“Groq Llama3 Python Implementation: Conversation Manager & JSON Schema Extraction”

Demo Project for AI Chat Summarization & Data Extraction

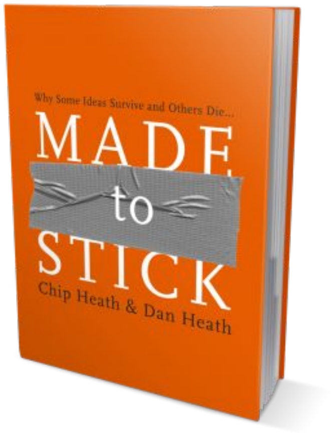
Mamta
Date = 25/09/25

Problem Statement

.Automate chat summarization after defined turns using Llama3.

.Extract structured user information (name, email, phone, age, location) from chat messages.

.Implement both tasks using Groq API in Python.



Task 1: Conversation Manager

.Manage chat history in Python class.

.Store messages with **role** (user/assistant) and **content**.

.Summarize chat automatically after **k turns**.

.Keep track of **turn count** and limit history if needed.

## Flow: User → Assistant → Summarize → Update History

Conversation Manager Class

Explanation:

- history stores all messages
- k = number of turns after which summary occurs
- turn_count counts total chat exchanges

Add message function:

- Appends message to history
  ## Increments turn count
- Summarizes chat if turn_count % k == 0

Summarize function:

- Concatenates messages
- Calls **Llama3 API** to generate summary
- Replaces old history with summarized version

```
def __init__(self, k=3):
    self.history = []
    self.k = k
    self.turn_count = 0
```

Demo Chat Summary

Chat Turns:

1. User: “Hi, I want to learn AI.”
2. Assistant: “Sure, where to start?”
3. User: “Tell me about AI and its popularity.”

## Next 3 turns summarized similarly.

After 3rd turn (summary applied):

“Summary:
User wants to learn AI and assistant explains its popularity and basics.

Task 2: JSON Schema Extraction

Extract structured info from chat messages.

Schema defines required fields: `name, email, phone, location, age`.

Use Llama3 API to convert free text → JSON object.

Flow: User Chat → API → JSON Output

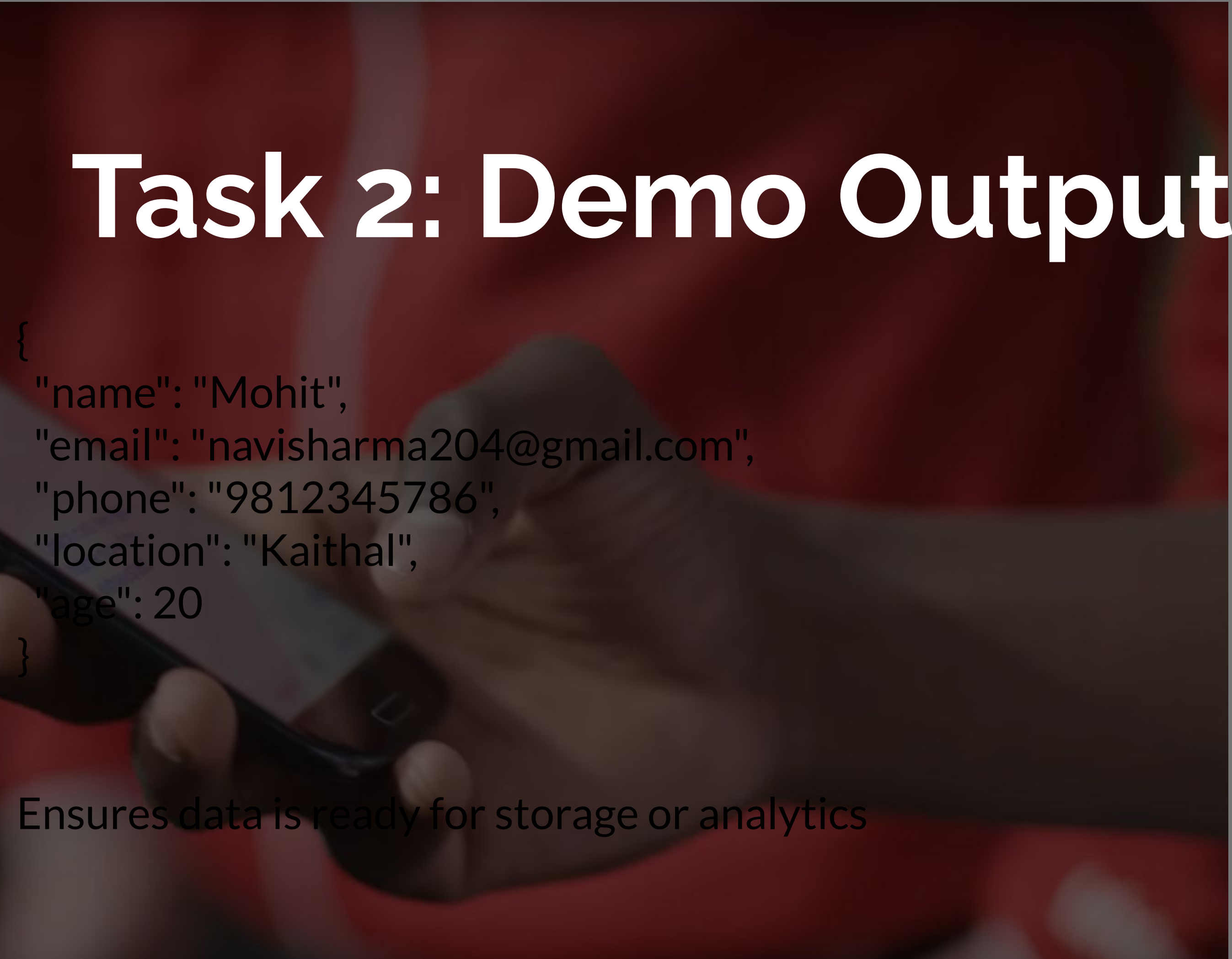
Python Function: extract_info()

Code snippet

```
def extract_info(chat_text):
    response = client.chat.completions.create(
        model="llama-3.1-8b-instant",
        messages=[...],
        tools=[schema],
        tool_choice="auto"
    )
    return response.choices[0].message.tool_calls[0].function.arguments
```

Explanation:

- Sends chat text to Llama3 model
- Model extracts info according to JSON schema
- Returns structured dictionary with user info



Task 2: Demo Output

```
{
 "name": "Mohit",
 "email": "navisharma204@gmail.com",
 "phone": "9812345786",
 "location": "Kaithal",
 "age": 20
}
```

Ensures data is ready for storage or analytics

Similar outputs for other users

Key Takeaways

Chat summarization improves conversation management.

JSON schema extraction standardizes unstructured chat data.

Python + Llama3 API can automate repetitive AI tasks efficiently.

Project demonstrates **real-world AI application** using Groq API.