

STUDY BOT – AI STUDY ASSISTANT WITH MEMORY

Submitted by: Mohammed Tousif Malekoppa

Email: tousif8722@gmail.com

Course: Bachelor of Engineering (Computer Science)

Submission Date: 21 February 2026

1. PROJECT OVERVIEW

Study Bot is an AI-powered chatbot designed to assist students with academic questions. The chatbot accepts user queries related to study topics and generates intelligent responses using a Large Language Model (LLM). The system stores conversation history in a MongoDB database and retrieves previous messages to provide contextual responses. This enables the chatbot to simulate real-world AI assistants that remember interactions and provide meaningful replies.

2. IMPLEMENTATION STEPS

Step 1 - Environment Setup

Installed required packages such as FastAPI, Uvicorn, LangChain, and MongoDB drivers. Environment variables were configured for API keys and database URI.

Step 2 - Basic Chatbot Creation

Connected backend to LLM API. The chatbot accepts user input and returns generated responses.

Step 3 - Database Memory Integration

Connected MongoDB database. Stored user messages and chatbot responses.

Retrieved past messages to maintain conversation context.

Step 4 - Study Bot Logic

Added system prompt to restrict chatbot responses to academic and study-related topics.

Step 5 - Deployment

Deployed the chatbot API using Render platform and configured environment variables for secure execution.

3. MEMORY IMPLEMENTATION

The chatbot stores conversation history inside MongoDB database. Each interaction is stored as a record containing:

- User message
- Bot response
- Timestamp

When a new query is received, the system retrieves previous conversations and appends them to the prompt before sending it to the LLM. This allows the chatbot to generate context-aware responses.

Importance of Memory:

Without memory → responses are isolated

With memory → responses are contextual and intelligent

4. GITHUB REPOSITORY

Project Source Code Link:

https://github.com/mohammed-tousif/AI_chatBot.git

5.HOSTED API LINK

Live API URL:

<https://ai-chatbot-gnmt.onrender.com>

API Documentation:

<https://ai-chatbot-gnmt.onrender.com/docs>

The chatbot API has been successfully deployed on the Render cloud platform. The live endpoint accepts user queries and returns AI-generated responses in real time. The Swagger documentation endpoint provides interactive testing of API routes.

Sample API test screenshot:

https://ai-chatbot-gnmt.onrender.com/docs#/default/chat_chat_post

The screenshot shows a browser-based API testing interface for the AI Chatbot. At the top, there's a curl command:

```
curl -X 'POST' \
  'https://ai-chatbot-gnmt.onrender.com/chat' \
  -H 'Accept: application/json' \
  -H 'Content-type: application/json' \
  -d '{
    "user_id": "user123",
    "question": "what kind of ChatBot are you"
}'
```

Below the curl command, the Request URL is displayed as <https://ai-chatbot-gnmt.onrender.com/chat>. The Server response section shows a status code of 200. The Response body contains a JSON object describing the AI's capabilities and focus areas. The Response headers section lists standard HTTP headers like Access-Control-Allow-Origin, Content-Type, and Server-Timing.

The screenshot shows the MongoDB Cloud Data Explorer interface. The left sidebar displays the organization (none), project (ChatBot), and a navigation tree for clusters (Cluster0, ChatBot, users, admin, local, oplog.rs). The main area shows the 'users' collection with 33 documents. A specific document is selected, displaying its details:

```
_id: ObjectId('69995a8719579ae5ca2fc6e8')
user_id: "user123"
role: "assistant"
message: "I don't actually know your name--unless you tell me, I'm just a virtual..."
timestamp: 2026-02-21T08:05:52.436+00:00
```

The screenshot shows the Render log viewer for the AI_chatBot instance. The left sidebar contains navigation links such as Dashboard, AI_chatBot, Events, Settings, MONITOR, Logs, Metrics, MANAGE, Environment, Shell, Scaling, Previews, Disk, Jobs, Changelog, Invite a friend, Contact support, and Render Status. The main area shows the log for February 21, 2026, at 5:05 PM, with a message indicating the instance will spin down if inactive. The log output is as follows:

```
Feb 21, 2026 at 5:05 PM Live
ca90ef7 deleted the unwanted zip file

All logs ▾ Q Search Feb 21, 5:04 PM - 5:09 PM GMT+5:30 ...
```

```
05:07:42 PM    >>> Setting WEB_CONCURRENCY=1 by default, based on available CPUs in the instance
05:07:42 PM    >>> Deploying...
05:08:11 PM [59fxt] >>> Running 'unicorn app:app --host 0.0.0.0 --port $PORT'
05:08:27 PM [59fxt] INFO: Started server process [55]
05:08:27 PM [59fxt] INFO: Waiting for application startup.
05:08:27 PM [59fxt] INFO: Application startup complete.
05:08:27 PM [59fxt] INFO: Unicorn running on http://0.0.0.0:10000 (Press CTRL+C to quit) ...
05:08:28 PM [59fxt] INFO: 127.0.0.1:51760 - "HEAD / HTTP/1.1" 405 Method Not Allowed
05:08:33 PM    >>> Your service is live!
05:08:33 PM    >>> -----
05:08:33 PM    >>> -----
05:08:33 PM    >>> Available at your primary URL https://ai-chatbot-gmmt.onrender.com
05:08:33 PM    >>> -----
05:08:33 PM    >>> -----
05:08:34 PM [59fxt] INFO: 34.145.120.165:0 - "GET / HTTP/1.1" 200 OK
```