

Study Bot - AI Powered Learning Assistant

Project Report

Aditya Kumar Singh,
B.Tech Final Year Student,
KIIT University, Bhubaneswar

 [LinkedIn](#)

 [My Portfolio](#)

1. Project Overview

Study Bot is an AI-powered chatbot designed to answer study-related questions while maintaining user-specific conversation memory. The system allows users to register/login securely and stores chat history in MongoDB Atlas.

The backend is built using FastAPI and integrates a Large Language Model via Groq API. The frontend is developed using HTML, CSS, and JavaScript to provide a responsive chat interface.

2. Tech Stack Used

- Backend: FastAPI (Python)
- Database: MongoDB Atlas
- AI Model: Groq LLM (openai/gpt-oss-20b)
- Authentication: Passlib (pbkdf2_sha256 & bcrypt hashing)
- Frontend: HTML, CSS, JavaScript
- Deployment: Render

3. Memory Implementation (How Chat History Works)

Memory is implemented using MongoDB. Each message is stored with the following fields:

- user_id
- role (user or assistant)
- message content
- timestamp

When a user sends a new message, the system fetches all previous messages from MongoDB, sorts them chronologically, and passes them into the LLM using LangChain's MessagesPlaceholder. This allows contextual responses.

Both user and assistant messages are stored immediately after each interaction, ensuring persistent conversation memory.

4. Authentication System

Users must provide a username and password. If the username does not exist, a new account is created. Passwords are securely hashed using Passlib before storage.

Login validation ensures only authenticated users can access chat history and send messages.

5. API Endpoints

GET / → Health check endpoint

GET /health → Server health status

POST /auth → Register/Login user

POST /chat → Send question to Study Bot

POST /history → Retrieve previous chat history

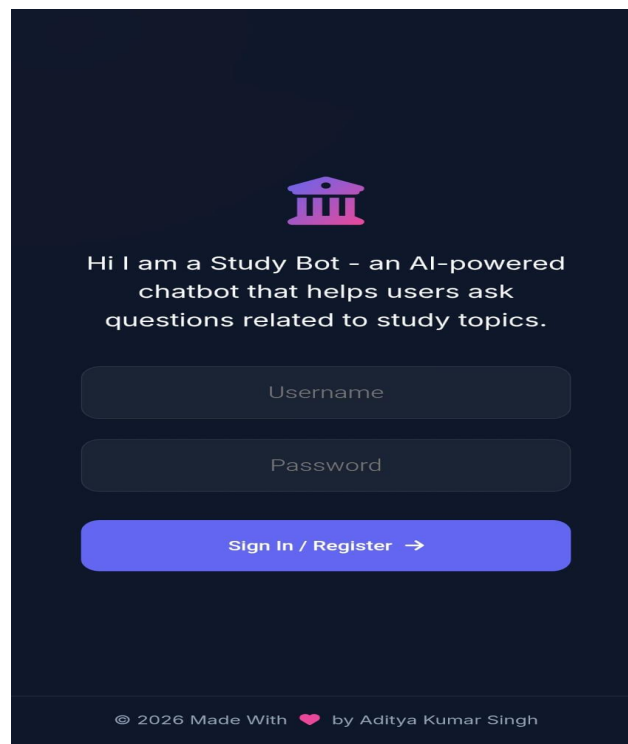
6. Frontend Description

The frontend interface includes a login/register screen and a chat interface. It communicates with the backend API using Fetch API.

Frontend Files Included:

- index.html
- script.js
- style.css

These files handle authentication, chat UI rendering, message formatting, typing indicators, and history synchronization.



7. GitHub Repository

<https://github.com/Adityaxletscode/StudyBot-----An-intelligent-AI-learning-companion>

8. Deployment Link

<https://adityaxletscode.github.io/StudyBot-----An-intelligent-AI-learning-companion/>

9. Hosted API Link

Render: <https://studybot-an-intelligent-ai-learning.onrender.com>

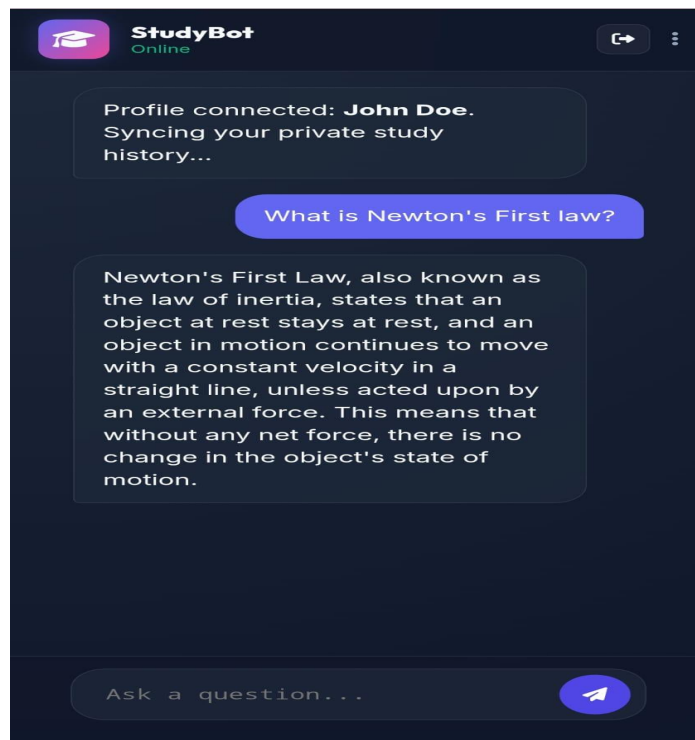
10. Sample API Testing Instructions

You can test the API using:

- Postman
- Thunder Client (VS Code)
- Browser (GET endpoints)

Example Chat Request Body:

```
{ "question": "What is Newton's First Law?", "user_id": "aditya", "password": "1234" }
```



11. Conclusion

Study Bot demonstrates the integration of AI, secure authentication, database-based memory, and cloud deployment into a full-stack AI application. It is a practical example of building an intelligent assistant with persistent memory.