

1. Smart Research Assistant

Problem Statement: Whether it's preparing an assignment, writing a project report, or doing market research for a startup idea, people spend hours Googling and reading PDFs. The problem is that current tools are fragmented:

- Search engines return pages of links, not concise answers.
- Chatbots trained on PDFs only work with one file and can't connect multiple sources.
- Summarizers often skip citations, making it hard to trust the answer.

This leads to shallow knowledge, wasted time, and risk of using outdated info. Students, teachers, and startup teams all face this problem.

The missing piece is a **Smart Research Assistant** — an AI agent that:

1. Takes a question as input.
2. Finds answers from uploaded files or online data.
3. Summarizes into a short, reliable report with citations.
4. Keeps itself updated with fresh information (news/blogs).

This saves time, gives trustable insights, and can grow into a SaaS tool where users pay per report.

Objective: Build a **Research Assistant Agent** that generates structured, evidence-based reports with fresh data.

Expected Deliverables:

- A frontend app where users type questions.
- Backend agent pipeline that:
 - Searches uploaded files + a live data source.
 - Extracts and summarizes key insights.
 - Adds citations for trust.
- Dashboard showing: key takeaways, sources, and usage count.

Flexprice Integration Requirement:

- Bill per **question asked**.
- Bill per **report generated**.
- Show counters inside the app (e.g., "3 reports generated → 3 credits used").

Pathway Integration Requirement:

- Use Pathway to **incrementally ingest one live source** (e.g., mock news/blog updates).
- Show how answers refresh when new info is added.

Evaluation Criteria:

- Depth and clarity of reports.
- Ease of use for non-technical users.
- Visible and correct usage tracking.
- Bonus for live/freshness integration.

2. Smart Doc Checker

Problem Statement: Every organization — whether it's a college club, a company, or even a startup — has multiple documents: rules, contracts, policies, project guidelines. Over time, contradictions appear:

- A project guide says "submit before 10 PM" while the rules doc says "midnight."
- An HR handbook says "2 weeks' notice" while the employment contract says "1 month."
- A college circular says "attendance required: 75%" but another circular says "minimum 65%."

Manually checking such conflicts is boring, time-consuming, and error-prone. Mistakes here can lead to disputes, penalties, or confusion among students and employees.

What's missing is a **Doc Checker Agent** that:

1. Lets users upload multiple docs.
2. Automatically finds contradictions and overlaps.

3. Suggests clarifications to fix conflicts.
4. Monitors external updates (like a policy page) to highlight new issues.

Objective: Build a **Smart Doc Checker Agent** that scans multiple docs, flags contradictions, and reports issues clearly.

Expected Deliverables:

- File upload portal for 2–3 docs.
- AI highlights conflicts with explanations.
- Auto-generated report listing all contradictions.
- Usage counter for number of docs checked/reports made.

Flexprice Integration Requirement:

- Bill per **document analyzed**.
- Bill per **report generated**.
- Show usage count visibly in the app.

Pathway Integration Requirement:

- Use Pathway to **monitor an external doc/policy update** (mock example: “college rule page”).
- Trigger conflict detection when update arrives.

Evaluation Criteria:

- Accuracy of flagged contradictions.
 - Simplicity and clarity of UI.
 - Proper Flexprice billing.
 - Bonus for showing live external updates.
-

3. Stock Market Consultant Agent

Problem Statement: Stock markets are confusing for beginners. Apps like Groww or Zerodha show prices, but they don't explain what a beginner should actually do. Students or small investors want simple advice like:

- “You already hold TCS. Risk is high — consider reducing.”
- “Your portfolio lacks IT stocks — add Infosys.”

Currently, they either rely on friends, half-baked YouTube videos, or make random decisions.

What's missing is a **Stock Consultant Agent** that:

1. Takes user's portfolio as input.
2. Uses live/updated stock data.
3. Suggests buy/sell/hold or diversification advice in simple terms.
4. Tracks usage so users can see how many questions they asked or reports generated.

Objective: Build a **Stock Consultant Agent** that provides actionable advice in plain English.

Expected Deliverables:

- UI for users to enter portfolio (list of stocks).
- Agent generates simple advice (buy/sell/hold/diversify).
- Usage counter for advice/queries.

Flexprice Integration Requirement:

- Bill per **advice generated**.
- Bill per **portfolio analyzed**.
- Show usage clearly (e.g., “2 portfolios analyzed today”).

Pathway Integration Requirement:

- Use Pathway to **fetch mock stock data** (CSV or sample API).
- Ensure portfolio analysis reflects updated data.

Evaluation Criteria:

- Simplicity and clarity of advice.
 - Correct usage billing integration.
 - Bonus for live market updates via Pathway.
-

4. CFO Helper

Problem Statement: Managing money is tough — whether for a startup founder, a student fest organizer, or a small business owner. People ask questions like:

- “If I hire 2 more engineers, can I survive 6 months?”
- “If we spend ₹10,000 extra on marketing, what’s left for prizes?”
- “If I increase product price by 10%, what happens to profit?”

Most use Excel sheets, but those are too basic and don’t show clear “what if” outcomes.

What’s missing is a **Finance Helper Agent** that:

1. Lets users adjust inputs (spend more, hire more, increase prices).
2. Shows the effect on budget or runway with charts or text.
3. Generates a small report they can share.

Objective: Build a **CFO Helper Agent** that simulates budget scenarios and gives clear outcomes.

Expected Deliverables:

- Simple UI with sliders (spending, pricing, hiring).
- Forecast shown in text or chart form.
- Option to export/share a short report.
- Usage counter (how many scenarios tested).

Flexprice Integration Requirement:

- Bill per **scenario simulated**.
- Bill per **report exported**.
- Show usage count in-app.

Pathway Integration Requirement:

- Use Pathway to pull **mock financial data** (like expenses or revenue updates).
- Show how fresh data changes outcomes.

Evaluation Criteria:

- Clarity of forecasts.
- Usability of sliders and reports.
- Proper Flexprice billing.
- Bonus for live data updates with Pathway.