



STARS Tutoring Chatbot

Presented by:

Richard Chong, Andres Lopez, Michael Camelio, Carlos Ramirez, Jason Martinez

CURRENT SYSTEM

Traditional academic support systems rely on in-person tutoring and static online resources. These systems are *limited* in scalability and accessibility, making real-time academic support *difficult* to achieve.



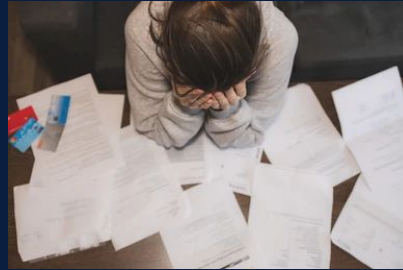
Challenges:

- Limited availability of tutors.
- Inability to provide 24/7 academic support.
- Difficulty scaling traditional solutions to meet growing student needs.



PROBLEM STATEMENT

Students need 24/7 academic support, but traditional systems fail to provide scalable and accessible solutions.



Our Objective:

To develop an **AI-driven** tutoring **chatbot** capable of real-time interaction, built on a robust, scalable, and secure infrastructure.

This chatbot will **assist** students and guide them step by step, ensuring they understand the entire process.

TASK AND GOALS



Team Tasks:

- Configure a cloud-based infrastructure to support the chatbot.
- Integrate **OpenAI** APIs for real-time AI responses.
- Develop a user-friendly interface using **Streamlit**.
- Test the chatbot thoroughly to ensure reliability.
- Implementing a database using **MongoDB**



Goals:

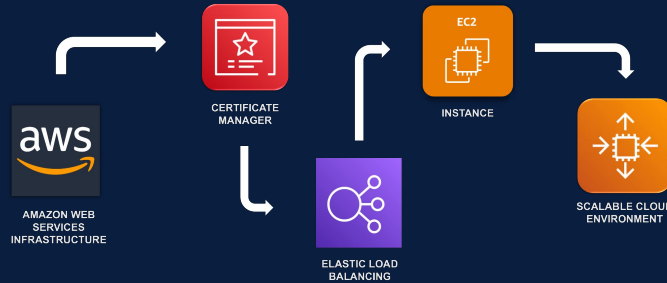
- Provide students with a responsive, AI-driven tutoring solution.
- Maintain *secure, scalable, and cost-effective* system operations.

PROPOSED SOLUTION

The STARS GPT chatbot leverages AWS EC2 for backend hosting, OpenAI's API for AI-driven responses, and Streamlit for an intuitive frontend interface.

Technologies Used:

- AWS EC2 for secure, scalable cloud hosting.
- OpenAI API for advanced natural language processing.
- Streamlit for interactive user engagement.
- MongoDB for storing data





KEY FEATURES

- **Create and personalize account:**
Able to create an account and returning users are able to login.
- **AI-Driven Tutoring:**
Provides real-time academic assistance using OpenAI GPT models.
- **Scalable Cloud Infrastructure:**
AWS EC2 supports high-demand scenarios with minimal latency.
- **Interactive Frontend:**
Streamlit offers a simple yet powerful interface for users.
- **Secure and Reliable:**
The system incorporates SSH protocols and robust error handling.
- **Help User Understand, Not Just Providing Answers:**
Our application is set up to ensure users are guided throughout the process and are learning at every step.

StarsGPT

User: Can you help me understand interfaces in Java?

Assistant: Absolutely! Interfaces in Java are a fundamental concept in object-oriented programming. They allow you to define a contract that classes can implement. Here are some key points to understand about interfaces:

What is an Interface?

- An interface is a reference type in Java, similar to a class, that can contain only constants, method signatures, default methods, static methods, and nested types.
- Interfaces cannot contain instance fields or constructors.

Purpose of Interfaces

- Interfaces are used to achieve abstraction and multiple inheritance in Java.
- They allow you to define methods that must be implemented by any class that chooses to implement the interface.

Defining an Interface

You define an interface using the `interface` keyword. Here's a simple example:





CHALLENGES AND SHORTCOMINGS

Challenges

- Loss of SSH keys requiring manual restoration using EBS volume adjustments.
- Initial halting issues during Streamlit deployment, resolved by implementing a startup delay script.
- API error handling and debugging during OpenAI integration.
- Having to modify certain items to exact specifications on Streamlit.

Shortcomings:

- Optimization for larger-scale traffic is pending.
- Limited personalization features in the chatbot responses.
- Limited personalization in the users account





FUTURE PLANS

- Implement load balancing and auto-scaling for higher traffic.
- Add personalized AI features tailored to individual students.
- Develop multi-language support to expand accessibility.
- Explore CI/CD pipelines for smoother deployment processes.
- Implement access to new and different API's to expand usage
- Allow actual tutors to view students' chat history, in order to better assist them and view the key areas they struggled with





SUMMARY

The STARS GPT project aimed to create a scalable, secure, and efficient AI-driven tutoring chatbot to improve academic support.

Expected Impact:

- 24/7 academic assistance for students.
- Seamless integration of backend, AI, and frontend components.
- A reliable foundation for future enhancements.

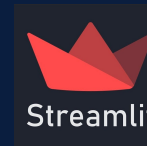


ACKNOWLEDGMENTS

Thanks to Dr. Patricia McDermott-Wells for her mentorship.



Thanks to AWS, OpenAI, Streamlit, and MongoDB for providing the tools and documentation that made this project possible.



FIU

Engineering
& Computing