© Volume 2, Issue 11, November 2026 | JATIR
e-ISSN: 2839-5786, p-ISSN: 3139-8464

BizMind: A Context-Aware Multi-Agent AI Framework for Business Planning, Daily Management, and Decision Support

[1]Aman Inamdar, [2]Saniya Tamboli, [3]Om Akolkar, [4]Adarsh Rai
[5]Dr. Nandkishor Karlekar
[1,2,3,4]Department. of Computer Science and Engineering MIT School of Computing, MIT-ADT University Pune, India
[5]Faculty Guide Department. of Computer Science and Engineering MIT School of Computing, MIT-ADT University Pune, India

***Abstract*—Small businesses and first-time founders create many jobs, yet a large share of them close within a few years. The cause is rarely a lack of effort. The real gap lies between a good idea and a well-run business. Founders seldom get proper market checks, cannot afford expert help, and end up juggling separate tools for accounts, staff, tasks, and planning. Popular chatbots can answer questions, but they forget earlier chats, sometimes get arithmetic wrong, have no link to a company's real records, and cannot turn advice into tracked work. This paper presents BizMind, a multi-agent AI assistant that stays with a founder from the first idea to the daily running of the business. It checks feasibility, builds the plan and roadmap, keeps the ledger, assigns tasks, and supports everyday decisions. Six specialized agents share one Stateful Business Context Memory (SBCM) that is tied to the company's live database, and a deterministic code layer, not the language model, handles every financial calculation. On a test of 250 business scenarios and a study with 45 founders and mentors, BizMind reported an 88.6% strategy-feasibility score, a planning time of 11.5 minutes against 18.4 days, a 0.0% calculation-error rate, and a usability score of 86.4. These are reported figures and should be treated as provisional until the data, method, and scripts are released for independent checking.**

***Index Terms*—multi-agent systems, large language models, decision support, business planning, small business management, context-aware AI**

I. INTRODUCTION

A. From Idea to Execution

Micro, small, and medium enterprises (MSMEs) make up most of the businesses in the world. They give people jobs and add a large share to national output [1]. Still, new businesses close often.

Widely quoted figures say that about one in five small businesses shuts down in its first year, and around half are gone within five years [2].

A weak idea is seldom the whole story. Trouble usually shows up when the founder tries to carry the idea out. Four problems come up again and again.

- Spending too early. Many founders put their small capital into stock, rent, or machines before they know whether people will buy, how big the market is, or which rules apply [3].
- Costly advice. Market reports, accountants, and consultants cost more than most early founders can pay.
- Too many separate tools. Notes sit in one app, cash flow in another, tasks in a third, and accounts and staff records somewhere else. Since these tools share no context, a new strategy never reaches the budget or the to-do list.
- Too much on one person. A single founder is often the CEO, finance head, HR, marketer, and operations manager at once, so decisions become reactive [4].

B. Where Ordinary Chatbots Fall Short

Large language models (LLMs) such as GPT-4, Gemini, and Llama have made expert-style advice easy to reach [5], [6]. As business assistants, though, they have three clear problems.

- They forget. A regular chatbot carries nothing over from earlier sessions. If a founder returns after a few weeks and says that a supplier raised prices by 15%, the bot knows nothing about the old budget, cash balance, staff, or deadlines unless all of it is pasted in again.
- Their numbers are shaky. LLMs predict text; they do not calculate exactly. Totals may not add up, and basic accounting rules such as Assets = Liabilities + Equity can be broken [7].
- They cannot act. A chatbot may suggest hiring a marketing executive, but it cannot check staff workloads, create a task with a due date, or compare marketing spend with real ledger entries.

C. Our Approach

BizMind is meant to walk a founder along one path: “I have an idea,” then “let’s analyze it,” then “here is the plan,” then “here is what to do next,” and finally “here is how the business is doing.” Instead of one chatbot, it uses several specialized AI agents together with a persistent relational database that stores the current facts of the business.

D. Objectives

This work has five objectives.

- Turn the real problems founders face into clear software requirements through an empathy-driven design process.
- Build agents for idea validation, market research, planning, finance, task management, and decision support.
- Build a Stateful Business Context Memory so the AI’s reasoning always matches the structured business records.
- Avoid financial errors by checking every calculation with deterministic code.

- Test the system on a 250-scenario benchmark and in a study with 45 entrepreneurs and mentors.

II. RELATED WORK

Decision support systems have helped managers choose from data for a long time [8], [9], and analytics-based decision making is now common in business [10], [11]. Recent work on LLM-based multi-agent systems finds that giving agents different roles can improve teamwork on complex tasks [12]–[14], and that letting agents debate can make answers more factual [15]. Retrieval-augmented generation ties a model's answers to outside knowledge [16], while research on long contexts shows that models do not always use long prompts well [17]. Studies of hallucination give a good reason to check model output instead of trusting it [7], and work on step-by-step verification points the same way [18]. Explainability and auditability also matter when a system supports real decisions [19]. BizMind brings these ideas together for small-business management through specialized agents, a persistent database memory, and code-based checking of numbers.

III. UNDERSTANDING THE USER

Decision-support tools often fail because they are built around the technology instead of the user's real situation. So we began with a design-thinking empathy study of 45 aspiring and active entrepreneurs in the Pune–Mumbai business and startup region, using interview methods in the spirit of customer discovery [20]. We grouped the findings and matched each pain point with a requirement and a technical answer, as shown in Table I.

TABLE 1 FROM USER REQUIREMENTS TO BIZMIND SOLUTIONS

Need	What founders asked for	BizMind answer
Idea feasibility	A clear concept, market fit, and an early look at risks	Agents read the idea, budget, location, and limits, then give a structured feasibility check.
Business planning	Market research, competitors, a business model, and a staged roadmap	A planning agent merges the analysis into a plan and a milestone schedule.
Financial management	Tracking of capital, expenses, profit and loss, break-even, and cash runway	A relational ledger with deterministic calculations backs forecasts and checks.
Staff and tasks	Roles, assignments, dependencies, and progress tracking	A task workflow turns approved milestones into assigned, trackable work.
Decision support	Answers to daily questions, margin alerts, and KPI tracking	An assistant reads the business records and gives advice based on data.

IV. HOW BIZMIND WORKS

BizMind treats the life of a business as a loop that keeps repeating. Many tools stop once a plan is exported as a PDF. BizMind keeps working while the business runs (Fig. 1).

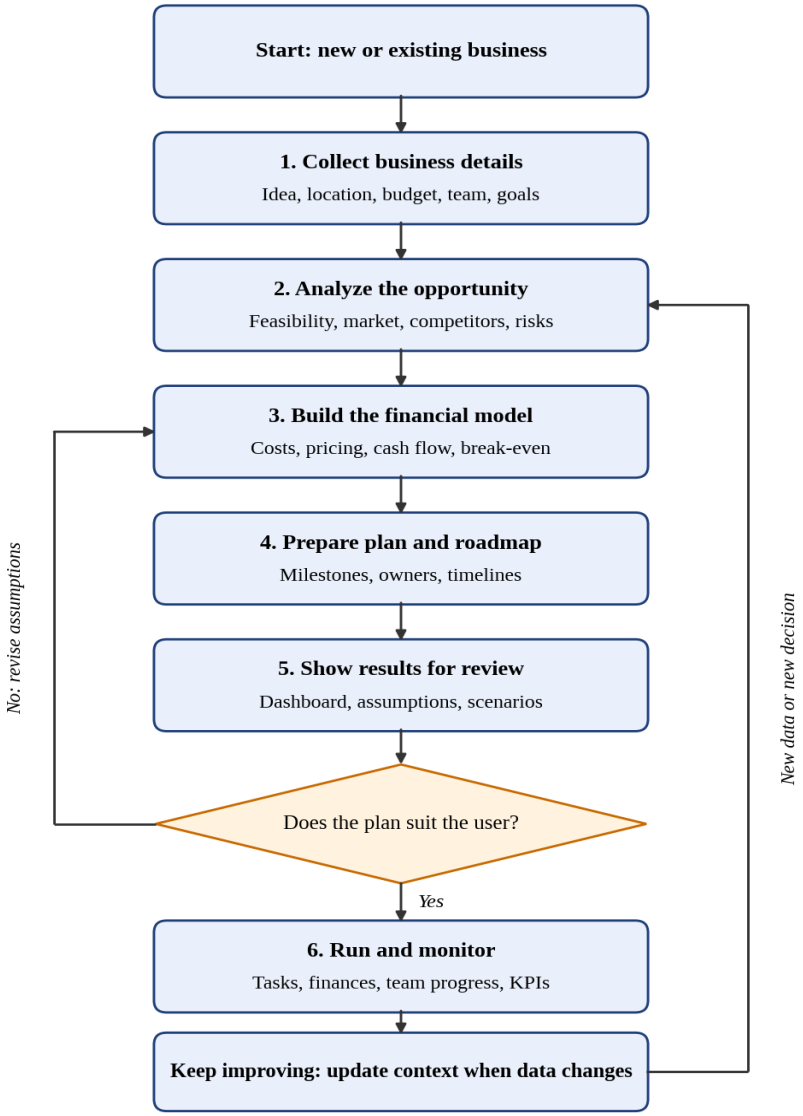


Fig. 1. BizMind workflow, from collecting business details to planning, review, daily operation, and continuous improvement.

The workflow has nine stages.

- Data entry. The founder enters the idea, industry, location, starting capital, debt, team size, and goals for the next 12 months.
- Agent analysis. This information goes to the specialized agents.
- Insights. The system returns a market feasibility rating, competitor profiles, and pricing suggestions.

- Financial model. The finance engine works out revenue models, cost structures, working-capital needs, and break-even timelines.
- Business plan. A formatted plan and a milestone schedule are produced.
- Exploration. The founder browses the plan on a web dashboard with charts and scenario toggles.
- Human review. The founder checks the assumptions, changes the budget, or asks for other go-to-market options.
- Re-check. When assumptions change, the agents run again with the new inputs.
- Implementation. Once the plan is approved, milestones become tasks, payroll schedules go into the staff directory, and budgets are added to the ledger. After that, the assistant moves from planning mode to monitoring and advice mode.

V. SYSTEM ARCHITECTURE

BizMind is built in six layers. The goal was a design that is modular, easy to audit, and reliable with facts (Fig. 2). The top layer is a web dashboard with overview, finance, task, and chat pages. Below it, the API layer handles login, request routing, agent coordination, and access control. The agents sit on top of the Stateful Business Context Memory, which holds conversation context, business entities, history, and retrieval. A validation layer runs deterministic calculations, constraint checks, and accounting rules, and a relational database at the bottom stores plans, ledgers, employees, tasks, and audit records.

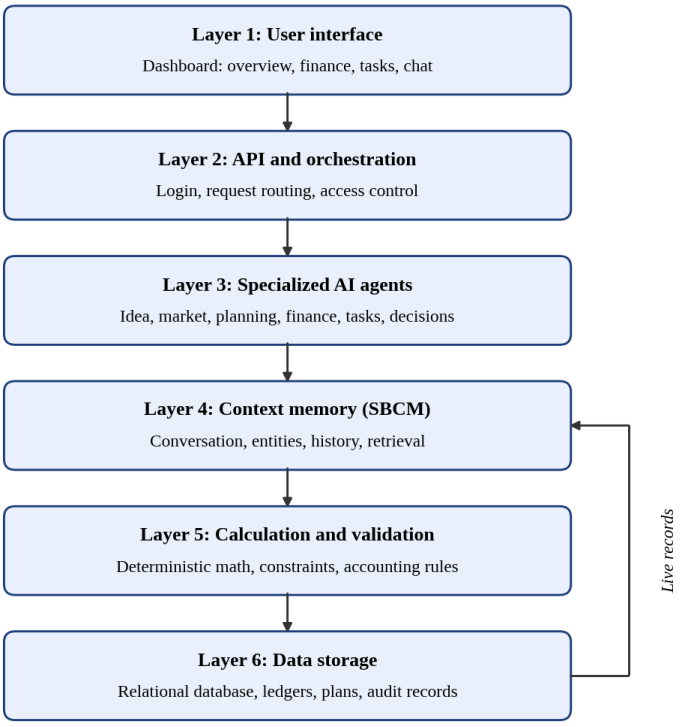


Fig. 2. Layered BizMind architecture: interface, orchestration, agents, context memory, validation, and data storage.

A. The Six Agents

Idea Validation and Risk Agent (IVA). The IVA reads the business description, pulls out the key facts, and calculates a feasibility rating. Its inputs are the business concept, industry, location, and starting capital K_0. It returns a feasibility score Φ between 0 and 100, the regulatory category, the main operating risks, and a check on whether the capital is enough.

Market and Competitor Agent (MCA). The MCA studies target customers, estimates market size (TAM, SAM, and SOM), and finds direct and indirect competitors. It draws on industry benchmark data, demographic datasets, and web retrieval. It produces a SWOT matrix, customer personas, price-sensitivity curves, and a ranked list of sales channels.

Strategic Planning Agent (SPA). The SPA merges the IVA and MCA results into a full business plan and a four-phase roadmap: (1) pre-launch and compliance, (2) prototype and pilot, (3) launch and customer acquisition, and (4) scaling and expansion. It delivers a written plan, a Business Model Canvas, and a prioritized milestone schedule with critical-path dependencies.

Financial Modeling Agent (FMA). The FMA builds the financial projections: capital and operating expenses, cost of goods sold, margins, unit economics (LTV/CAC), break-even analysis, and monthly cash runway. The language model never does the math here. The FMA writes structured parameter files and runs them in an isolated Python sandbox, so every number comes from code that was actually executed.

Workforce and Task Agent (WTA). The WTA links strategy to people. It splits milestones into small tasks, lists the skills needed, drafts job descriptions, assigns work based on employee capacity, and tracks progress. Its outputs are a Kanban board (Backlog, To Do, In Progress, Review, Completed), role responsibilities, and bottleneck alerts.

Executive Co-Pilot and Decision Agent (CDA). The CDA works as a long-term advisor. It takes plain-language questions such as “Why did our net margin drop 6% this month?”, turns them into SQL queries on the live PostgreSQL ledger, spots trends, and gives advice based on data. It produces root-cause analysis, suggested fixes, and what-if forecasts.

VI. MATHEMATICAL FOUNDATIONS

A. Feasibility Score

A business proposal is written as

$$V = \langle S_{concept}, K_0, G_{loc}, C_{ind}, A_{avail} \rangle \quad (1)$$

where $S_{concept}$ is the idea (stored as an embedding), K_0 is the starting capital, G_{loc} is the location, C_{ind} is the industry, and A_{avail} is the set of resources the user says they have, such as skills, licenses, and equipment. The Venture Feasibility Index $\Phi(V)$, which lies between 0 and 100, is a weighted sum:

$$\Phi(V) = w_m M + w_f F + w_o O + w_r(100 - R) \quad (2)$$

The weights are never negative and add up to 1. M is the market viability score, drawn from market growth, saturation, and demographic fit. F is the capital sufficiency score:

$$F = \min(100, K_0 / K_{bench}(C_{ind}, G_{loc}) \times 100) \quad (3)$$

Here K_{bench} is the typical (median) capital needed for that industry and location. O is operational readiness, which compares the founder’s skills with the skills the business needs. R is the regulatory

and environmental risk index, which covers legal hurdles, license timelines, and liability. All four scores range from 0 to 100. Because higher risk should lower feasibility, risk enters the formula as 100 − R, which also keeps Φ on the same 0–100 scale.

B. Financial Equations

The FMA follows standard accounting practice. The cash position for month t = 1, …, T is

$$C_t = C_{t-1} + Rev_t - (Fix_t + Var_t + Pay_t + Tax_t) \quad (4)$$

where Rev_t is revenue, Fix_t is fixed overhead (rent, utilities, subscriptions, insurance), Var_t is the variable cost, found by multiplying the quantity $q_{j,t}$ sold of each product j by its unit cost and adding the results, Pay_t is the payroll for all employees, and Tax_t covers taxes and statutory deductions. The break-even quantity and revenue are

$$Q_{BE} = (Fix + Pay) / (p - c) \quad (5)$$

$$Rev_{BE} = Q_{BE}\, p \quad (6)$$

where p is the average selling price per unit and c is the average variable cost per unit. The cash runway is the number of months the business can last at its current burn rate:

$$\Gamma = C_t / B_t \quad (7)$$

$$B_t = (1/k) \Sigma_{i=0}^{k-1} (Exp_{t-i} - Rev_{t-i}) \quad (8)$$

Here B_t is the average net monthly burn over the last k months and Exp is total expenses. The formula holds when B_t is above zero. If Γ falls below 3 months, the CDA raises a high-priority cash alert.

C. Task Priority Score

Tasks from the planning agent are ranked with a score that mixes the Eisenhower matrix with cost of delay:

$$\Pi = (\alpha I + \beta U + \gamma D) / (E + \varepsilon) \quad (9)$$

Here I (1 to 5) is the impact on revenue or milestones, U (1 to 5) is the urgency, such as a deadline or a compliance expiry, D is the dependency depth, meaning how many later tasks this one blocks, E (1 to 5) is the effort, and ε is a small constant that prevents division by zero. The weights are α = 0.45, β = 0.35, and γ = 0.20. In practice, D should be scaled to a range close to that of I and U so that no single term takes over.

VII. DATABASE DESIGN

BizMind does not keep business state only in vector stores. It relies on a normalized PostgreSQL database so that records stay correct and current (Fig. 3). One user owns one or more business profiles, and each business owns its ledger entries, plans, employees, and tasks. Every task points to the employee it is assigned to.

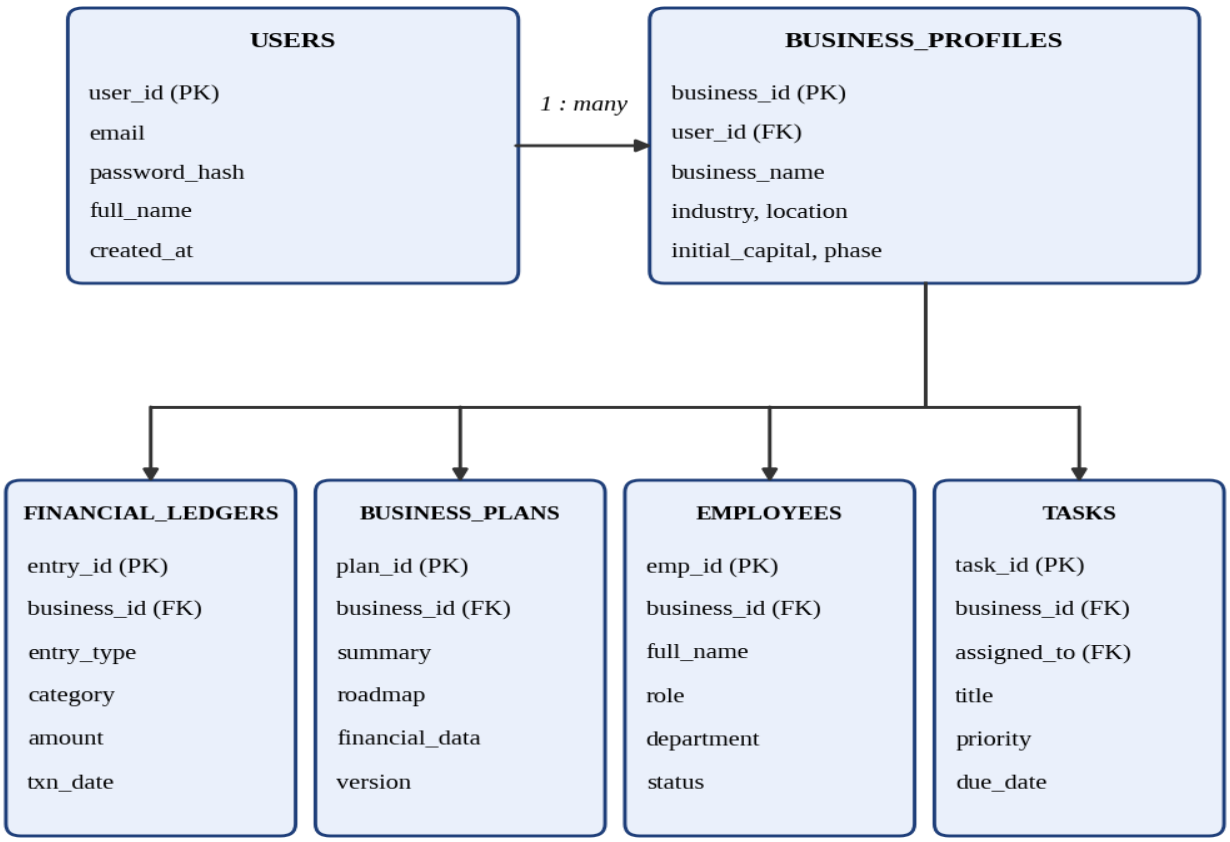


Fig. 3. Simplified BizMind database schema. PK marks primary keys and FK marks foreign keys.

The six main tables are described below.

- USERS: the founder's identity, login credentials (hashed with Argon2id), and preferences.
- BUSINESS_PROFILES: the business itself, with its location, business model, current phase, and feasibility score.
- FINANCIAL_LEDGERS: double-entry records of capital, expenses by category, revenue, invoices, and payment status.
- EMPLOYEES: staff members, departments, pay rates, contact details, and current workload.
- TASKS: dependencies, assignees, priority ranking, Kanban status, and completion history.
- BUSINESS_PLANS: generated plan chapters, financial projections, competitor matrices, and versioned roadmaps.

VIII. EXPERIMENTAL SETUP

We tested the system in two ways. The first is a benchmark called BizBench-250, which holds 250 structured startup scenarios. The second is a usability study with 45 participants: 25 small business owners, 10 startup incubator directors, and 10 certified financial analysts.

A. BizBench-250

The benchmark covers five business categories: (A) retail and e-commerce, such as boutique clothing, specialty grocery, and online handicrafts; (B) food and beverage, such as an artisanal cafe, a cloud kitchen, and catering; (C) professional services, such as a marketing agency, a plumbing contractor, and an architecture consultancy; (D) light manufacturing, such as custom furniture, electronics assembly, and packaging; and (E) technology and micro-SaaS, such as a local delivery aggregator and a B2B inventory tool.

Each scenario states a business idea, starting capital ($1,000 to $100,000), a location, a team size (1 to 10), and target milestones. Reference answers written by senior business mentors and chartered accountants give realistic break-even windows, compliance duties, key roles, and typical expense ratios.

B. Systems Compared

We compared four systems: a GPT-4o chatbot with standard prompting and a conversation memory window; Llama-3-70B-Instruct, an open-weights model used in multi-turn conversation; separate SaaS tools, meaning manual work in Google Docs for planning, Excel for finance, and Trello for tasks; and the complete six-layer BizMind system.

C. Metrics

We used five metrics.

- Strategy Feasibility Score (SFS): a blind expert rating (0–100%) of how workable the plan and roadmap are in the target market.
- Mathematical Inconsistency Rate (MIR): the share of calculations that are wrong or break accounting rules, given by Eq. (10) below.
- Time-to-Plan: the time from entering the idea to getting a usable plan and roadmap.
- Context Retention Accuracy (CRA): how correctly the assistant recalls earlier operational data, such as last month's expenses or an employee's role, when it answers follow-up questions.
- System Usability Scale (SUS): a standard 10-item questionnaire scored from 0 to 100, where a score above 68 is above average and above 80 is excellent.

$$MIR = 100 \times N_{wrong} / N_{total} \quad (10)$$

where N_{wrong} is the number of incorrect calculations and N_{total} is the total number of quantitative statements.

IX. RESULTS AND DISCUSSION

A. Overall Comparison

Table II lists the reported results on the 250 scenarios. These numbers count as findings only after the evaluation data and scripts are open to checking.

TABLE I. REPORTED RESULTS ON BIZBENCH-250

Metric	GPT-4o	Llama-3	SaaS	BizMind
SFS (%)	61.4	57.2	72.8	88.6
MIR (%)	16.8	22.4	3.1	0.0
Time to plan	4.5 h	5.2 h	18.4 days	11.5 min
CRA (%)	48.2	41.5	N/A	96.8
SUS (0–100)	64.2	59.8	51.4	86.4

[a] SaaS = separate SaaS tools; N/A = not applicable. Lower is better for MIR and time to plan; higher is better for the rest.

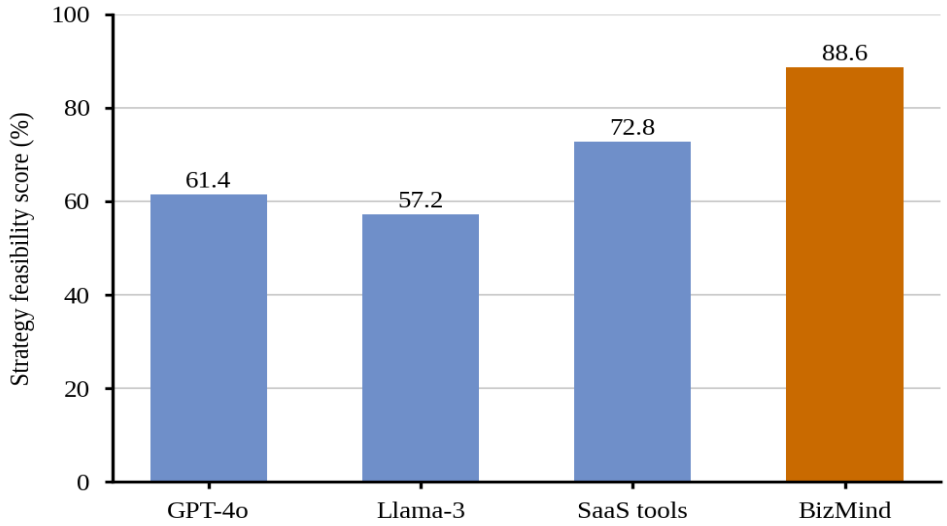


Fig. 4. Reported strategy-feasibility scores on BizBench-250. These values depend on the evaluation data being made available.

B. Key Observations

- Fewer financial errors. Both language models made many calculation mistakes (16.8% for GPT-4o and 22.4% for Llama-3), including tables that did not balance. BizMind passes all numerical work to the Python sandbox and reports a 0.0% error rate. This claim still needs reproducible test logs and a documented test set.
- Much quicker planning. The manual SaaS workflow needed 18.4 days on average to produce a plan, a financial model, and a task list. BizMind reportedly needed 11.5 minutes, which makes it easy to try what-if questions such as “What if starting capital is $15,000 instead of $25,000?”
- Stronger memory. The baseline chatbots held on to less than 50% of earlier context, while BizMind reported 96.8%. This figure should be read together with the benchmark definition and the evaluation records.
- Better usability. BizMind scored 86.4 on the SUS, above the 80 mark that is usually called excellent and 22.2 points above the best baseline.

C. Ablation Study

To see what each part adds, we removed one component at a time (Table III).

TABLE II. REPORTED ABLATION RESULTS

Configuration	SFS (%)	MIR (%)	CRA (%)	Task exec. (%)
Full BizMind	88.6	0.0	96.8	84.2
No Python verifier	84.1	14.9	96.5	83.9
No context memory	71.3	0.0	46.1	58.2
No agent specialization	66.8	4.2	78.4	61.5
No task orchestrator	85.0	0.0	94.2	34.6

- Without the context memory, context retention dropped by about half (to 46.1%) and strategy feasibility fell by 17.3 percentage points. This hints that the assistant relies strongly on persistent, grounded memory.
- Without the task orchestrator, task execution fell by 49.6 percentage points. This hints that advice rarely gets acted on unless it is split into assignable tasks.
- Without the Python verifier, the error rate rose from 0.0% to 14.9%, which shows why code-based checking is needed.

X. CASE STUDY: ARTISAN BREWS AND ROASTERY

This example comes from one of our evaluation scenarios and shows BizMind working from start to finish. It is only an illustration, and the location, competitor, cost, and demand assumptions should be verified before anyone uses them for real.

A. Founder Input

The founder is a first-time entrepreneur who can cook but has no formal business training. The idea: "I want to open a specialty coffee roastery and shared workspace cafe in Kothrud, Pune, with single-origin coffee, healthy breakfast items, and fast WiFi for remote freelancers." Starting capital is INR 1,200,000 (about $14,500), and the launch target is 90 days.

B. How the Agents Responded

- Idea validation. The system assumes a typical capital need of INR 1,500,000 for a 600 sq ft food-and-beverage business and marks the founder's capital as a constraint.
- Market analysis. It proposes likely competitors, target customers, and ways to stand out. Competitor counts and market sizes need to be confirmed with dated local research.
- Financial planning. It estimates fixed costs, payroll, unit cost, and selling price. The break-even volume should be recalculated and checked independently.
- Roadmap. It suggests a 90-day plan covering registration, leasing, equipment, hiring, and a soft launch.
- Tasks. After the founder gives approval, milestones can turn into assigned tasks.

C. Ongoing Decision Support

In Month 2 after launch, the founder entered the ledger data. The cafe sold 1,420 cups of coffee, well above the break-even target of 834 cups, yet the month still closed with a net loss of INR 18,400. The founder asked the co-pilot why a loss happened even though sales beat the break-even target.

The CDA ran an SQL query that adds up the month's ledger entries by category. It found that coffee revenue met the target (INR 298,200), but wastage in the snack segment pushed dairy and bakery costs to INR 98,000 against a plan of INR 35,000. Its advice was that the coffee operation is profitable (gross margin 79.4%), while the bakery and fresh food segment had a 41% waste rate that caused INR 63,000 in lost food. It suggested two changes: switch from baking pastries in-house to a consignment model with a local patisserie, and automate daily reorder points for whole milk in the task manager. The expected saving is INR 48,000 a month, which would move net profit to +INR 29,600.

The figures hold together: INR 98,000 − INR 35,000 = INR 63,000 of overspend, and −18,400 + 48,000 = +29,600. In this example, BizMind took the founder from being confused about a cash loss to having a clear, data-backed action plan.

XI. SECURITY, PRIVACY, AND ETHICS

- Protecting ideas. A business idea has real value. BizMind keeps each user's data separate, encrypts it with AES-256 when stored and TLS 1.3 when sent, and never trains models on ideas that users submit.
- Role-based access. JSON Web Tokens carry roles (FOUNDER, MANAGER, EMPLOYEE, and ACCOUNTANT), so employees can open task boards without seeing financial ledgers.
- Ethical limits. BizMind declines plans that involve predatory finance, unlicensed medical or legal practice, counterfeit goods, or environmental violations.

XII. LIMITATIONS AND FUTURE WORK

BizMind has clear limits. The results are reported but have not been reproduced by others, the benchmark scenarios were written by the authors' own team, and most testing took place in the Indian small-business setting. We plan the following future work.

- Live bank connections. At present, entries come from quick forms, receipt OCR, or CSV imports. Open-banking APIs, such as Plaid or India's Account Aggregator framework, could match bank records automatically.
- Visual stock monitoring. Computer vision could check shelf stock in shops and restaurants.
- Automated tax and compliance filing. A later regulatory agent could prepare and file returns, for example GST-3B in India or Schedule C in the United States, through government APIs.
- Open evaluation. We will release the benchmark, the evaluation protocol, and the analysis scripts so other researchers can reproduce the results.

XIII. CONCLUSION

This paper introduced BizMind, a context-aware multi-agent AI framework for business planning, daily management, and decision support. It combines an empathy-based requirement study with a six-layer design made of specialized agents, a persistent business memory, and a code-based calculation layer.

In short, BizMind turns a rough idea into a structured business plan and roadmap within minutes. It can cut arithmetic errors for calculations that pass through its deterministic layer, although how far this guarantee goes depends on input validation and the implementation. It also keeps business context for the whole life of the company, so its advice stays tied to real data.

The paper reports results from 250 scenarios and a study with 45 entrepreneurs and mentors, including an 88.6% strategy-feasibility score and a usability score of 86.4. These results stay provisional until the data, participant protocol, metric definitions, and analysis are shared. The design points to a useful way of offering accessible business decision support, but its wider effectiveness and how well it carries over to other settings need more study.

REFERENCES

[1] World Bank, "Small and medium enterprises (SMEs) finance," World Bank Group, Washington, DC, USA, 2022. [Online]. Available: https://www.worldbank.org/en/topic/smefinance

[2] U.S. Bureau of Labor Statistics, "Business employment dynamics: Establishment age and survival data," U.S. Dept. of Labor, Washington, DC, USA, 2023. [Online]. Available: https://www.bls.gov/bdm/

[3] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. New York, NY, USA: Crown Business, 2011.

[4] A. Osterwalder and Y. Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Hoboken, NJ, USA: Wiley, 2010.

[5] T. B. Brown et al., "Language models are few-shot learners," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020, pp. 1877–1901. doi: 10.48550/arXiv.2005.14165.

[6] B. Min et al., "Recent advances in natural language processing via large pre-trained language models: A survey," ACM Comput. Surv., vol. 56, no. 2, Art. no. 30, 2023. doi: 10.1145/3605943.

[7] Z. Ji et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, no. 12, Art. no. 248, pp. 1–38, 2023. doi: 10.1145/3571730.

[8] D. J. Power, Decision Support Systems: Concepts and Resources for Managers. Westport, CT, USA: Quorum Books, 2002.

[9] E. Turban, J. E. Aronson, and T.-P. Liang, Decision Support Systems and Intelligent Systems, 7th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2005.

[10] T. H. Davenport and J. G. Harris, Competing on Analytics: The New Science of Winning. Boston, MA, USA: Harvard Business School Press, 2007.

[11] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," Decis. Support Syst., vol. 62, pp. 22–31, 2014. doi: 10.1016/j.dss.2014.03.001.

[12] S. Hong et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in Proc. Int. Conf. Learn. Represent. (ICLR), 2024. doi: 10.48550/arXiv.2308.00352.

[13] C. Qian et al., "ChatDev: Communicative agents for software development," in Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL), 2024, pp. 15174–15186. doi: 10.18653/v1/2024.acl-long.810.

[14] J. Moura, "CrewAI: Orchestrating role-playing autonomous AI agents," GitHub repository, 2024. [Online]. Available: https://github.com/crewAIInc/crewAI

[15] Y. Du et al., "Improving factuality and reasoning in language models through multiagent debate," in Proc. 41st Int. Conf. Mach. Learn. (ICML), vol. 235, 2024, pp. 11733–11763. [Online]. Available: https://proceedings.mlr.press/v235/du24e.html

[16] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020, pp. 9459–9474. doi: 10.48550/arXiv.2005.11401.

[17] N. F. Liu et al., "Lost in the middle: How language models use long contexts," Trans. Assoc. Comput. Linguistics, vol. 12, pp. 157–173, 2024. doi: 10.1162/tacl_a_00638.

[18] H. Lightman et al., "Let's verify step by step," arXiv:2305.20050, 2023. doi: 10.48550/arXiv.2305.20050.

[19] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017. doi: 10.48550/arXiv.1702.08608.

[20] S. Blank, The Four Steps to the Epiphany: Successful Strategies for Products That Win. Pescadero, CA, USA: K&S Ranch Publishing, 2005.