BANKBOT CHATBOT AI

BY

VIBHEESHAN N K

PROBLEM STATEMENT

- Customers ask the same banking questions again and again.
- Examples: balance enquiry, loan details, transaction status, branch details.
- Manual customer service takes more time to respond.
- Human support increases operational cost for banks.
- Customer support is not available 24×7.
- Users want quick and simple answers.
- Manual systems are not scalable for many users.
- There is no natural conversation in existing systems.
- An automated chatbot is required to solve these issues.
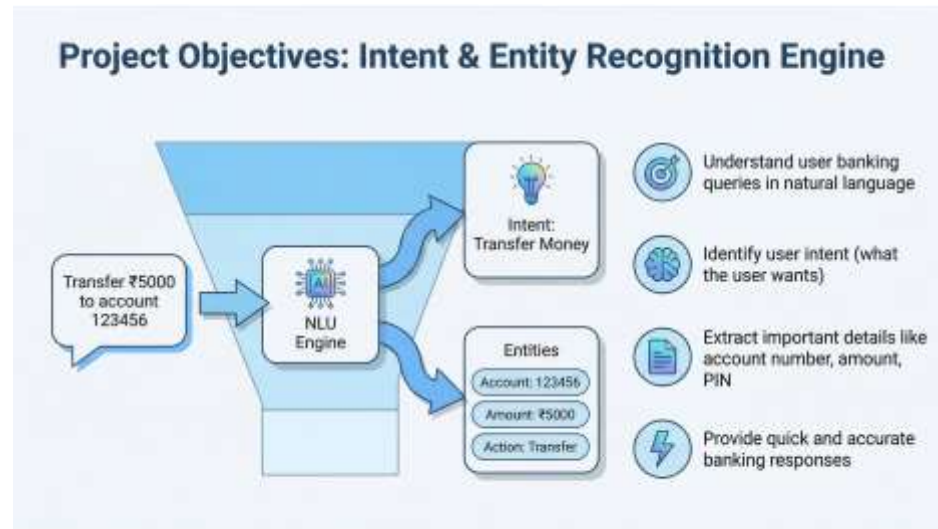
Solution Statements

- Develop an **AI-powered banking chatbot** to automate customer support.
- Use **Natural Language Processing (NLP)** to understand user queries in simple, conversational language.
- Implement **intent classification** to identify user requests such as balance enquiry, money transfer, card blocking, and ATM search.
- Apply **entity extraction** to capture important details like account number, amount, and transaction PIN.
- Integrate a **rule-based dialogue manager** to handle multi-step conversations accurately.
- Store and manage user data securely using an **SQLite database**.

Milestone 1 : Intent & Entity Recognition Engine
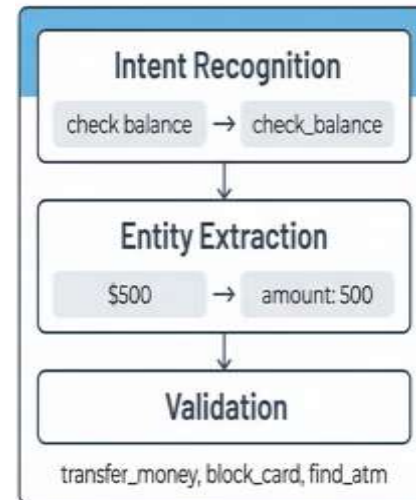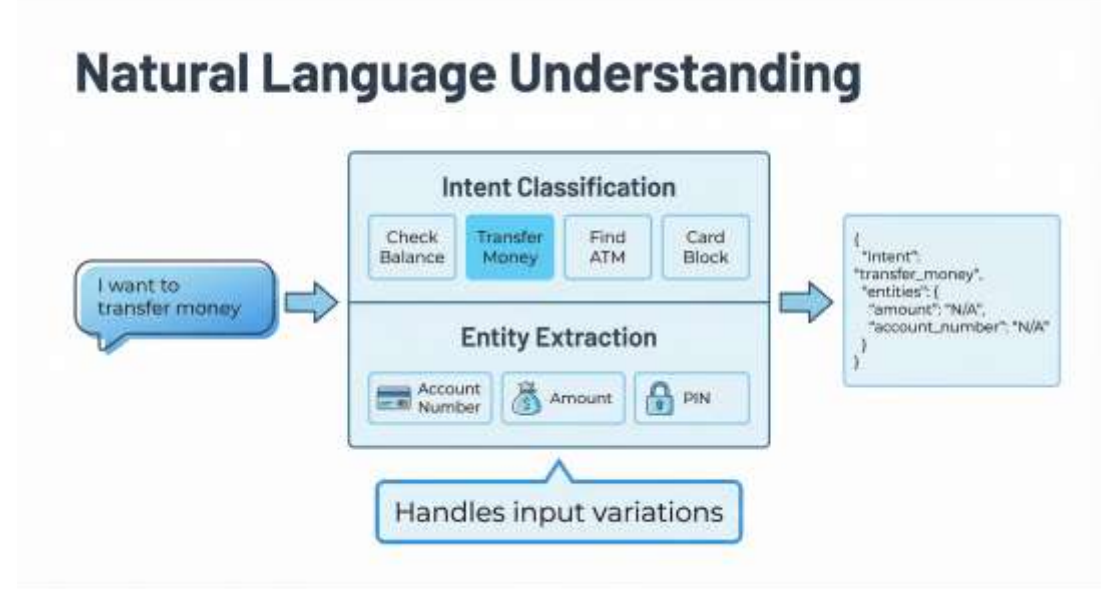
Objective of Milestone 1

- Understand user banking queries

- Identify what the user wants (Intent)

- Extract key information (Entities)

- Build the foundation for chatbot intelligence

- Designed system architecture.

Project Objectives: Intent & Entity Recognition Engine

Milestone 1 : Intent & Entity Recognition Engine

ARCHITECTURE

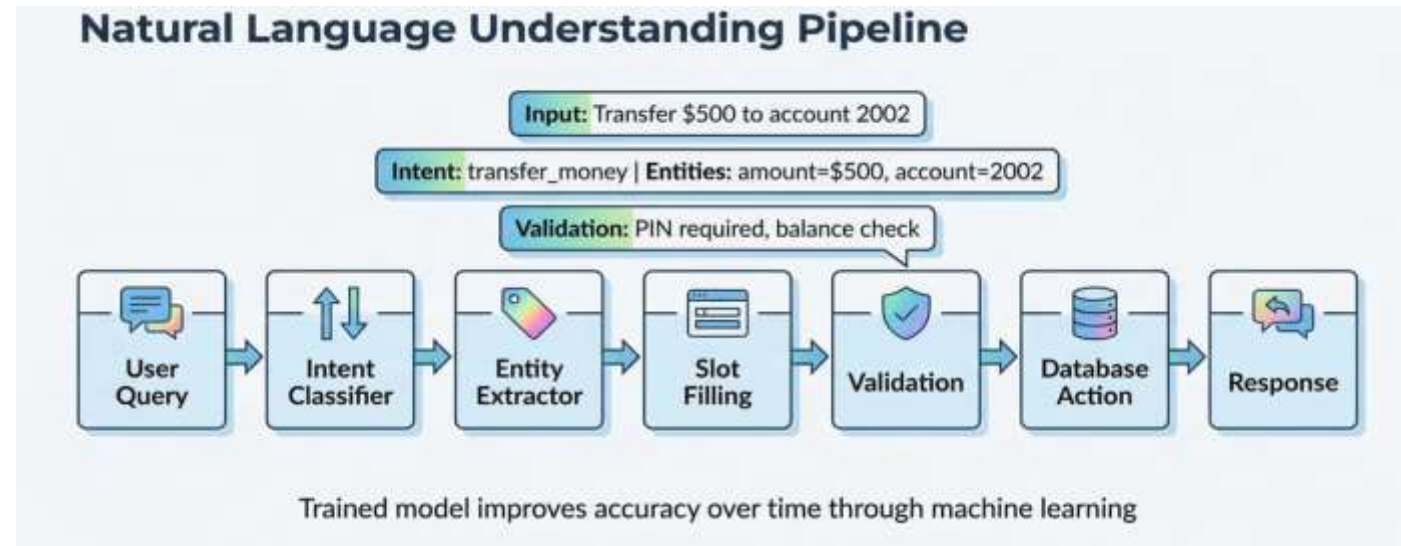- **Intent Recognition :** identifies the purpose of a user message
- **Examples**
- "Check my balance" → check_balance
- **Intents Implemented :** Banking Intents Covered
- EG: transfer_money , block_card
- **Entity Extraction :** finds important values from text.
- EG : Account number →1001
- Entities Used (Eg: Amount($,Rs ,INR) .
- Built NLU engine and train the model

Natural Language Understanding

Milestone 1: NLU Engine

NLU Architecture

Components: User Query, Intent classifier Entity extractor, Validation , NLU Visualizer , Model training – increase accuracy.
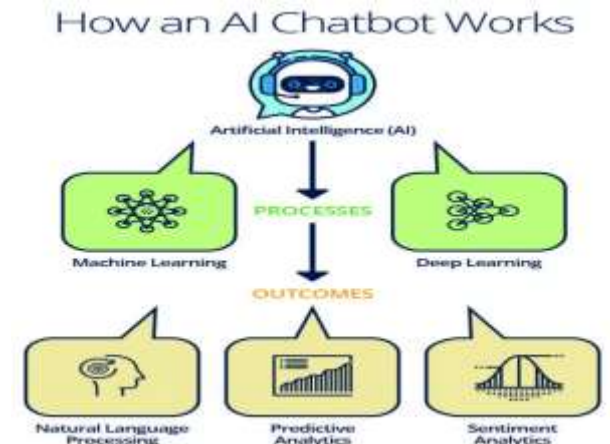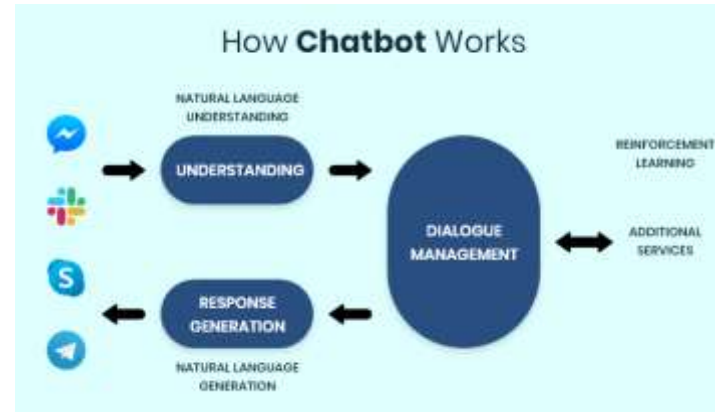
NLU Architecture Workflow

User Query → Intent Detection → Entity Extraction → Validation → Database Action → Response

Natural Language Understanding Pipeline

Input: Transfer $500 to account 2002

Intent: transfer_money | Entities: amount=$500, account=2002

Validation: PIN required, balance check

User Query → Intent Classifier → Entity Extractor → Slot Filling → Validation → Database Action → Response

Trained model improves accuracy over time through machine learning

'NLU Engine: Natural Language Understanding'

Intent Classification
- Check Balance
- Transfer Money
- Find ATM
- Card Block

Entity Extraction
- Account Number
- Amount
- Transaction PIN

Handles variations in user input (typing mistakes, short queries)

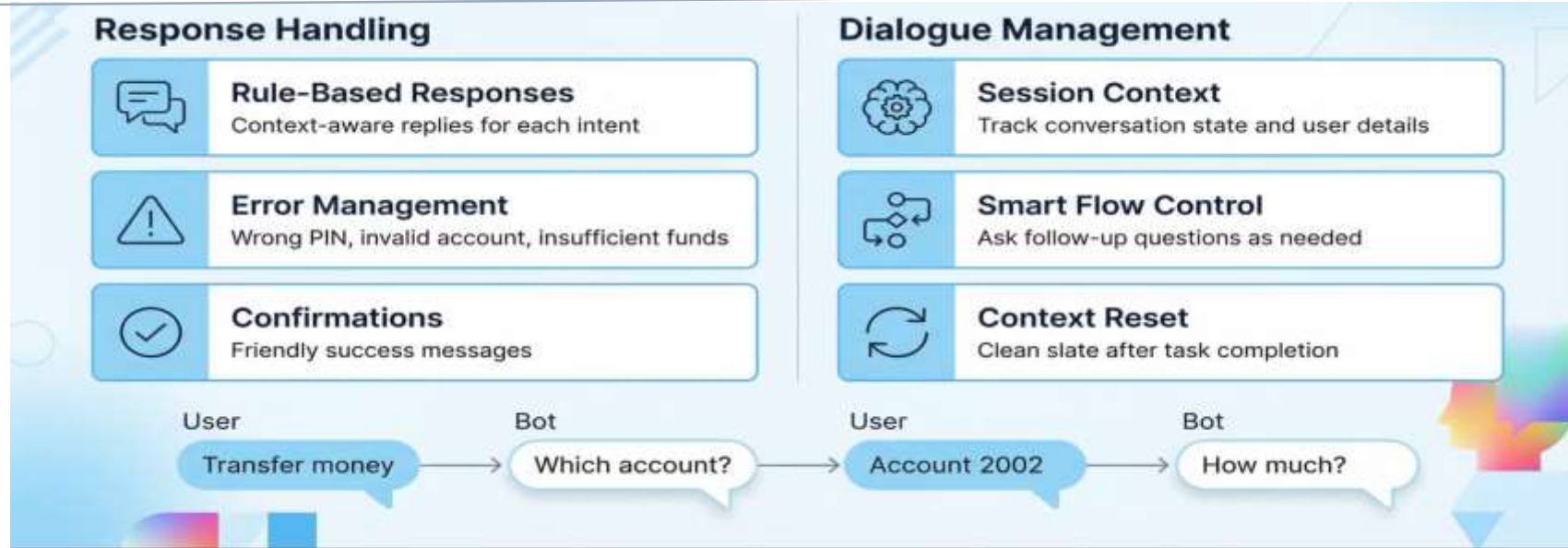# Milestone 2 : Response Handling & Dialogue Flow

- **Milestone 2 Objective**
- Handle multi-step conversations
- Maintain session context
- Ask follow-up questions
- Generate correct responses
- **Goal:** Make chatbot conversational and intelligent.
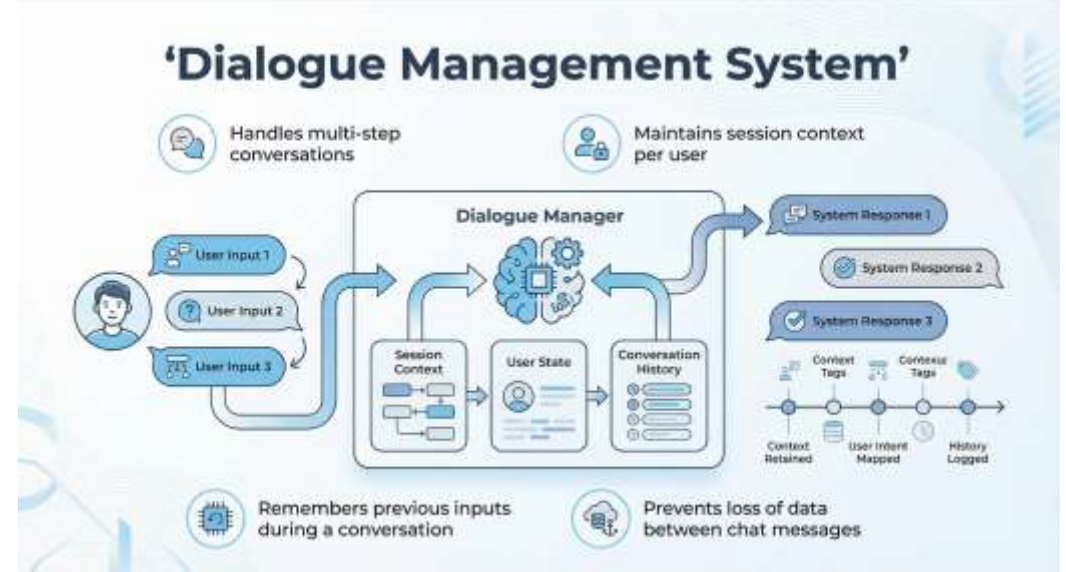
Milestone 2 : Response Handling & Dialogue Flow

Response Handling

- Rule-based responses
- Context-aware replies
- Error messages for:
  - Wrong PIN
  - Invalid account
  - Insufficient balance
- Friendly confirmation messages

- **Dialogue Management :**

- Track conversation state
- Store user-provided details
- Decide next question or response
- Reset context after completion

Response Handling

Rule-Based Responses
Context-aware replies for each intent

Error Management
Wrong PIN, invalid account, insufficient funds

Confirmations
Friendly success messages

Dialogue Management

Session Context
Track conversation state and user details

Smart Flow Control
Ask follow-up questions as needed

Context Reset
Clean slate after task completion

User — Transfer money → Bot — Which account? → User — Account 2002 → Bot — How much?

'Dialogue Management System'

Handles multi-step conversations

Maintains session context per user

Dialogue Manager

User Input 1
User Input 2
User Input 3

Session Context | User State | Conversation History

System Response 1
System Response 2
System Response 3

Context Tags | Context Tags

Context Retained | User Intent Mapped | History Logged

Remembers previous inputs during a conversation
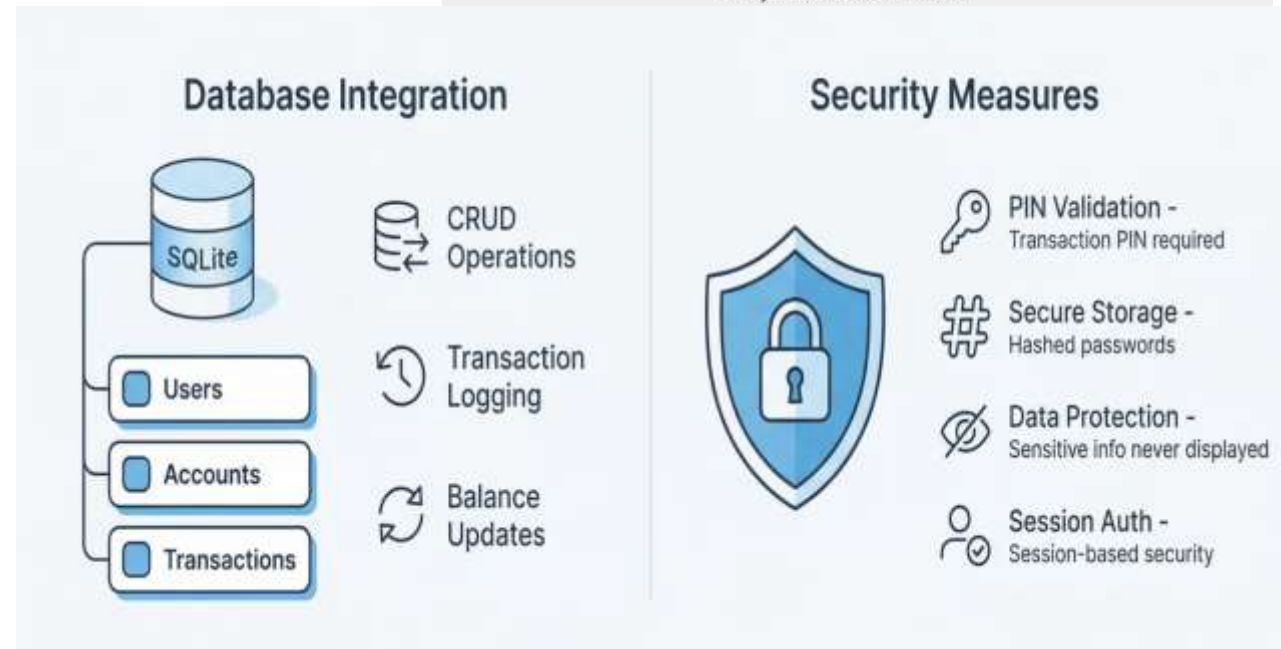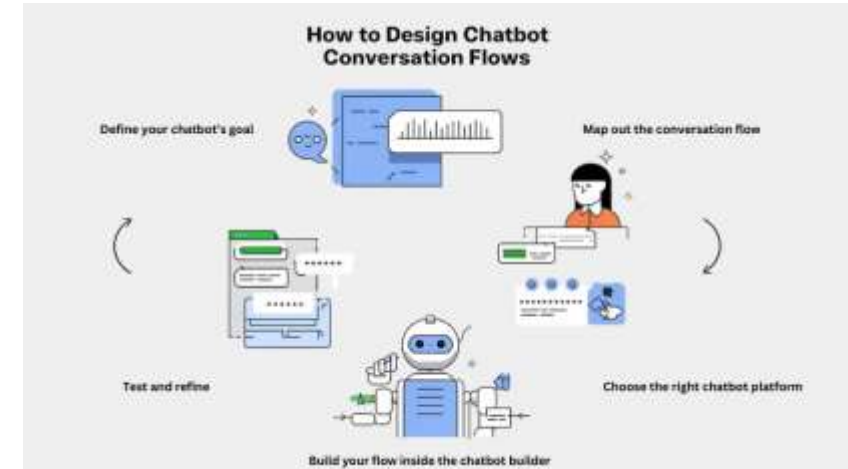
Prevents loss of data between chat messages

Milestone 2:Response Handling & Dialogue Flow

Database Integration

- SQLite database used

- Tables: users, accounts, transactions

- CRUD operations for balance and transfer

- Transaction history logging

Security Handling

- Password / Transaction PIN validation

- PIN stored securely (hashed)

- Sensitive data not displayed

- Session-based authentication
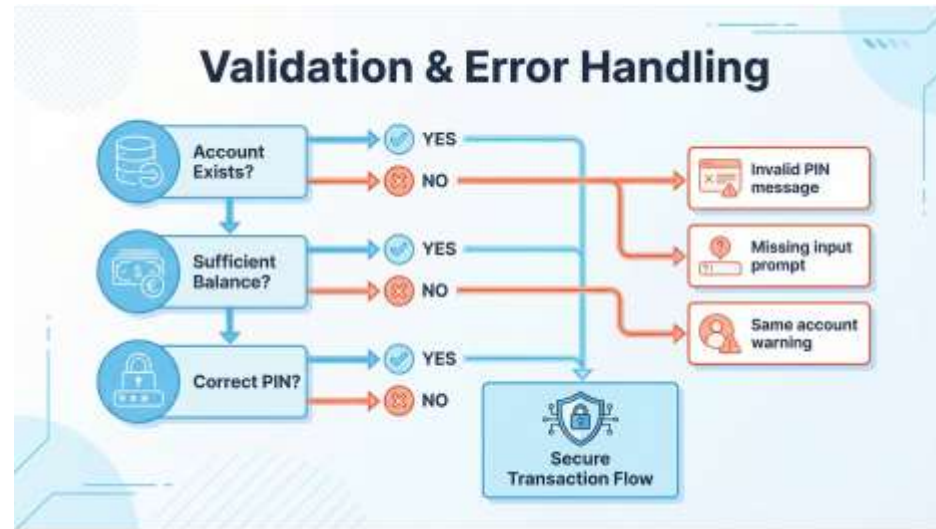
Milestone 2 : Response Handling & Dialogue Flow

Validation & Error Handling

Validates:
- Account existence
- Sufficient balance
- Correct transaction PIN

Handles errors:

- Invalid PIN
- Missing inputs
- Same sender and receiver account
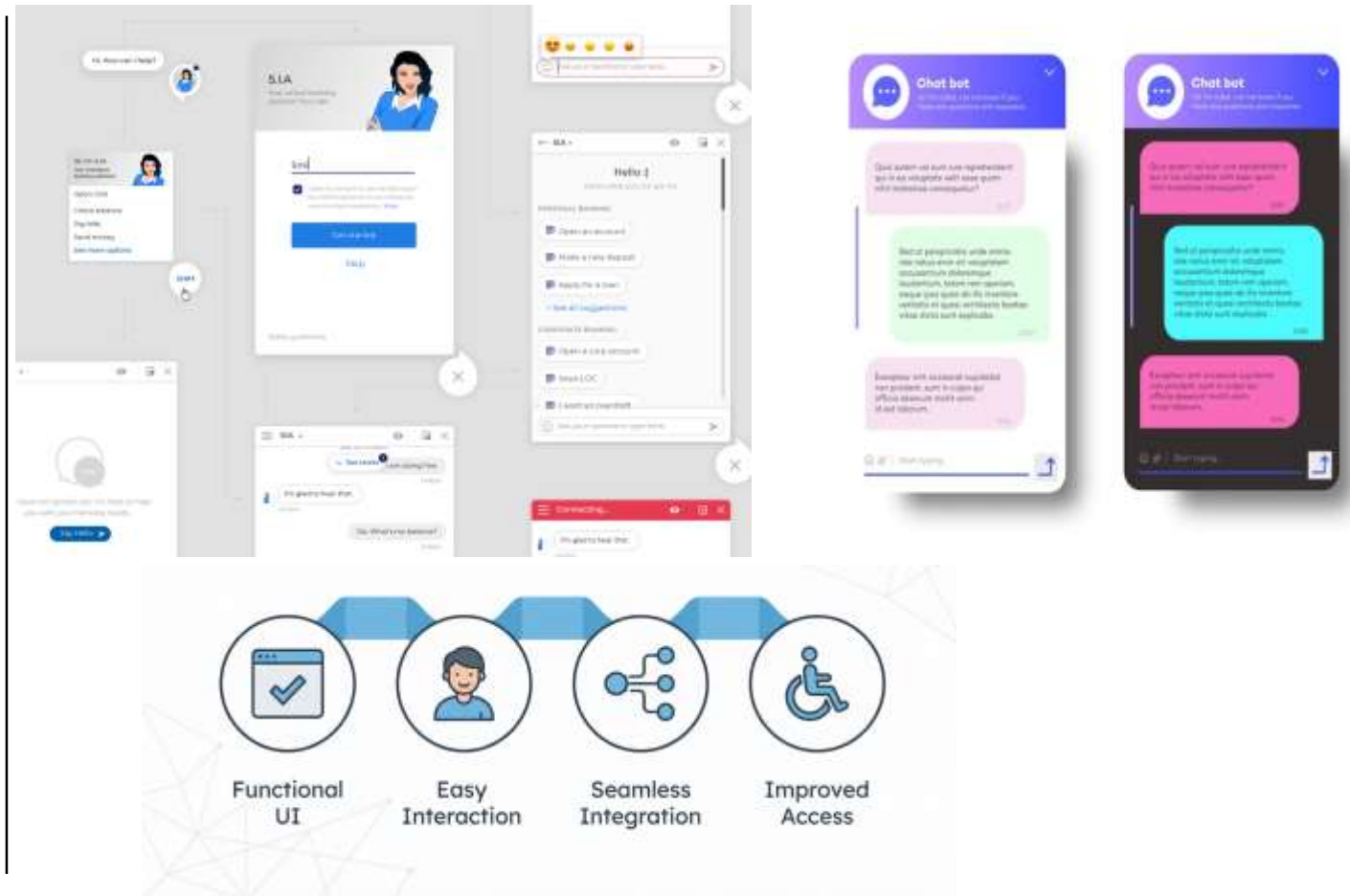- Ensures secure transaction flow

System Architecture
➢Frontend: Streamlit
➢Backend: Python
➢NLU Engine
➢Dialogue Manager
➢SQLite Database

Validation & Error Handling

Account Exists? — YES / NO
Sufficient Balance? — YES / NO
Correct PIN? — YES / NO

Invalid PIN message
Missing input prompt
Same account warning

Secure Transaction Flow

Slot Filling Strategy

1. 'Transfer from my checking.' — From Account — Checking Account — Which account would you like to transfer to?
2. 'To my savings.' — To Account — Savings Account
3. $500. — Amount — $500 — How much would you like to transfer?
4. PIN is 1234. — Transaction PIN — Please provide your transaction PIN.

Step-by-step completion

Milestone 3: UI Integration &chat interface

OVERVIEW

• Streamlit-based user interface

•Real-time chatbot interaction

•Multi-page banking dashboard

•Improved user experience

Milestone 3: UI Integration &chat interface

Login & Account Creation

Login using:

- Account number.
- Login PIN.
- Create Account option inside login page.

New users can:

- Create account.
- Set login PIN.
- Set transaction PIN.
- Login immediately after creation.

Milestone 3: UI Integration &chat interface

Purpose of Database
- To store and manage **user, account, and transaction data**
- To support **real-time chatbot responses** with accurate information
- To ensure **secure and reliable data access** for banking operations
- Database Component **User Table, Transactions Table**
- **Accounts Table, Transcation pin ,login credential.**

Technologies Used
- **SQLite** (lightweight and efficient) ,**Python (CRUD operations)** -database handling
- **Secure hashing** - PIN verification
- Integrated with backend API and chatbot logic

Security Measures in Database
- Password and transaction PIN stored in **hashed format**
- No plain-text sensitive data stored
- Validation checks before every transaction
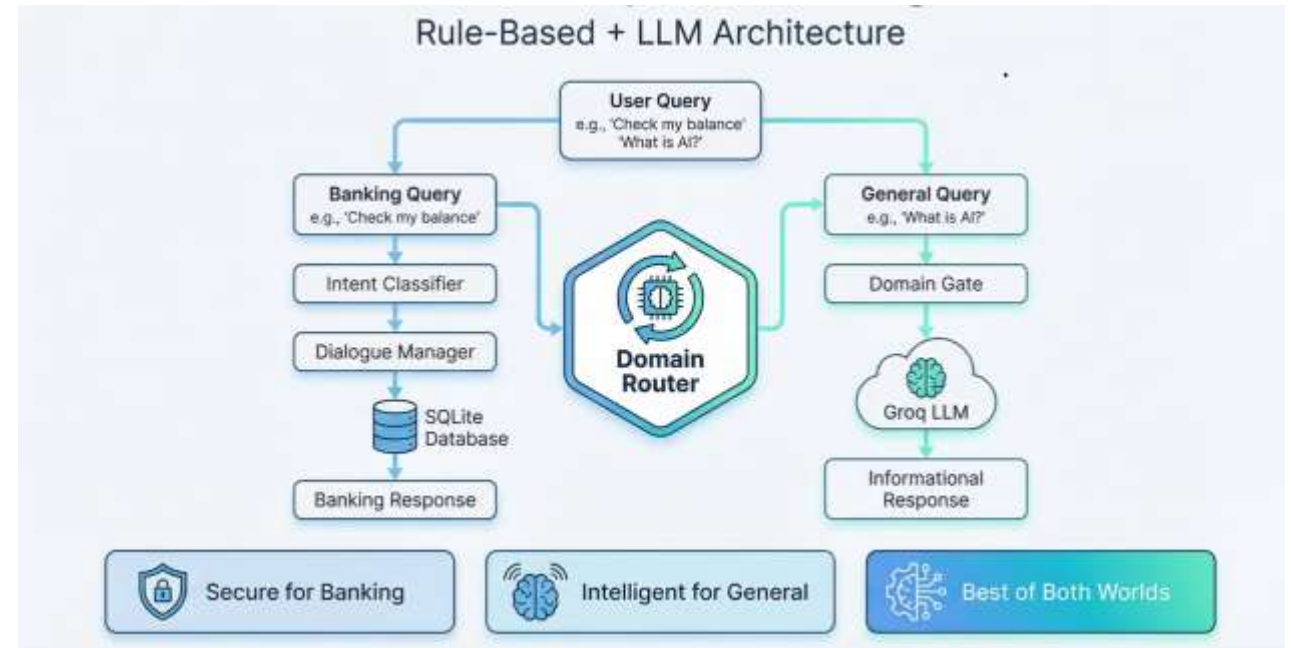- Prevents same-account money transfer

Database Design

Database Integration — Security Measures

Milestone 3 : UI Integration &chat interface

Security Enhancements

❏ Advanced PIN Security

❏ Dual-Layer
Authentication

❏ Zero Plain Text Storage

❏ LLM Isolation

Hybrid AI Architecture
- Frontend Layer
- Streamlit-based responsive web interface
- Backend Processing
- Python-powered NLU Engine and Dialogue
  Manager
- Data Layer
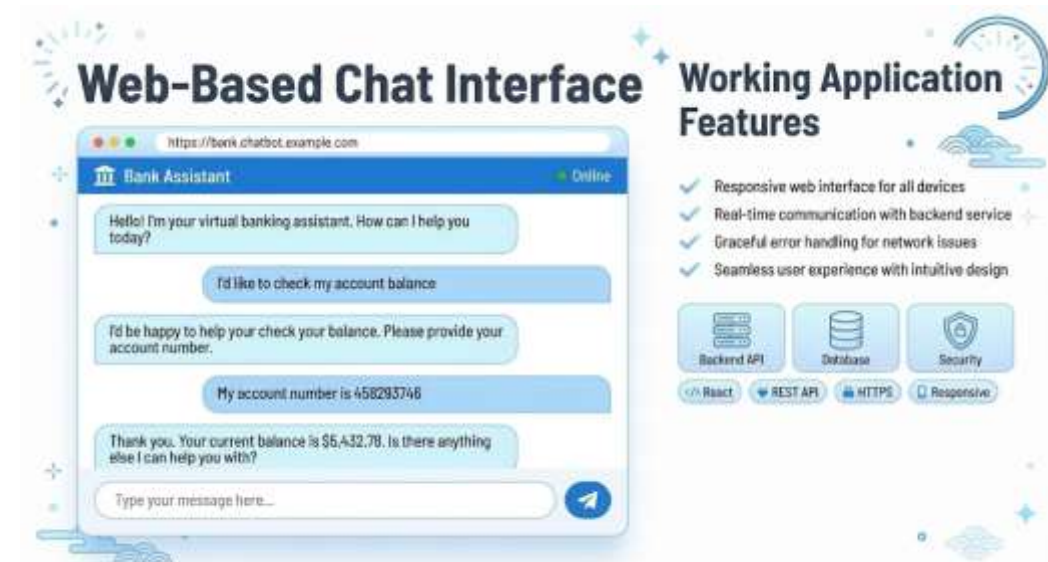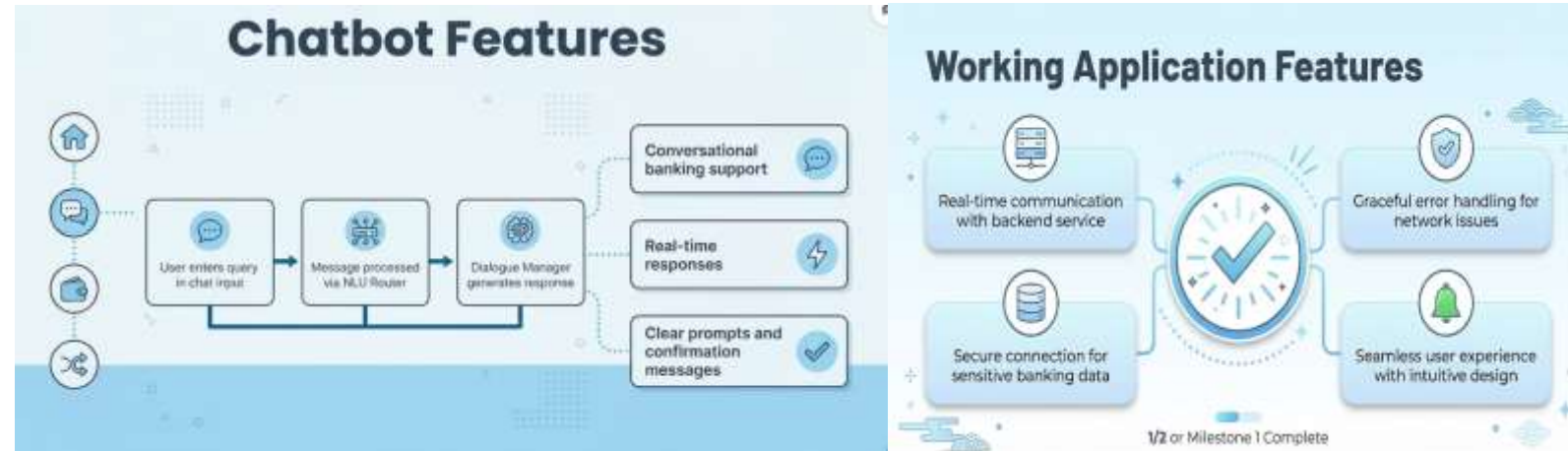- SQLite database with secure transaction
  management

Milestone 3: UI Integration &chat interface

Features Implemented

- Login page with authentication

- Chatbot interface

- Sidebar navigation

- Account information display

- Transaction history view

Chatbot UI Flow

- User enters query in chat input
- Message processed via NLU Router
- Dialogue Manager generates response
- Response displayed instantly

Milestone 4:Admin Panel & Knowledge Base

Objective of Milestone 4
- To design and develop an **Admin Panel** for effective management of the BankBot AI system.
- To enable administrators to **monitor chatbot usage and performance** in real time.
- To provide **analytics and logs** for tracking user queries, intents, and success rates.
- To allow admins to **manage and update the knowledge base** dynamically.
- To support **editing training data** for intents and entities without code changes.
- To identify errors, low-confidence responses, and improve chatbot accuracy.

Milestone 4:Admin Panel & Knowledge Base

Admin Panel Features

- It provides a centralized dashboard for administrators
- View overall chatbot performance
- Monitor user queries and intents
- Manage training data and FAQs
- Export logs for analysis
- This ensures **continuous improvement** of the chatbot system.

Admin Panel Features

Knowledge Base Management
It enables administrators to:
- Add, edit, or remove **FAQs**
- Update **question–answer pairs** dynamically
- Improve chatbot responses without changing core code
- Handle new banking queries easily
- This makes the chatbot **scalable and adaptable**

Knowledge Base Management
Dynamic & scalable

Milestone 4:Admin Panel & Knowledge Base

Query Analytics & Monitoring
- ✓ The admin dashboard includes analytics such as:
- ✓ Total number of user queries
- ✓ Intent detection accuracy
- ✓ Success and failure rates
- ✓ Frequently asked questions
- ✓ Confidence scores for predictions
- ✓ These insights help identify **errors, weak intents, and user behavior patterns**.

- **Training Data Management**
- Admins can:
- Review incorrect or low-confidence predictions
- Update intent labels and entity examples
- Retrain the model using improved datasets
- Maintain high chatbot accuracy over time
- This supports **continuous learning**.

Milestone 4:Admin Panel & Knowledge Base

Logs & CSV Export
- The system allows:
- Exporting chat logs as **CSV files**
- Downloading analytics reports
- Using logs for debugging, audits, or research
- Sharing data with stakeholders or mentors

Security & Admin Access
- ❑ Admin panel is accessible only to authorized users
- ❑ Sensitive data is protected
- ❑ User privacy is maintained
- ❑ Role-based access control can be implemented

Logs & CSV Export

Debug   Audit   Research

Data for stakeholders

Security & Access

Role-based protection

Authorized Only

Access Control

Privacy Protected

Data Secure

Future Scope & Conclusion

Future Scope

- Add support for **multiple languages**
- Enable **voice-based chat** for users
- Connect with **real banking APIs** for live services
- Improve accuracy using **advanced AI models**
- Deploy chatbot on **mobile apps and WhatsApp**
- Add **fraud alerts and security notifications**

Conclusion

This project successfully developed an **AI-based banking chatbot** that answers common customer queries.

- It understands user questions using **NLU**
- Provides quick and accurate responses
- Works **24/7** without human support
- Reduces workload of bank staff
- Can be easily upgraded in the future

The system is **useful, scalable, and suitable for real-world banking applications**.