



TITLE: DEVELOPMENT OF A MULTI-PURPOSE CHATBOT : INTEGRATING CONVERSATIONAL AI WITH TASK-SPECIFIC FUNCTIONALITIES

ORGANIZED BY:

CODING CLUB

IIT GUWAHATI

PRESENTED BY:

IDEATIONX



TRUMIO PROBLEM STATEMENT

CHALLENGE OVERVIEW:

The objective is to develop a sophisticated Multi-Purpose ChatBot that combines the capabilities of general conversational AI with task-specific functionalities. The ChatBot should be designed to perform general discussions similar to GPT models while also executing a range of command tasks that provide practical, real-world value. This dual capability aims to create a dynamic tool that can interact naturally with users and also handle specific, predefined tasks efficiently.

NOVELTY:

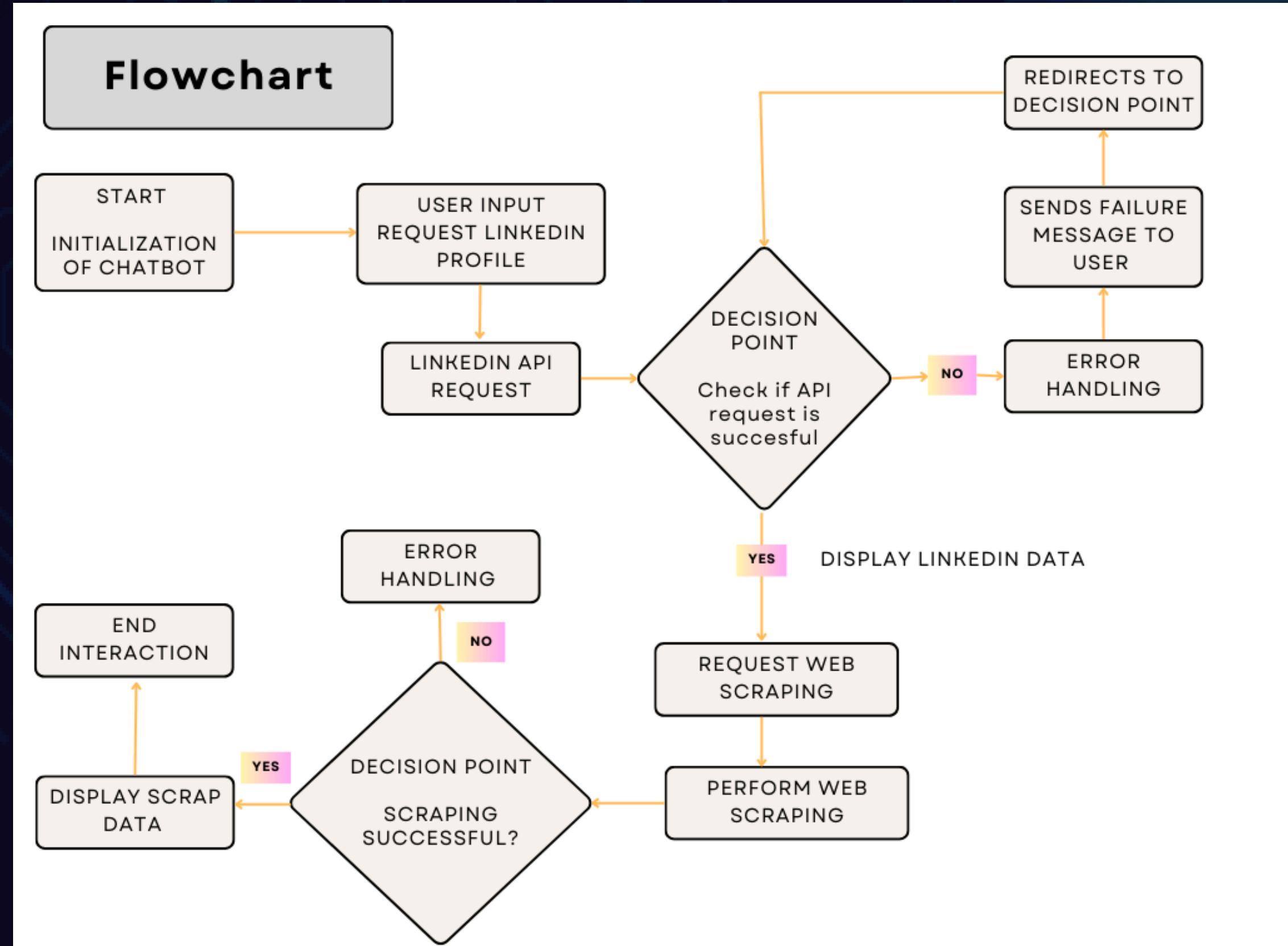
Our ‘INDIEI’ chatbot combines conversational AI with automated tasks like LinkedIn data retrieval and web scraping, offering real-time, context-aware interactions. It integrates multiple functions into one platform, improving efficiency and productivity.

OBJECTIVE:

- Engages users in natural, context-aware conversations.
- Performs specific command tasks that deliver real-world utility, such as web scraping and LinkedIn interactions.
- Ensures seamless API integrations and real-time responsiveness.



FLOWCHART:



FUNCTIONAL SCOPE OF THE CHATBOT

CONVERSATIONAL CAPABILITIES:

Utilizing models like GPT to handle a wide array of user interactions.

COMMAND DRIVEN TASKS:

- Focused on task automation and API integration.
- Retrieving LinkedIn profiles.
- Posting updates on LinkedIn.
- Web scraping for specific content.

HIGH-LEVEL ARCHITECTURE DIAGRAM ILLUSTRATING:

- Interaction between the user, frontend (ReactJS) and backend (FastAPI).
- Command processing flow.
- API integration for task-specific functions.
- Key Components:

User interface, API gateway, WebSockets handler, NLP engine (LangChain/OpenAI), task manager, external APIs (LinkedIn, web scraping services).



TECHNOLOGICAL INTEGRATION

- **FASTAPI:** For developing RESTful API endpoints.
- **PYTHON:** Leveraging its rich ecosystem for AI, automation, and web scraping.
- **REACTJS:** Building an interactive and responsive user interface.
- **WEBSOCKETS:** Facilitating real-time communication between frontend and backend.
- **LANGCHAIN:** Managing task flow and chaining NLP tasks.
- **OPENAI:** Driving conversational capabilities.

DATA FLOW AND RESPONSE MECHANISM

- **INPUT FLOW:** Data entered by the user (commands, queries).
- **PROCESSING LAYER:** API integration and task execution.
- **RESPONSE FLOW:** Real-time output delivered back to the user.

API INTEGRATION AND SECURITY CONSIDERATIONS

API SECURITY:

- Use of OAuth 2.0 for LinkedIn API.
- Token management and refresh strategies.
- Data privacy concerns and encryption.

WEB SCRAPING ETHICAL CONSIDERATIONS:

- Respecting robots.txt.
- Rate-limiting and avoiding data scraping from restricted content.

ERROR SCENARIOS:

- Invalid API responses (LinkedIn, scraping targets).
- Command misinterpretation in NLP tasks.

MITIGATION:

- Exception handling in Python.
- Fallback responses for unsupported tasks or command failures.

THANK YOU