

ARTIFICIAL INTELLIGENCE &  
MACHINE LEARNING

# PROJECT

## CHAT BOT DEVELOPMENT

By Team ASK

ASHISH TRIVEDI

SRINATH

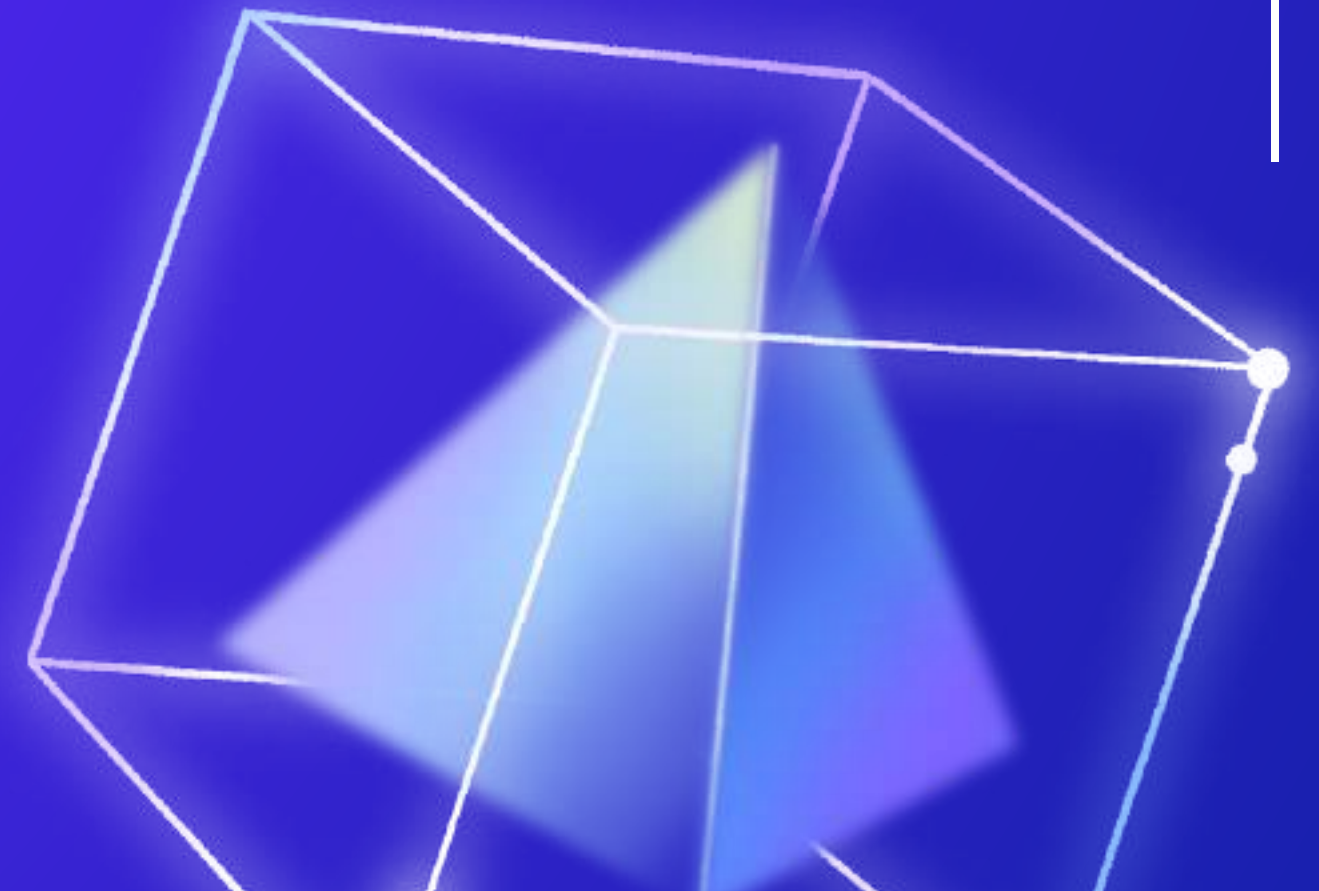
KALKI





# TABLE OF CONTENTS

• Goal/Objective	03
• Model Selection	04
• Code	05
• Output	11
• Resources	12



# GOAL/OBJECTIVE



The goal of this project is to develop a versatile and intelligent chatbot that utilizes Large Language Models (LLMs) for multiple use cases, including text summarization, answering custom questions (e.g., college-related queries), and managing fitness tracking data. This chatbot aims to enhance user interaction by providing quick, accurate responses in various contexts, making it a valuable personal assistant for organizing tasks, retrieving information, and managing fitness goals.

The primary objectives of the project include:

- **Text Summarization:** Leverage LLMs to generate concise summaries of large text inputs, enabling users to quickly grasp key points.
- **Question Answering:** Utilize pre-trained models to answer specific queries related to college information, such as contact details, fees, admissions, and more, streamlining the user's search for relevant data.
- **Fitness Management:** Enable users to log, view, and delete their workout history, set fitness goals, and track calories burned, all through natural language input.
- **Seamless User Experience:** Provide a user-friendly interaction environment with intelligent responses that adapt to user needs, whether they require educational information or personal fitness insights.

By integrating advanced NLP techniques and leveraging LLM capabilities, the chatbot is designed to improve efficiency, reduce task load, and act as an all-in-one digital assistant.





# MODEL SELECTION

For the chatbot development, the model selection focuses on utilizing state-of-the-art pre-trained Large Language Models (LLMs) that are highly effective for natural language understanding and generation. Specifically, the following models are chosen based on their performance, versatility, and ability to handle diverse tasks:

- Summarization Model: facebook/bart-large-cnn (BART)

Reason for Selection: This model is a transformer-based encoder-decoder architecture designed for tasks like text summarization. BART has proven to generate high-quality, coherent summaries of long text inputs, making it ideal for simplifying complex information while maintaining accuracy.

- Question Answering Model: distilbert-base-uncased-distilled-squad (DistilBERT)

Reason for Selection: DistilBERT is a lightweight version of BERT (Bidirectional Encoder Representations from Transformers), optimized for speed and efficiency while retaining much of the accuracy. Trained on the SQuAD dataset, it is well-suited for extracting precise answers from input text, especially in cases of college-related queries.

- Fitness Tracking and Conversational Logic:

Custom logic is implemented alongside LLMs to manage user data such as fitness logs and goals, ensuring real-time, dynamic interactions.

Reason for Selection: While LLMs handle natural language input and generate human-like responses, custom functions help manage structured data, enabling the bot to offer practical functionality for tracking and managing fitness activities.

The combination of these models allows for a broad range of functionalities while maintaining high efficiency and accuracy.

# CODE

```
# TEAM ASK

!pip install transformers
!pip install torch

# Import necessary libraries
from transformers import pipeline
import datetime

# Load pre-trained models from Hugging Face
summarizer = pipeline('summarization', model="facebook/bart-large-cnn")
qa_model = pipeline('question-answering', model="distilbert-base-uncased-distilled-squad")

# Fitness data dictionary to store workout logs
fitness_data = {}
fitness_goals = {}

# Function to answer custom questions (college-related, etc.)
def answer_question(question):
    follow_up = question.lower()
    if 'contact' in follow_up:
        print("\033[33mBot:\033[0m You can email us at ourcollege@gmail.com or reach us at 123456789.")
    elif 'fees' in follow_up:
        print("\033[33mBot:\033[0m The fees for different courses vary. You can find a detailed fee structure on the college website.")
    elif 'hostel' in follow_up:
        print("\033[33mBot:\033[0m Hostel facilities are available for both boys and girls. The application process begins after admissions.")
    elif 'faculty' in follow_up:
        print("\033[33mBot:\033[0m Our faculty members are highly qualified and come from prestigious backgrounds. You can check their profiles on our website.")
    elif 'courses' in follow_up or 'programs' in follow_up:
        print("\033[33mBot:\033[0m Our college offers Computer Science, Electrical Engineering, and Mechanical Engineering.")
    elif 'admission' in follow_up:
        print("\033[33mBot:\033[0m The admission process includes submitting an online application, entrance exam, and interview.")
    else:
        print("\033[33mBot:\033[0m I'm sorry, I couldn't find information on that. Please choose from contact, fees, hostel, faculty, courses, or admission.")
```



# CODE

```
# Function to summarize text
def summarize_text(text):
    input_length = len(text.split())
    max_length = min(max(30, input_length // 2), 200)
    summary = summarizer(text, max_length=max_length, min_length=7, do_sample=False)
    return summary[0]['summary_text']

# Function to log fitness workout data
def log_workout(user, exercise, calories):
    if user not in fitness_data:
        fitness_data[user] = []
    fitness_data[user].append({
        'exercise': exercise,
        'calories': calories,
        'date': datetime.datetime.now()
    })
    return f"Workout logged: {exercise}, {calories} calories"

# Function to view workout history
def view_workout_history(user):
    if user in fitness_data and fitness_data[user]:
        history = ""
        for idx, log in enumerate(fitness_data[user]):
            history += f"{idx + 1}. {log['date']}: {log['exercise']} - {log['calories']} calories\n"
        return history.strip()
    else:
        return "\033[31mNo workout history found.\033[0m"

# Function to delete a specific workout entry
def delete_workout(user, entry_number):
    if user in fitness_data and fitness_data[user]:
        if 0 < entry_number <= len(fitness_data[user]):
            removed_entry = fitness_data[user].pop(entry_number - 1)
            return f"Removed workout: {removed_entry['exercise']} - {removed_entry['calories']} calories"
```

# CODE

```
        else:
            return "\033[31mInvalid entry number.\033[0m"
    else:
        return "\033[31mNo workout history found.\033[0m"

# Function to calculate total calories burned
def total_calories_burned(user):
    if user in fitness_data:
        total = sum(log['calories'] for log in fitness_data[user])
        return f"Total calories burned: {total} calories"
    else:
        return "\033[31mNo workout history found.\033[0m"

# Function to set fitness goals
def set_fitness_goal(user, goal):
    fitness_goals[user] = goal
    return f"Fitness goal set to: {goal}"

# Function to view fitness goal
def view_fitness_goal(user):
    if user in fitness_goals:
        return f"Your fitness goal: {fitness_goals[user]}"
    else:
        return "\033[31mYou have not set any fitness goals.\033[0m"

# Function for basic greetings
def greet_user(user_input):
    greetings = ["hello", "hi", "hey", "greetings", "good morning", "good evening"]
    responses = [
        "Hello! How can I assist you today?",
        "Hi there! What can I do for you?",
        "Hey! I'm here to help you.",
        "Greetings! What would you like to know?",
        "Good morning! What can I help you with today?",
```

# CODE

```
"Good evening! How can I assist you?"
]

for i, greeting in enumerate(greetings):
    if greeting in user_input:
        return responses[i]
return None

# Chatbot interaction function
def chatbot():
    print("\n👋 \033[33mHello there! I'm your personal assistant, here to make your life easier and more organized. \nWhether you need a quick text s

    while True:
        user_input = input("\033[36mYou: \033[0m").lower()

        # Check for greetings
        greeting_response = greet_user(user_input)
        if greeting_response:
            print(f"\033[33mBot:\033[0m {greeting_response}")
            continue

        if 'summarize' in user_input:
            text = input("Enter the text you want to summarize: ")
            if len(text.split()) < 10:
                print("\033[33mBot:\033[0m \033[31mPlease enter text with atleast 10 words and try again!\033[0m")
                continue
            else:
                summary = summarize_text(text)
                print(f"\033[33mBot:\033[0m Here's your summary: {summary}")

        elif 'question' in user_input or 'college' in user_input:
            question = input("Please specify what you would like to know: contact, fees, hostel, faculty, courses, or admission: ")
            answer = answer_question(question)
```



# CODE

```
elif 'fitness' in user_input:
    action = input("Do you want to log a workout, view history, delete an entry, set a goal, or view your goal? (log/view/delete/set/goal): ").lower()
    if action == "log":
        exercise = input("What exercise did you do? ")
        calories = int(input("How many calories did you burn? "))
        response = log_workout('User1', exercise, calories)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "view":
        history = view_workout_history('User1')
        print(f"\033[33mBot:\033[0m Your workout history:\n{history}")
    elif action == "delete":
        entry_number = int(input("Enter the entry number you want to delete: "))
        response = delete_workout('User1', entry_number)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "set":
        goal = input("What is your fitness goal? ")
        response = set_fitness_goal('User1', goal)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "goal":
        goal = view_fitness_goal('User1')
        print(f"\033[33mBot:\033[0m {goal}")
    else:
        print("\033[33mBot:\033[0m I didn't understand that. Please choose an appropriate action.")

elif 'total calories' in user_input:
    total = total_calories_burned('User1')
    print(f"\033[33mBot:\033[0m {total}")

elif 'exit' in user_input or 'bye' in user_input:
    print("\033[33mBot:\033[0m Goodbye! Have a good day.")
    break
```

# CODE

```
elif 'fitness' in user_input:
    action = input("Do you want to log a workout, view history, delete an entry, set a goal, or view your goal? (log/view/delete/set/goal): ").lower()
    if action == "log":
        exercise = input("What exercise did you do? ")
        calories = int(input("How many calories did you burn? "))
        response = log_workout('User1', exercise, calories)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "view":
        history = view_workout_history('User1')
        print(f"\033[33mBot:\033[0m Your workout history:\n{history}")
    elif action == "delete":
        entry_number = int(input("Enter the entry number you want to delete: "))
        response = delete_workout('User1', entry_number)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "set":
        goal = input("What is your fitness goal? ")
        response = set_fitness_goal('User1', goal)
        print(f"\033[33mBot:\033[0m {response}")
    elif action == "goal":
        goal = view_fitness_goal('User1')
        print(f"\033[33mBot:\033[0m {goal}")
    else:
        print(f"\033[33mBot:\033[0m I didn't understand that. Please choose an appropriate action.")

elif 'total calories' in user_input:
    total = total_calories_burned('User1')
    print(f"\033[33mBot:\033[0m {total}")

elif 'exit' in user_input or 'bye' in user_input:
    print(f"\033[33mBot:\033[0m Goodbye! Have a good day.")
    break

else:
    print(f"\033[33mBot:\033[0m I didn't understand that. Please ask about summarization, questions, or fitness.")

if __name__ == "__main__":
    chatbot()
```





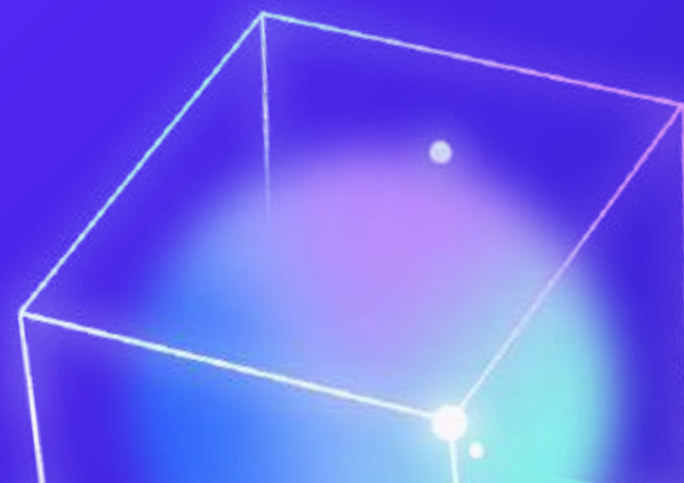
# OUTPUT

```
👋 Hello there! I'm your personal assistant, here to make your life easier and more organized.
Whether you need a quick text summary, have questions about our college, or want to track your fitness goals, I've got you covered!
Just type in your request, and let's get started on this journey together. 🙌💎
You: Hello, good evening
Bot: Hello! How can I assist you today?
You: Help me summarize a text
Enter the text you want to summarize: Machine learning is a subset of artificial intelligence that focuses on building systems that learn from and make decisions based on data.
Your max_length is set to 30, but your input_length is only 25. Since this is a summarization task, where outputs shorter than the input are typically wanted, you might consider
Bot: Here's your summary: Machine learning is a subset of artificial intelligence that focuses on building systems that learn from and make decisions based on data.
You: Summarize again other text
Enter the text you want to summarize: Python is a programming language
Bot: Please enter text with atleast 10 words and try again!
You: Help me with college related questions
Please specify what you would like to know: contact, fees, hostel, faculty, courses, or admission: About fees
Bot: The fees for different courses vary. You can find a detailed fee structure on the college website.
You: Fitness related
Do you want to log a workout, view history, delete an entry, set a goal, or view your goal? (log/view/delete/set/goal): Set
What is your fitness goal? go for cycling tomorrow
Bot: Fitness goal set to: go for cycling tomorrow
You: Fitness
Do you want to log a workout, view history, delete an entry, set a goal, or view your goal? (log/view/delete/set/goal): Goal
Bot: Your fitness goal: go for cycling tomorrow
You: Thanks! Bye
Bot: Goodbye! Have a good day.
```



# RESOURCES

- **Hugging Face Transformers Library:** <https://huggingface.co/transformers/>
- **Facebook BART Model (Summarization):** <https://huggingface.co/facebook/bart-large-cnn>
- **DistilBERT for Question Answering:** <https://huggingface.co/distilbert-base-uncased-distilled-squad>
- **Large Language Models (LLMs) and their Applications:**  
<https://towardsdatascience.com/what-are-large-language-models-llms-and-how-do-they-work-d5b5f84bb98a>





THANK YOU!

