlining the key metrics that would move as a result of solving the above problem statement via building “Anjali/Ankush” (live AI assistants):

Metric	Measurement	Importance
Average Order Value (overall north star)	The average total transaction value per user per month	Measures overall top line success of cross-selling and upselling
Cross-Service Attach Rate	The percentage of successful orders that include items from two or more distinct Swiggy verticals	Indicates the user acceptance and utility of an ‘integrated’ offering
Assistant-triggered AOV (north star for this feature)	The average total transaction value in sessions where the Assistant was engaged and provided a suggestion	Useful for untangling the effectiveness of the AI assistant; will need controls to isolate impact
Conversion Rate: AI Suggestion -> Added to Cart -> Ordered	The percentage of AI suggestions that are ultimately accepted and ordered	Indicates the accuracy of assistant suggestions and will eventually drive all the above metrics

Solution Detailing: **The GenAI Advisor**

- First, I will give an overview of the solution and how it would be structured, followed by a deeper dive into the “AI elements” of the assistant.
- *Overview of the solution:*
  - “Anjali/Ankush” is a unified, optional chat interface that sits on top of the entire Swiggy app. Its purpose is to analyze the user's intent (either through the search bar or through the assistant chat window) and connect it to multiple possible fulfillment paths across our ecosystem.
  - The fulfillment paths can be determined through various methods:
    - i) an explicitly stated user objective such as “eating healthy”
    - ii) in-built workflows to increase future user delight by anticipating future needs based on current cart (e.g. suggest adding ‘sourdough’ from Instamart if they add sourdough in a restaurant, since it is likely that the user may want to have this item again in the future), and,
    - iii) intermediate products/services that may bridge the gap between what is available on one offering v.s. what the user truly desires (e.g. a customer

who has a tendency to ask for cutlery at the same location >10 times, may benefit from ordering a small cutlery set from Instamart).

    - The Assistant's final output is ideally a comparative, actionable recommendation that solves the user's underlying objective, transforming the app experience from a collection of isolated services into a cohesive platform

- *Key GenAI elements to the solution:*
    - Past user behaviour mapping and contextualisation to help the underlying LLM form a cohesive understanding of the user's patterns across services. It will also help create an early view of the key use cases the user opens the app for. Further, we can load metadata like "time spent on discovery or time spent on cart" to identify existing patterns. This can help potentially fine tune future recommendations by creating archetypes across our millions of users (e.g. if a user consistently spends little time on picking a restaurant, but lots of time on the menu, we know that they are sure about which cuisine/restaurant but like to ponder upon the dish)
    - Real-time intent mapping through direct and indirect means. Direct is if the user prompts their intention in the chat window. Additionally, direct will also include the user responding to prompts around their priorities (e.g. ranking the importance they give to cost, health, rating, speed, etc.). Indirect is triangulating the user's intent through a combination of search, offerings opened, filters applied and current cart. These will be combined and along with the past understanding of the user, the LLM will create specific workflows to help the user in the current session.
    - Offering intelligent suggestions by parsing through real-time supply side data across all offerings. Each suggestion will have an option to add to cart, and a small menu with three options: i) "Why this?" - which provides an explanation about why it was recommended, how it would be good for the user, and also what prior data helped draw this conclusion; ii) "Show more" - offer more similar suggestions; iii) "What else?" with an option to tweak certain characteristics through a prompt.

This is a win-win for the user and our business - By intelligently cross-selling items based on the user's preferences (e.g. user wants to optimise for healthy options, despite price or time to prepare), the user has a more personalised experience and is able to find what they want efficiently. At the same time, key business metrics would move for Swiggy, there would be a stronger data-backed understanding of our user's behaviour and we would reduce redundancies across our different offerings.