



Florida International University

Fall 2024 Senior Design Project

STARS GPT

Students: Jason Martinez

Mentor: *Patricia Mcdermott-Wells*

Instructor/Faculty: *Masoud Sadjadi*, Florida International University

PROBLEM

Allow the addition of AI Tutoring for students in an accessible, easy-to-use environment through FIU STARS.

CURRENT SYSTEM

The current system utilizes a brand new database, alongside editing the API Endpoints to be used for OpenAI Integration.

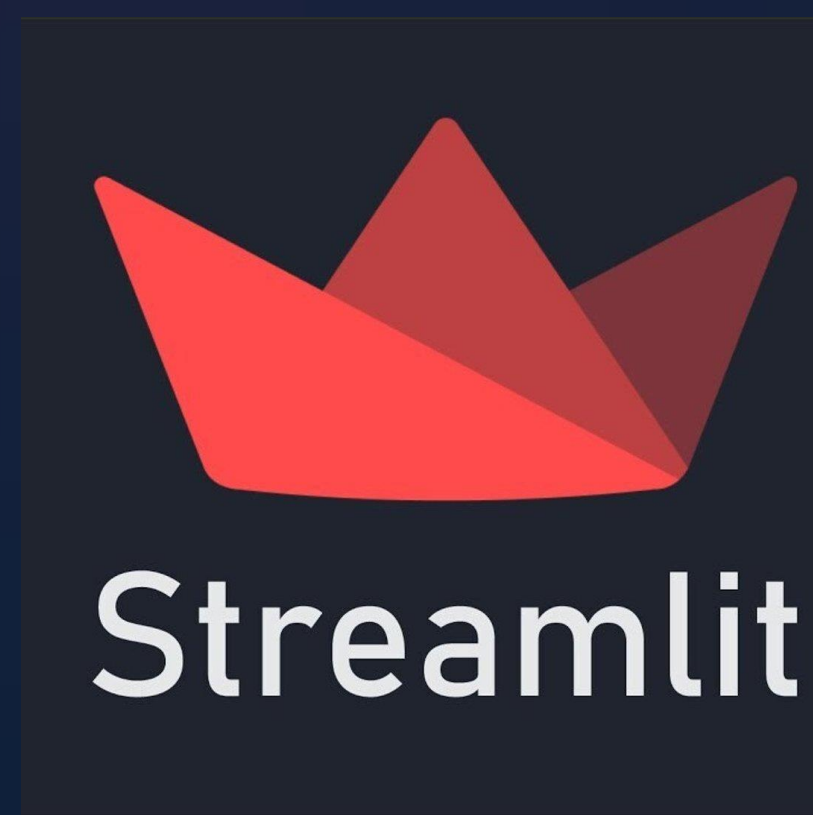
REQUIREMENTS

For us, we had to:

1. Make front end to integrate model with
2. Setting up back end to AI System
3. Setting up Imports for future development
4. Set up new functionalities and upgrading current ones

I focused on 1, 2, and 3 for the project.

SYSTEMS LEARNED



mongoDB Compass



INDIVIDUAL IMPLEMENTATION



EXPLANATION ON IMPLEMENTATION

Pics 1-2.

Implementation of new front end design using Canva, inspired by FIU Canvas/MyFIU and implemented it using Streamlit. Also worked on concept designs for what the STARS GPT Logo could look like, but I couldn't figure out a final design.

The full implementation through streamlit wasn't 100% finished, unfortunately, but the idea is laid out for future implementation.

Pic 3. I also created CRUD API endpoints for the database. It was finished, and very simple since it was my first time implementing one, but it had to be scrapped since Michael created a much better one for the application.

```
# used gpt for the imports only
from fastapi import FastAPI, HTTPException, Depends
from sqlalchemy import Session
from sqlalchemy import create_engine, Column, Integer, String, Text, DateTime
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker
from pydantic import BaseModel
from datetime import datetime # Import datetime for timestamp
# Uses FastAPI and SQLAlchemy mainly
# INSTALL PSYCOPG2 (pip install psycopg2) IF USING POSTMAN
# DOCS I used: https://docs.sqlalchemy.org/en/13/index.html
# https://fastapi.tiangolo.com/

DATABASE_URL = "postgresql://username:password@localhost/mydatabase" # TODO: get HOST username/password; using this as placeholder till i do
engine = create_engine(DATABASE_URL) # produces engine object based off URL
SessionLocal = sessionmaker(autocommit = False, autoflush = False, bind = engine)
app = FastAPI()

Base = declarative_base() #base class from where all mapped classes inherit

# fields inspired off gpt prompt, could change in future
class ChatInteraction(Base): #uses base as a class
    """tablename = "chat_interactions" #gives a name to the table"""
    id = Column(Integer, primary_key = True, index = True) #Column() defines column in database table and also gives
    user_question = Column(Text, nullable = False)
    ai_response = Column(Text, nullable = False)
    timestamp = Column(DateTime, nullable = False, default = datetime.utcnow) # DateTime stores time where interaction occurred

class ChatInteractionCreate(BaseModel):
    user_question: str
    ai_response: str
    timestamp: datetime

class ChatInteractionResponse(BaseModel):
    id: int
    user_question: str
    ai_response: str
    timestamp: datetime

class Config:
    orm_mode = True #Just supports mapping to ORM objects (Object Relational Mapping - connects OOP code to database)
```

FIU

Engineering
& Computing

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by XXXXX (name of the project sponsor: federal, state or local government, or corporate partners, where applicable). We/I thank XXXX for his/her/their assistance and mentorship that I/we received throughout the senior design project.

FLORIDA INTERNATIONAL UNIVERSITY