

Bank Bot AI Chatbot For Banking FAQs

Transforming Banking Through AI-Powered Conversations

Presented By Team A:

- Yuvaraj Gowda
- Sreekakula Geethika
- Gandla Rupasree
- BhavaniChittuluri

Project Overview: Automating the Banking Experience

The Challenge

Traditional banking customer service relies heavily on manual systems, leading to lengthy wait times and high operational costs for routine queries.

The Solution: BankBot

An intelligent AI chatbot leverages Natural Language Processing (NLP) and Machine Learning (ML) to provide instantaneous, accurate responses.

Key Benefits

Ensures 24/7 availability, significantly reduces staff workload, and leads to measurably improved customer satisfaction scores.

Milestone 1: Intent & Entity Recognition

The foundation of BankBot's intelligence relies on accurately understanding customer requests.

1

Define Key Intents

Defined and cataloged core user goals, such as “Check Balance,” “Apply for Loan,” and “Report Lost Card,” providing diverse training examples.

2

NLU Model Development

Built and rigorously tested the NLU model using the spaCy library for rapid and accurate language processing and analysis.

3

Entity Extraction & Slot Filling

Implemented logic to extract critical details (entities) like account numbers or loan amounts and use them to fill necessary data slots for task completion.

❏ Outcome: The model successfully identifies and processes the top 10 banking intents and associated entities with high precision.

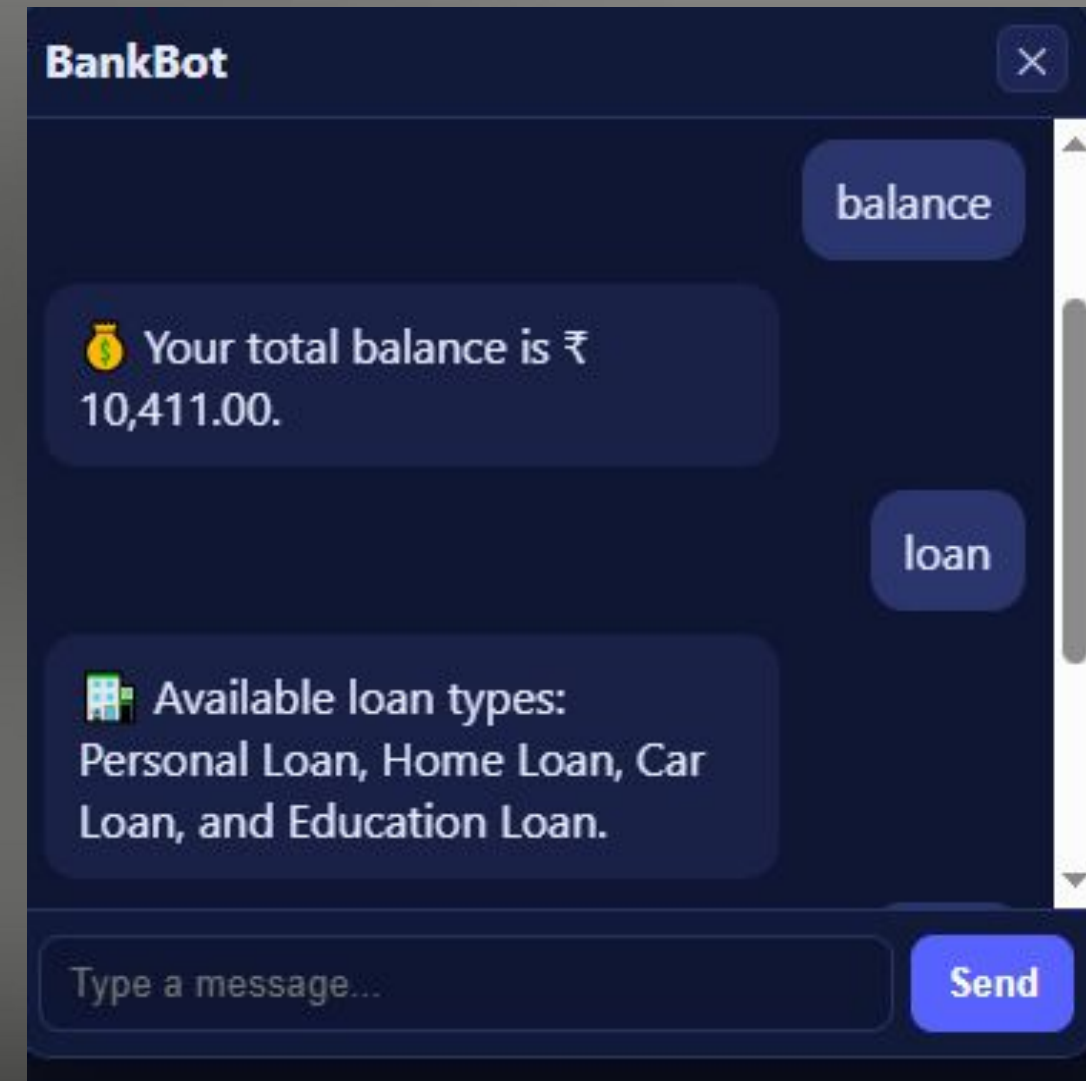
Milestone 2: Dialogue Flow and Context Handling

Ensuring Coherent Multi-Turn Conversations

A superior chatbot must maintain context across multiple user exchanges to feel natural and effective.

- **Conversation Flow:** Structured the dialogue using state-of-the-art ML-based stories and rules, ensuring a logical progression.
- **Context Tracking:** Implemented robust mechanisms for tracking context to support sophisticated multi-turn interactions and follow-up questions.
- **Graceful Fallback:** Developed specialized handlers to gracefully manage ambiguous, irrelevant, or unknown user intents, preventing frustrating dead ends.

Outcome: Successfully achieved a smooth, consistent, and context-aware dialogue flow that mimics human interaction quality.



Milestone 3: UI Integration and User Experience

Web Interface Development

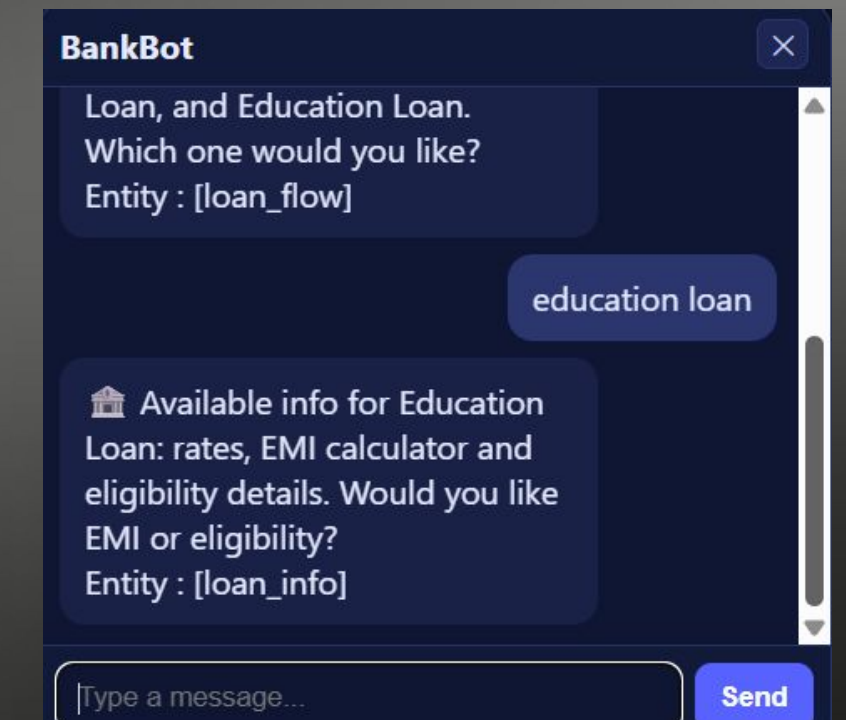
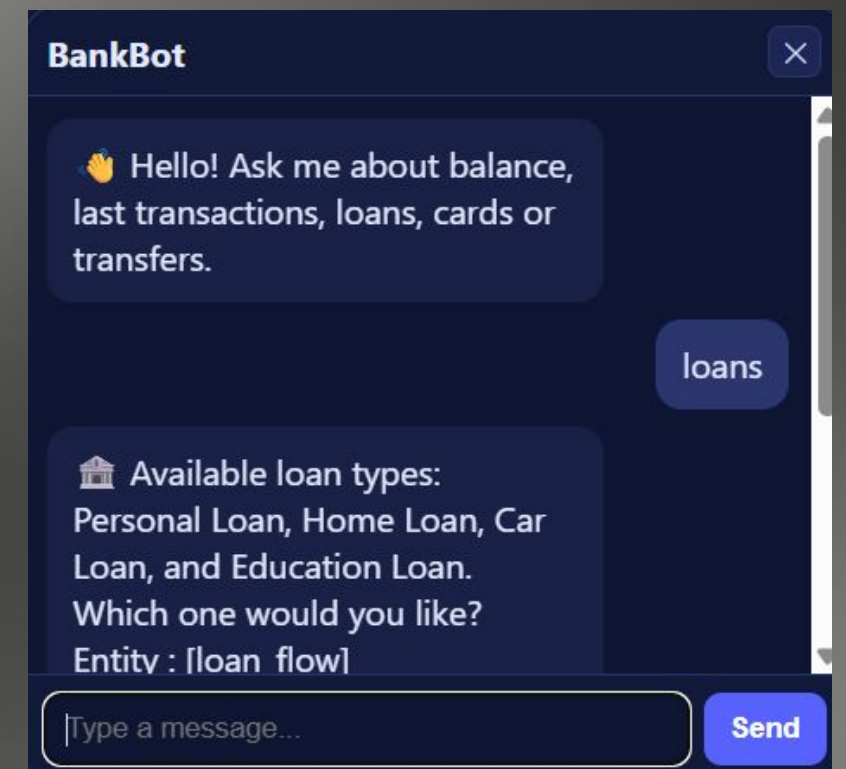
Utilized Streamlit for rapid development of a dynamic, interactive, and responsive web interface for the chatbot.

Real-Time Backend Connection

Established a stable, real-time connection between the Streamlit UI and the Flask-based chatbot backend via a secure REST API.

Error Handling & Design

Focused on a user-friendly layout with robust error handling and mechanisms for graceful restarts following any potential system failures.



Milestone 4: Admin Panel & Knowledge Base Management

The Admin Panel provides necessary controls for maintenance and continuous learning.

1

Dashboard for Maintenance

Developed a secure admin dashboard to manage intents, FAQs, and review raw user query logs.

2

Analytics & Reporting

Integrated features for CSV export, providing insights into intent distribution, top queries, and system performance metrics.

3

Model Retraining

Enabled a critical feature allowing administrators to trigger and execute model retraining directly from the UI using new data.

Training Data

Edit the CSV-like training content below
(intent,text,response or your chosen format). Press Save
then Retrain.

```
text,intent,response
hi,greet,👋 Hello! How can I help you today?
hello,greet,👋 Hi there! Ask me about
balance, transactions, loans, cards or
transfers.
hey,greet,👋 Hey! What can I do for you?
bye,goodbye,👋 Goodbye! Have a great day.
goodbye,goodbye,👋 See you next time.

what is my balance,balance_check,💰 Your
total balance appears on the left. I can also
read it out if you want.
check balance,balance_check,💰 Sure –
showing your account balance above.
```

 **Save Changes**

 **Retrain Now**

Export / Utilities

Download logs or manage data.

 **Download Logs (CSV)**

Recent window size: **40**

Core Technology Stack

A selection of robust and scalable open-source technologies powers the BankBot system.



Python & Flask

Used for all backend logic, handling REST API calls, and orchestrating component interactions.



spaCy

The primary NLP engine, providing highly efficient intent recognition and entity extraction capabilities.



Streamlit

Empowers the responsive and user-friendly frontend web interface for seamless customer interaction.



SQLite / MySQL

Selected for reliable and persistent data management, storing conversation logs and the knowledge base.



Render / Vercel

Chosen deployment infrastructure, enabling fast, scalable, and reliable cloud-based hosting.

Outcomes

Current Achievements

→ Automation of Core Queries

Successfully handles key banking queries in real-time, reducing the need for human intervention.

→ Scalability & Real-Time Performance

Built on a scalable architecture that ensures fast response times even during peak load periods.

→ Retrtrainable & Adaptable

Easy integration of new data and quick retraining cycles to adapt to evolving customer needs.

Thank You

SREEKAKULA GEETHIKA , GANDLA RUPASREE , YUVARAJ GOWDA ,
BHAVANI CHITTULURI