



BankMate: An AI Agent for Banking Operations

Komal Vinayak Kuyte ^a, Aachal Umesh Tayde ^b, Ishwari Rajendra Wankhade ^c, Anuj Shirish Kuralkar ^d, Deep Kanahiya Yadav ^e, Prof. Amol D. Chokhat ^f

^a Student of B.Tech. Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

^b Student of B.Tech. Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

^c Student of B.Tech. Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

^d Student of B.Tech. Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

^e Student of B.Tech. Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

^f Asst. Professor, Department of Computer Science and Engineering, P. R. Pote Patil College of Engineering and Management Amravati-444602, India

ABSTRACT:

Traditional digital banking interfaces often require customers to move through several menus and pages to reach account information, review spending, or explore financial products. BankMate is an AI-powered conversational banking prototype that simplifies access to banking information through natural-language interaction. A centralized chat interface lets users ask banking-related questions and open dedicated features for customer profile information, spending breakdowns, credit card limits, and loan-related services.

The system integrates Google Gemini with the Vercel AI SDK for conversational interaction and is built with Next.js, React, TypeScript, and Tailwind CSS. Structured mock banking data simulates customer, account, and transaction records, while interactive tables, widgets, and a spending pie chart present the information visually. Sensitive operations such as fund transfers require six-digit PIN verification, and guardrails reject unsupported queries. Manual black-box testing of nine scenarios, covering balance inquiry, loan services, fund transfer, PIN validation, and unsupported queries, produced the expected outcomes. BankMate is a simulated prototype and is not connected to live banking infrastructure; secure banking API integration, stronger authentication, and formal evaluation remain future work.

Keywords: *Conversational Banking; Artificial Intelligence; Google Gemini; Vercel AI SDK; Natural-Language Processing; Interactive Banking Interface; Mock Banking Data*

1. INTRODUCTION

Conversational banking refers to the use of text-based or voice-based interfaces through which customers interact with banking services in natural language. Instead of navigating several menus, a customer can submit a question and receive a response through text or interactive visual components. AI-powered assistants and natural-language processing have created new possibilities for accessing digital services [1], [2]. The use of AI in banking extends to customer service, financial analysis, fraud detection, and intelligent automation [3]–[6], while machine learning and deep learning provide methods for learning patterns from data [7], [8]. These opportunities come with challenges: AI-generated responses may contain inaccuracies, and financial information must be handled with care. A conversational banking prototype should therefore distinguish AI-generated explanations from authoritative banking records and state clearly which operations are simulated.

BankMate explores these ideas through a web-based conversational banking prototype. The application provides a centralized chat interface together with dedicated options for My Bank Profile, Spending Breakdown, Credit Card & Limits, and Apply for Loans. The Spending Breakdown feature presents monthly expenditure as a pie chart so that users can see how their spending is distributed. The prototype uses Google Gemini and the Vercel AI SDK for conversational interaction, a Next.js, React, TypeScript, and Tailwind CSS frontend, and simulated banking data. It does not connect to a live banking database, and the loan application interface should not be read as evidence of real loan processing.

1.1 Problem Statement

Traditional banking interfaces organize information through menus, forms, and predefined navigation paths, so users may need to visit several sections to find account details, spending patterns, credit card information, or loan services. Conversational AI offers an alternative, but combining it with structured banking information requires attention to response reliability, information presentation, and the handling of sensitive operations. The problem addressed here is to design and develop a conversational banking prototype that lets users explore simulated banking information through AI-assisted interaction and organized visual components.

1.2 Objectives

The objectives of the project are to develop a centralized chat interface for banking-related queries; explore conversational AI integration using Google Gemini and the Vercel AI SDK; organize simulated banking information into structured application components; provide interface features for customer profiles, spending breakdowns, credit card details, and loan-related information; build the application with Next.js, React, TypeScript, and Tailwind CSS; demonstrate conversational access to banking information without live banking infrastructure; and identify limitations related to AI response reliability, data security, and real-world banking integration.

1.3 Contributions

The contribution of BankMate is a conversational banking prototype that combines a web-based chat interface, an AI integration approach, mock banking data, and interactive information components. It demonstrates an alternative interface for accessing banking-related information while recognizing that secure authentication, live banking integration, and comprehensive response validation would be required for a production-ready system.

2. RELATED WORK

Artificial intelligence is being explored in banking for customer service, fraud detection, credit assessment, financial forecasting, and process automation. Industry reports discuss its potential to support automation, improve operational efficiency, and enable more personalized customer experiences [3], [4], [6]. Machine learning, a principal approach in these applications, is described as the development of programs that improve their performance through experience [8], and deep learning enables the learning of complex representations from data [7]. BankMate does not develop a fraud-detection or credit-scoring model; it focuses on conversational access to banking-related information.

Chatbots are a common form of conversational AI. Traditional chatbots rely on predefined rules, keywords, decision trees, or fixed response templates, and they can struggle with ambiguous questions or unfamiliar wording. AI-enabled systems use language-processing techniques to interpret queries and generate more flexible responses, and research on banking chatbots and AI-driven customer service provides a foundation for these applications [2], [9]. Documentation for conversational assistant platforms describes how assistants can support defined conversational workflows [10]. A chatbot's usefulness depends on understanding user intent, providing relevant information, and communicating its limitations.

Natural-language processing underpins these systems by enabling computers to interpret and generate human language [1]. Large language models, a major category of generative AI [11], are being explored for customer support, document assistance, and information retrieval in financial services [5], [12]. However, a language model may misinterpret a question or produce statements not supported by the available data, which matters when users rely on balances, transactions, or credit limits. Generated text should therefore not be treated as the authoritative source of financial records, and operations such as filtering, sorting, and calculation are better handled by deterministic application logic. Table 1 summarizes the approaches considered.

TABLE 1: COMPARISON OF EXISTING APPROACHES AND BANKMATE

Approach	Main Characteristics	Limitations or Considerations
Traditional banking interface	Menus, forms, dashboards, and predefined navigation	Users may need to visit several sections to locate information
Rule-based chatbot	Predefined rules and responses	May struggle with unexpected wording or complex queries
Conversational assistant platform	Structured conversational workflows	Requires suitable configuration and integration with relevant data
Generative AI assistant	Flexible natural-language responses	May generate inaccurate or unsupported information
BankMate prototype	Chat interface, AI-assisted interaction, mock banking data, and interactive components	Uses simulated data; not connected to live banking infrastructure

2.1 Research Gap

Traditional interfaces rely on predefined navigation, rule-based chatbots are limited by their response patterns, and generative AI raises concerns about response reliability. For an academic prototype, there is value in exploring how these capabilities can be combined in a unified interface without access to live banking infrastructure. BankMate addresses this by offering a unified conversational interface, interactive presentation of simulated banking information, a combination of AI assistance with deterministic application logic, and awareness of privacy and reliability limits.

3. BANKMATE SYSTEM DESIGN AND METHODOLOGY

3.1 System Overview

BankMate follows a modular, component-oriented design that separates the user interface, conversational interaction, banking-related functionality, verification and validation, and application state management. A user enters a query or selects a quick action; the application interprets the request and routes it either to the conversational AI integration or to the relevant banking logic; and the resulting information, confirmation, or error message is shown through chat messages or dedicated interface components (Fig. 1). The development methodology is organized into five modules, summarized in Table 2.

Fig. 1. Conceptual system architecture of BankMate

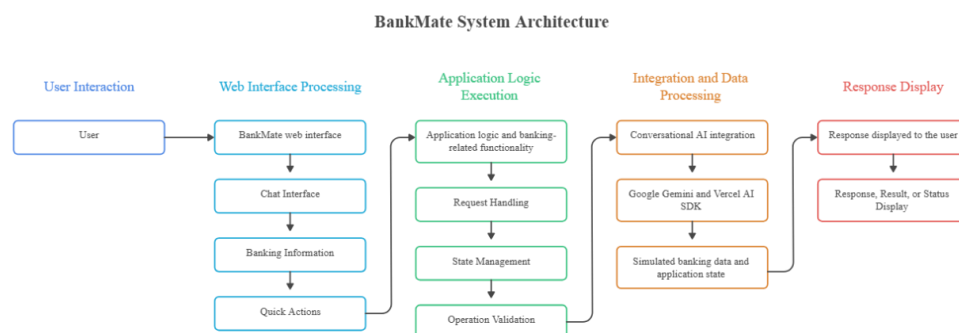


TABLE 2: OVERVIEW OF METHODOLOGY MODULES

Module	Name	Main Activities
1	Requirement Analysis and System Architecture	Identify requirements, define scope, and plan architecture and workflows
2	Mobile-First Chat Interface	Design the chat interface, message input, response display, and banking components
3	Backend Services, APIs and Database Management	Handle application logic, data management, and transaction-related workflows
4	Vercel AI SDK and Google Gemini Integration	Support conversational interaction and process banking-related queries
5	Testing, Evaluation and Deployment	Verify functionality, test security-related controls, and deploy the application

3.2 Workflow Design

A typical interaction proceeds as follows: the user enters a banking-related query or selects an action; the application processes the input; the request is interpreted and routed to the relevant functionality; verification or validation is performed where the operation requires it; the supported request is processed; and the result, confirmation, or error message is presented. This pattern is applied consistently to information inquiries and to supported banking operations.

3.3 Data and State Management

The prototype uses structured mock banking data stored as JSON files for loan catalogues, transactions, and user profiles, with browser local storage used for selected application state. Account data and interface state must remain consistent as users interact with the application. For operations that change state within the prototype, the interface should reflect the resulting information, and validation and error handling should prevent a failed or rejected operation from being shown as successful.

3.4 Conversational AI and Tool Calling

Google Gemini is used through the Vercel AI SDK to interpret natural-language requests, so that the same need, such as checking a balance or calculating an EMI, can be expressed in different ways. Tool calling connects an interpreted request with a supported application function, for example loan calculations, fund transfers, fixed-deposit booking, and loan applications. The AI component supports conversation, whereas sensitive workflows rely on operation-specific validation and PIN verification rather than on generated text alone. Guardrails restrict the assistant to banking-related functionality and politely reject unsupported requests.

4. SYSTEM IMPLEMENTATION

4.1 Frontend Implementation

The interface is implemented with Next.js and React, using TypeScript for typed components and Tailwind CSS for styling and adaptable layouts. It contains the chat area, message input, quick actions, and structured components for customer profile details, spending information, credit card details, and loan-related information (Fig. 2). The component-based structure keeps the presentation consistent across sections, and the mobile-first design emphasizes readable text, accessible controls, and clear navigation.

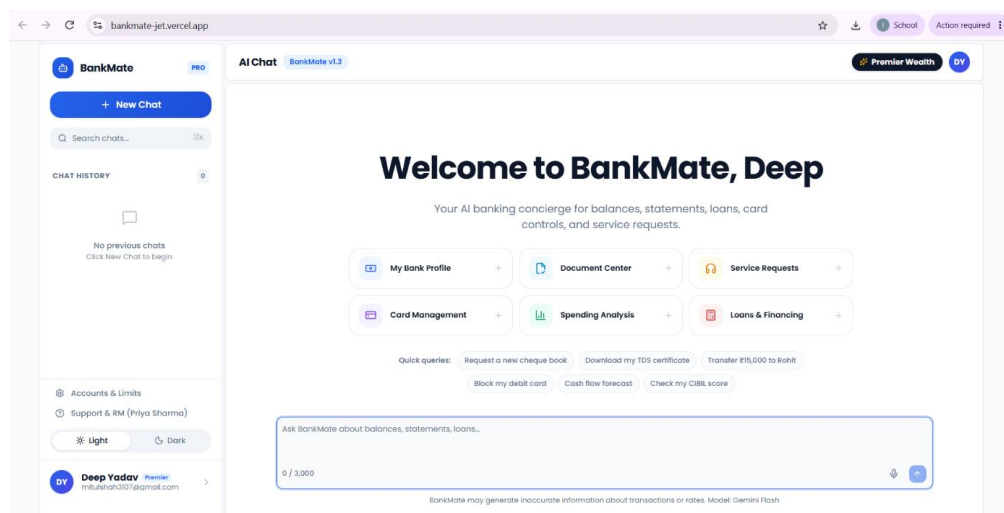


Fig. 2. Main user interface of BankMate

4.2 Banking Operations and Verification

BankMate provides interface support for balance inquiries, loan-related requests, and fund-transfer workflows. Informational requests display the relevant information available to the prototype, and loan-related functionality may involve calculations or an application workflow. The fund-transfer workflow adds a verification step: the application requests a six-digit PIN, checks it against its configured verification logic, and rejects the operation when the PIN is invalid. All operations run on simulated data and do not perform real financial transactions.

4.3 Spending Visualization

The Spending Breakdown feature presents monthly expenditure as a pie chart (Fig. 3), allowing users to compare spending categories as proportions of the monthly total instead of reading figures only as text. The chart reflects the categories present in the prototype data.

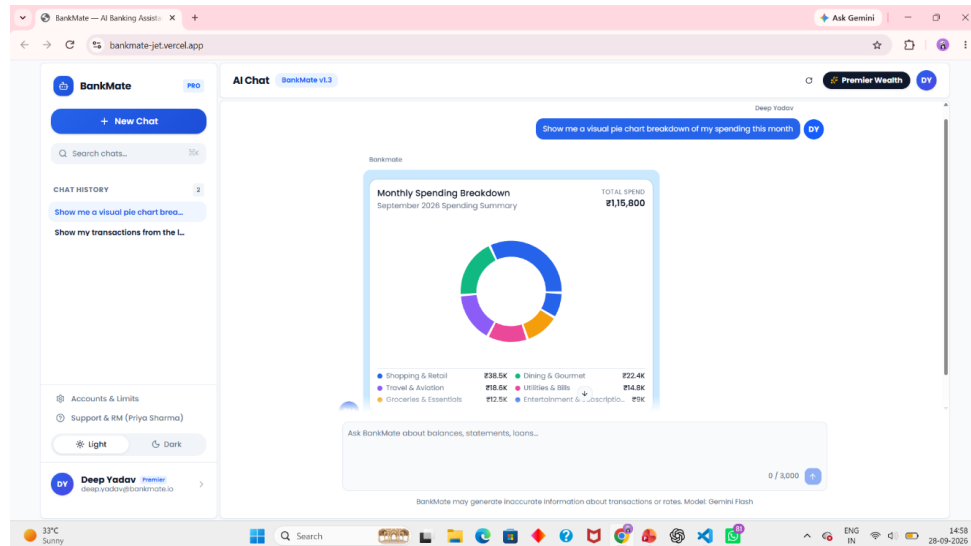


Fig. 3. Monthly spending breakdown using a pie chart

4.4 Deployment

BankMate is deployed as a web application on Vercel and can be accessed through a browser at <https://bankmate-jet.vercel.app/>. The functionality available in the deployed version depends on the deployment configuration and the services the application uses.

5. TESTING AND RESULTS

5.1 Testing Approach

BankMate was tested manually through the deployed application using a black-box approach. Requests were submitted through the interface and the observed behavior was compared with the expected results. Scenario-based testing covered common banking interactions, and negative testing covered invalid PINs and unsupported requests. Automated testing and comprehensive security testing were not performed.

5.2 Functional Results

Table 3 lists the nine scenarios tested. All of them produced the expected outcome, including rejection of an invalid PIN with the displayed balance unchanged and a polite refusal of an unsupported query.

TABLE 3: FUNCTIONAL AND VALIDATION TEST CASES

Test ID	Test Case	Expected Result	Status
TC-01	Balance inquiry	Available balance information is displayed	Pass
TC-02	Loan inquiry	Relevant loan information is displayed	Pass
TC-03	Loan application	The request is processed and the application result is displayed	Pass
TC-04	Conversational banking interface	The application responds to supported banking queries	Pass
TC-05	Fund transfer request	The transfer workflow starts and PIN verification is requested	Pass
TC-06	Enter an invalid PIN	The transaction is rejected	Pass
TC-07	Check account state after invalid PIN	The account balance remains unchanged	Pass
TC-08	Submit an unsupported query	The application politely rejects the request	Pass
TC-09	Verify operation status	The interface displays the appropriate operation result	Pass

5.3 Conversational Interaction and Loan Services

During testing, supported banking requests submitted through the chat interface received relevant responses, showing that features can be reached through natural language instead of menu navigation. A loan inquiry displayed loan information and eligible offers from the simulated data, and a loan application scenario created the application record and presented its outcome. These results apply to the prototype data and do not represent loan approval by a financial institution.

5.4 Performance and Responsiveness

The project evaluation reported an AI response time of approximately 650 ms to 1.2 s, smooth interface behavior at about 60 FPS, and expected results in the tested banking tool executions. These are reported observations rather than controlled measurements; formal benchmarking under different network conditions, concurrent users, and heavier workloads has not been conducted.

5.5 Discussion

The results indicate that conversational AI can be combined with a web-based banking interface to support selected workflows. Balance and loan requests were reachable through natural language, the fund-transfer workflow included PIN verification as an extra step for a sensitive operation, and an unsupported query was rejected. These observations provide preliminary functional validation only. They do not establish the security of the whole transaction process or correctness against a real banking system.

6. LIMITATIONS

BankMate uses simulated banking data and is not connected to a live bank database, so it does not provide verified real-time account information or execute real transactions. Evaluation was limited to manual testing of selected scenarios. Conversational responses may be incomplete, inaccurate, or inconsistent, so AI-generated information needs validation before use in financial decisions. The prototype covers only selected features, and PIN verification and validation controls have not undergone comprehensive security or penetration testing. The usefulness of the spending visualization for real financial planning depends on the availability and accuracy of actual transaction records.

7. CONCLUSION AND FUTURE WORK

BankMate demonstrates how conversational AI can be incorporated into a web-based banking interface to make banking-related information more accessible. By integrating Google Gemini with the Vercel AI SDK and modern web technologies, the prototype brings account information, balance inquiries, loan services, fund-transfer workflows, and monthly spending visualization into one interface. Manual testing showed that the tested workflows behaved as expected: fund transfers required PIN verification, invalid PINs were rejected without changing the displayed balance, and unsupported queries were declined. BankMate remains a prototype that does not connect to a live banking system.

Future work includes integration with authorized banking APIs; multi-factor authentication, secure session management, and stronger transaction authorization; advanced spending analysis such as weekly and yearly reports and budget tracking; improved conversational context handling; personalized financial assistance; expanded operations such as bill payments and fixed-deposit management; multilingual and accessibility support; and automated testing, security assessment, and performance evaluation under varied workloads before any wider use.

REFERENCES

- [1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2023.
- [2] S. Sharma and P. Singh, "Natural language processing-based chatbots for banking applications," in *Proc. Int. Conf. Computing and Communication Technologies (ICCCCT)*, 2021, pp. 145–150.
- [3] McKinsey & Company, *The State of AI in Financial Services*. New York, NY, USA: McKinsey Global Institute, 2023.
- [4] Deloitte, *AI and Digital Banking Trends Report*. New York, NY, USA: Deloitte Insights, 2023.
- [5] Microsoft, *AI in Banking: Transforming Financial Services*. Redmond, WA, USA: Microsoft Corporation, 2023.
- [6] Accenture, *Banking on AI: Intelligent Automation in Financial Services*. Dublin, Ireland: Accenture Research, 2022.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [8] T. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
- [9] M. Patel and V. Shah, "AI-driven customer service automation in banking," *International Journal of Computer Applications*, vol. 184, no. 12, pp. 15–21, 2022.
- [10] IBM, *IBM Watson Assistant Documentation*. Armonk, NY, USA: IBM Corp., 2023.
- [11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2021.
- [12] Google Cloud, *AI Solutions for Financial Services*. Mountain View, CA, USA: Google LLC, 2023.