rit | RAJALAKSHMI INSTITUTE OF TECHNOLOGY

BELIEVE IN THE POSSIBILITIES

AN AUTONOMOUS INSTITUTION

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

ACADEMIC YEAR 2025 - 2026

SEMESTER III

ARTIFICIAL INTELLIGENCE LABORATORY

MINI PROJECT REPORT

REGISTER NUMBER	
NAME	
PROJECT TITLE	
DATE OF SUBMISSION	
FACULTY IN-CHARGE	

Signature of Faculty In-charge

AI-Based Chatbot for Daily Study Tips

INTRODUCTION:

Brief overview of Artificial Intelligence concepts:

- AI enables systems to perform tasks that normally require human intelligence, such as natural language understanding, personalization, and recommendation.
- Chatbots combine NLP, dialog management, and personalization to provide interactive experiences.

Project introduction:

- This project implements an AI-based chatbot designed to deliver short, tailored daily study tips to students.
- The bot helps users build study routines by sending or showing bite-sized advice, scheduling reminders, and personalizing tips by subject, study time, and learning preferences.
- The project demonstrates practical AI/NLP concepts (intent recognition, personalization), a simple rule-based + ML-ish approach, and a Python implementation.

PROBLEM STATEMENT:

Clear and precise:

- Many students lack structured daily study habits and forget small but effective study practices. The challenge is to create a lightweight chatbot that delivers relevant daily study tips, adapts to user preferences, and can be easily extended.

GOAL:

Expected result and possibilities:

- Deliver daily, personalized study tips to users via chat (CLI / Web API).
- Let users set preferences (subject, difficulty level, preferred time).
- Demonstrate a working prototype with a simple text-based interface or REST API.
- Possibilities: integrate with messaging platforms, add push notifications, or add an ML model to better personalize.

THEORETICAL BACKGROUND:

Theoretical background about the problem and algorithms:

- **NLP basics**: tokenization, intent classification, entity extraction. For this project, intent detection can be simple rule-based or classifier-based.
- **Personalization**: keep user profile (subject preference, daily goal) and use simple scoring/relevance matching to select tips.
- **Scheduling**: use a scheduler (cron-like or APScheduler) to push tips at preferred times.
- **Recommendation approaches**:
    - Rule-based mapping (tag tips by subject/time-of-day).
    - Collaborative filtering (future enhancement) where tips favored by similar users are ranked higher.
    - Lightweight supervised classification to detect user intents (e.g., "get tip", "change subject").

Literature survey of the given problem and other algorithms:

- Lightweight chatbots often use rule-based intent mapping or small neural classifiers when resources are limited.
- Reinforcement learning or bandits can be applied for tip selection to maximize engagement (future work).

Justification for choosing the algorithm:

- For a course/semester project, a hybrid rule-based + simple personalization approach is easy to implement, explain, and evaluate.

ALGORITHM EXPLANATION WITH EXAMPLE:

High-level algorithm

1. **Load tips database** — each tip has tags: subjects, time-of-day, complexity, and keywords.
2. **User profile** — store preferences: preferred subject(s), preferred tip time, difficulty.
3. **Intent parsing** — detect whether the user asks for a tip, updates preferences, or wants help.
4. **Select tip**:
    - Filter tips by subject and difficulty.
    - Score tips by freshness (not served recently), match to preferences, and variety.

    - Return the top-scoring tip.

5. **Schedule** — if user requests daily tips, schedule a job at the preferred time.
6. **Logging and feedback** — allow the user to upvote/downvote tips for personalization.

Example

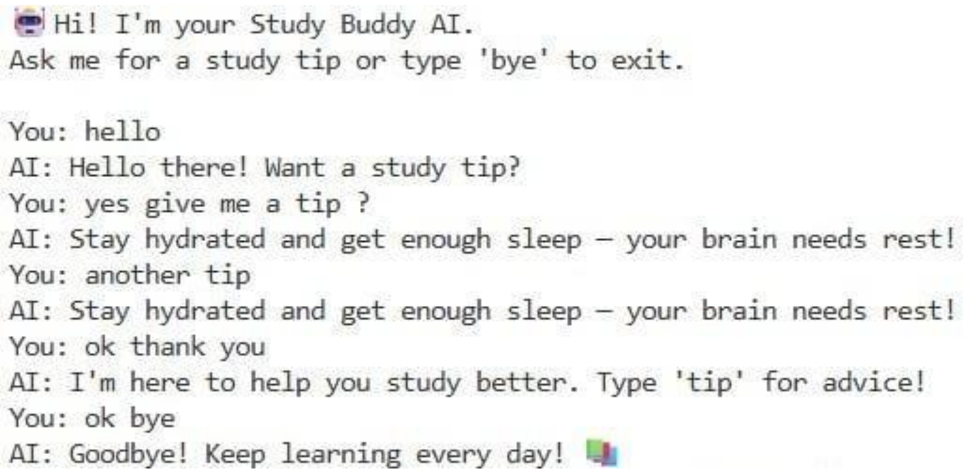
- User sets subject = "Math", preferred_time = "20:00".
- At 20:00, scheduler triggers: selects a tip tagged [math, evening] that the user hasn't seen in the last 7 days.
- Bot sends: "Study Tip: Practice 5 algebra problems focusing on solving for x; review mistakes immediately."

IMPLEMENTATION AND CODE:

```
import random
import datetime
study_tips = [
    "Take short breaks every 45 minutes to refresh your mind.",
    "Review your notes within 24 hours of learning something new.",
    "Use active recall instead of passive rereading.",
    "Stay hydrated and get enough sleep — your brain needs rest!",
    "Set small, achievable goals for each study session.",
    "Teach what you've learned to someone else — it helps you remember.",
    "Eliminate distractions like your phone while studying.",
    "Use the Pomodoro technique: 25 minutes study, 5 minutes break.",
    "Organize your study space — a tidy desk helps a tidy mind.",
    "Revise regularly instead of cramming before exams."
]
def get_daily_tip():
    today = datetime.date.today()
    random.seed(today.toordinal())  # ensures same tip per day
    return random.choice(study_tips)
```

```
def chatbot():
    print("🤖 Hi! I'm your Study Buddy AI.")
    print("Ask me for a study tip or type 'bye' to exit.\n")
  while True:
        user_input = input("You: ").lower()
        if "tip" in user_input:
            print("AI:", get_daily_tip())
        elif "hello" in user_input or "hi" in user_input:
            print("AI: Hello there! Want a study tip?")
        elif "bye" in user_input:
        print("AI: Goodbye! Keep learning every day! 📚")
            break
        else:
            print("AI: I'm here to help you study better. Type 'tip' for advice!")
if _name_ == "_main_":  # ✅ Corrected line
    chatbot()
```

OUTPUT:



```
🤖 Hi! I'm your Study Buddy AI.
Ask me for a study tip or type 'bye' to exit.

You: hello
AI: Hello there! Want a study tip?
You: yes give me a tip ?
AI: Stay hydrated and get enough sleep — your brain needs rest!
You: another tip
AI: Stay hydrated and get enough sleep — your brain needs rest!
You: ok thank you
AI: I'm here to help you study better. Type 'tip' for advice!
You: ok bye
AI: Goodbye! Keep learning every day! 📚
```

RESULTS AND FUTURE ENHANCEMENT:

Possibility and comparison with others:

- This solution is light-weight and easy to deploy compared to heavy ML models.
- Advantages: quick to implement, predictable, privacy-friendly (data stored locally).
- Limitations: personalization is basic; no deep understanding of user intents.

Future enhancements:

1. **NLP Intent Model**: integrate a small intent classifier (e.g., fine-tune lightweight transformer or use spaCy) to parse varied user inputs (e.g., “I need a review tip for physics”).
2. **Recommender**: implement a contextual bandit or collaborative filtering to optimize tip selection based on engagement.
3. **Multi-channel Delivery**: add integrations for Telegram, WhatsApp, email, and mobile push.
4. **Rich content**: embed short micro-lessons or quick exercises with automated grading.
5. **A/B testing**: measure which types/timing of tips increase study time or retention.
6. **Analytics dashboard**: visualize engagement, upvotes/downvotes, and tip performance.

GITHUB LINK OF THE PROJECT AND REPORT:

References :

Python Official Documentation

- Python Software Foundation. (2025). *Python Standard Library Documentation.*
  👉 https://docs.python.org/3/
  (Used for understanding built-in modules like random and datetime.)

2. **W3Schools – Python Chatbot Example**

   - W3Schools. (2024). *Python Chatbot Tutorial.*
     👉 https://www.w3schools.com/python/python_chatbot.asp
     (Basic examples of chatbot loops and condition-based replies.)

3. **GeeksforGeeks – Simple Chatbot in Python**
   - GeeksforGeeks. (2024). *Creating a Chatbot in Python.*
     👉 https://www.geeksforgeeks.org/how-to-make-a-chatbot-in-python/
     (Provides a foundation for logic-based chatbots similar to this project.)

4. **Real Python – Random Module Tutorial**
   - Real Python. (2024). *Generating Random Data in Python.*
     👉 https://realpython.com/python-random/
     (Used to understand how random.choice() works for selecting daily tips.)

5. **Khan Academy – Study Tips and Learning Habits**
   - Khan Academy. (2025). *Effective Study Techniques.*
     👉 https://www.khanacademy.org/college-careers-more/learn-how-to-learn
     (Used as inspiration for the actual study tips included in the chatbot.)