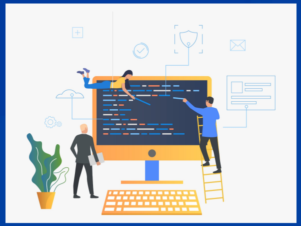
The Development and Introduction in Computer-Science Courses of an AI Tutor

Master's Project Presentation

Joshua Zingale
December 10, 2025
Full Report:
https://github.com/joshua-zingale/masters-project

Presentation Outline

Related Works



Development

Deployment

AI Tutor

Student Data

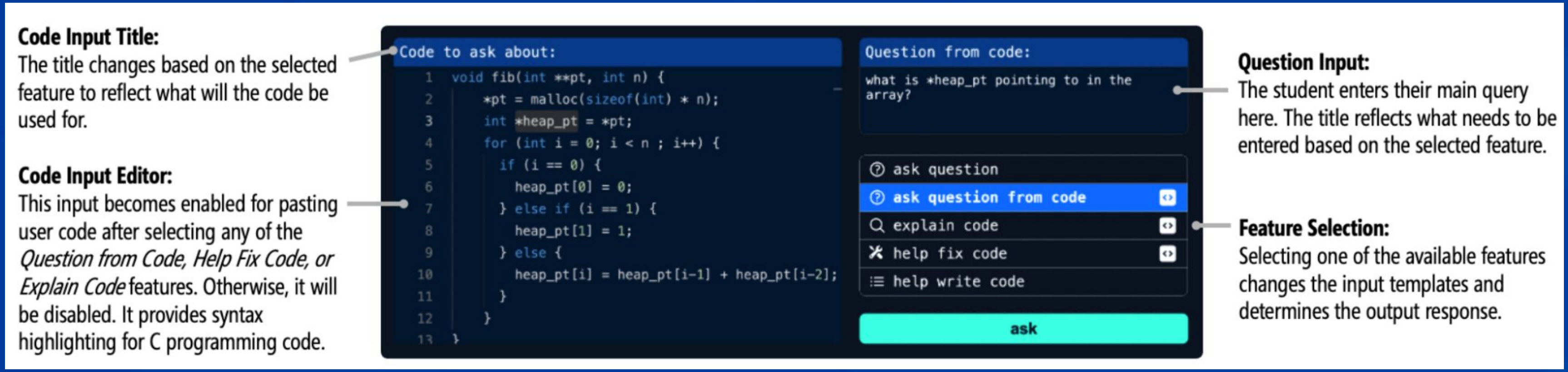
Future

UC RIVERSIDE

Related Works

Related Works

- Jia et al., 2024 & Neyem et al., 2024
    - ➢ Automatic feedback on reports submitted by students
- Taylor et al., 2024
    - ➢ Integrated LLM into compiler for enhanced error messages for students
- Kazemitabaar et al., 2024
    - ➢ Interactive code helper, “CodeAid”



Code Input Title: The title changes based on the selected feature to reflect what will the code be used for.

Code Input Editor: This input becomes enabled for pasting user code after selecting any of the *Question from Code, Help Fix Code, or Explain Code* features. Otherwise, it will be disabled. It provides syntax highlighting for C programming code.

Code to ask about:

```c
void fib(int **pt, int n) {
    *pt = malloc(sizeof(int) * n);
    int *heap_pt = *pt;
    for (int i = 0; i < n ; i++) {
      if (i == 0) {
        heap_pt[0] = 0;
      } else if (i == 1) {
        heap_pt[1] = 1;
      } else {
        heap_pt[i] = heap_pt[i-1] + heap_pt[i-2];
      }
    }
}
```

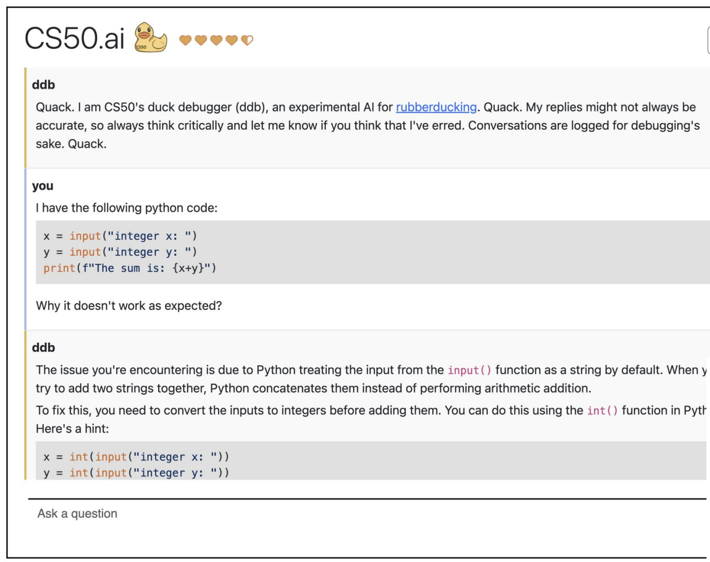
Question from code:

what is *heap_pt pointing to in the array?

Question Input: The student enters their main query here. The title reflects what needs to be entered based on the selected feature.

Feature Selection: Selecting one of the available features changes the input templates and determines the output response.

Harvard’s CS50 Duck



ddb

Quack. I am CS50's duck debugger (ddb), an experimental AI for rubberducking. Quack. My replies might not always be accurate, so always think critically and let me know if you think that I've erred. Conversations are logged for debugging's sake. Quack.

you

I have the following python code:

```
x = input("integer x: ")
y = input("integer y: ")
print(f"The sum is: {x+y}")
```

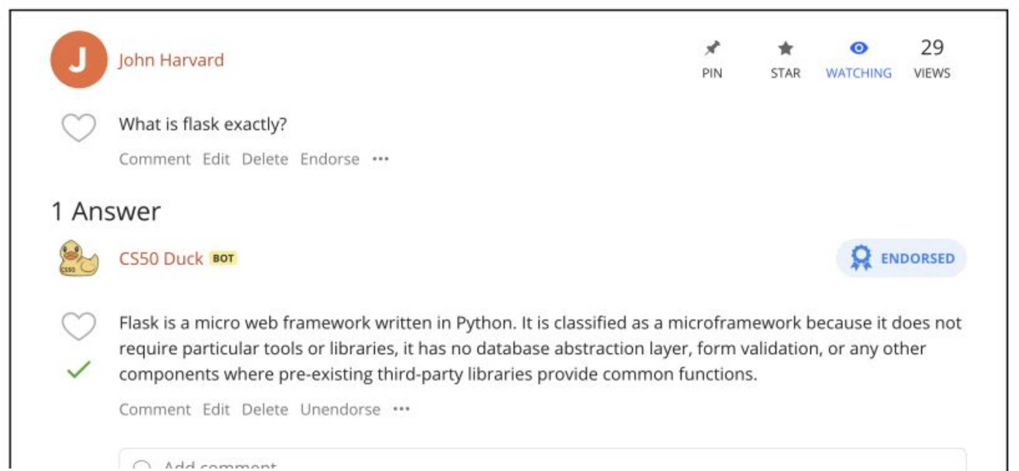
Why it doesn't work as expected?

ddb

The issue you're encountering is due to Python treating the input from the `input()` function as a string by default. When y
try to add two strings together, Python concatenates them instead of performing arithmetic addition.

To fix this, you need to convert the inputs to integers before adding them. You can do this using the `int()` function in Pyth
Here's a hint:

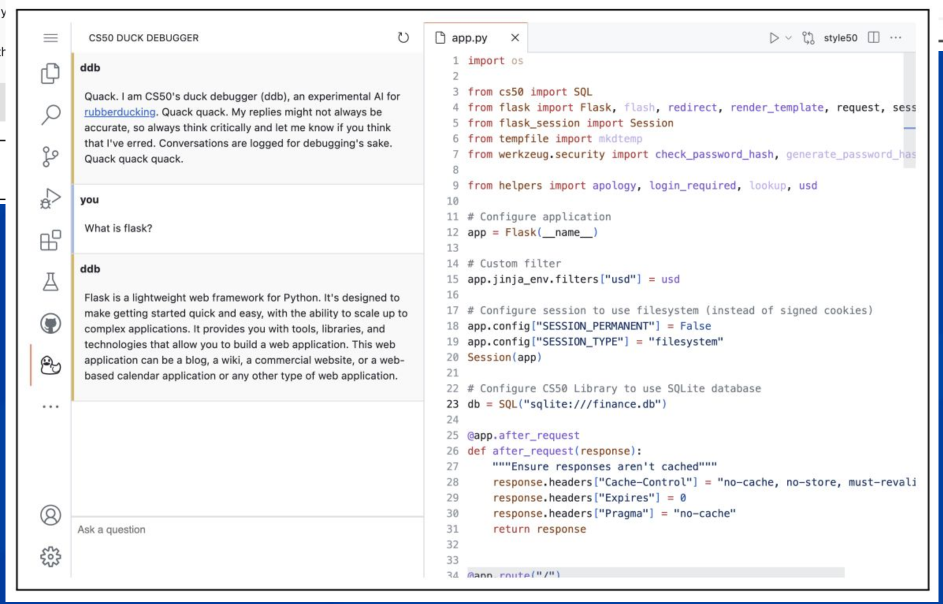
```
x = int(input("integer x: "))
y = int(input("integer y: "))
```



What is flask exactly?

1 Answer

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries, it has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.



ddb

Quack. I am CS50's duck debugger (ddb), an experimental AI for rubberducking. Quack quack. My replies might not always be accurate, so always think critically and let me know if you think that I've erred. Conversations are logged for debugging's sake. Quack quack quack.

you

What is flask?

ddb

Flask is a lightweight web framework for Python. It's designed to make getting started quick and easy, with the ability to scale up to complex applications. It provides you with tools, libraries, and technologies that allow you to build a web application. This web application can be a blog, a wiki, a commercial website, or a web-based calendar application or any other type of web application.

```
import os

from cs50 import SQL
from flask import Flask, flash, redirect, render_template, request, sess
from flask_session import Session
from tempfile import mkdtemp
from werkzeug.security import check_password_hash, generate_password_has

from helpers import apology, login_required, lookup, usd

# Configure application
app = Flask(__name__)

# Custom filter
app.jinja_env.filters["usd"] = usd

# Configure session to use filesystem (instead of signed cookies)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)

# Configure CS50 Library to use SQLite database
db = SQL("sqlite:///finance.db")

@app.after_request
def after_request(response):
    """Ensure responses aren't cached"""
    response.headers["Cache-Control"] = "no-cache, no-store, must-revali
    response.headers["Expires"] = 0
    response.headers["Pragma"] = "no-cache"
    return response


@app.route("/")
```

Liu et al., 2024

General Problems with Generative AI in Education

- Hallucination
  - ➢ General models often fabricate information, presenting false facts to students with high confidence.
- Oversharing
  - ➢ Standard chatbots may solve homework problems directly, hindering the learning process rather than aiding it.
- Lack of Context:
  - ➢ Off-the-shelf models are unaware of specific course policies, lecture notes, or local definitions.

Development

UC RIVERSIDE

Student-Driven Development

- **Six** upperclassmen from the Summer 2025 Data Science Fellowship
- **Eight** weeks of development and professional development
- **Five** development sprints
- **One** git repository
- **Continuous Integration** for automated code verification

Development Sprints

Each Print had a header containing

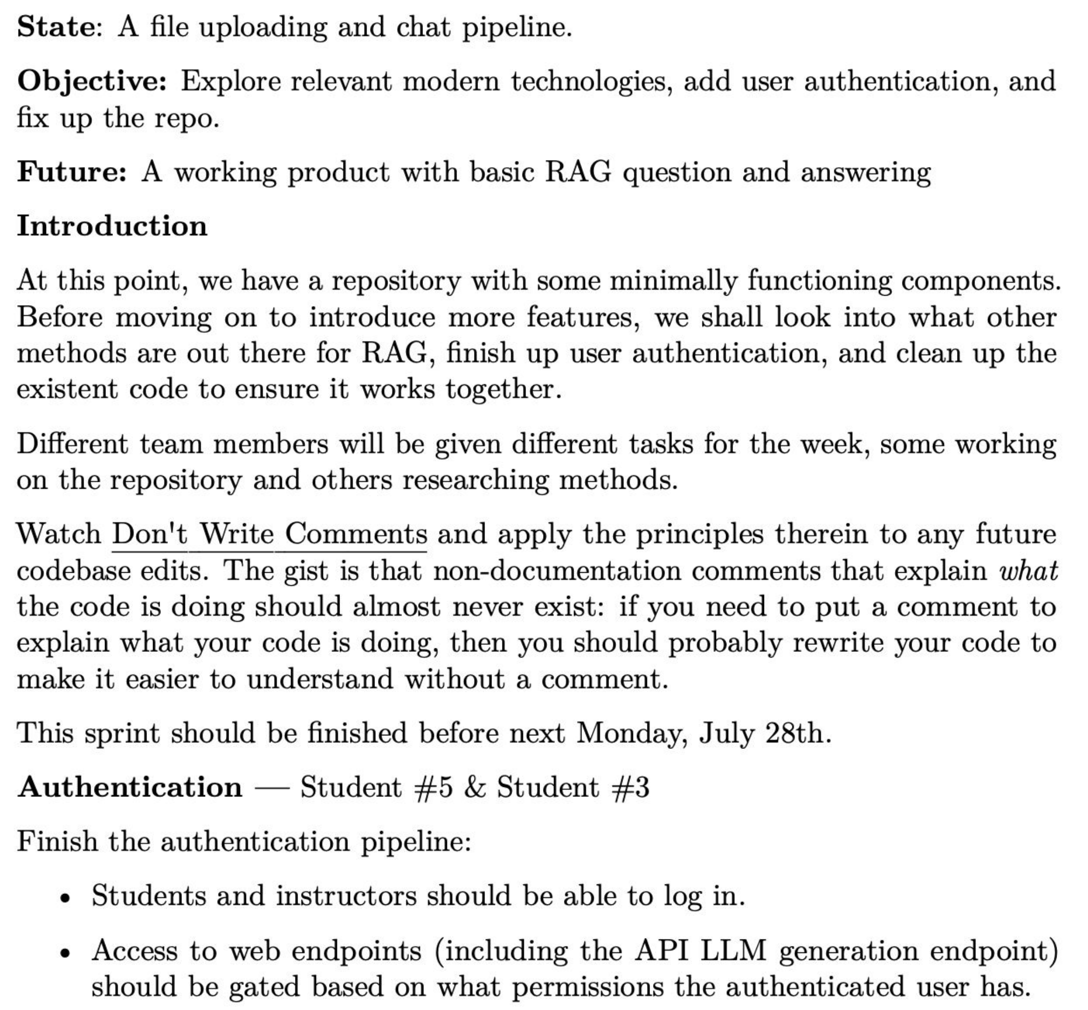
State, the current state of the project,

Objective, that to be accomplished during the sprint, and

Future, work to be accomplished after the sprint had been completed,

followed by an **Introduction**, then Individual and Group **Assignments.**

Example from a Sprint



State: A file uploading and chat pipeline.

Objective: Explore relevant modern technologies, add user authentication, and fix up the repo.

Future: A working product with basic RAG question and answering

Introduction

At this point, we have a repository with some minimally functioning components. Before moving on to introduce more features, we shall look into what other methods are out there for RAG, finish up user authentication, and clean up the existent code to ensure it works together.

Different team members will be given different tasks for the week, some working on the repository and others researching methods.

Watch Don't Write Comments and apply the principles therein to any future codebase edits. The gist is that non-documentation comments that explain *what* the code is doing should almost never exist: if you need to put a comment to explain what your code is doing, then you should probably rewrite your code to make it easier to understand without a comment.

This sprint should be finished before next Monday, July 28th.

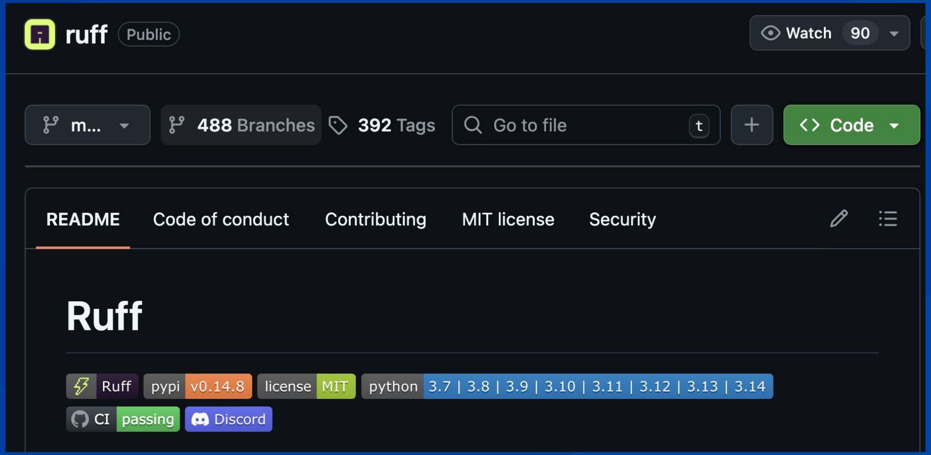
Authentication — Student #5 & Student #3

Finish the authentication pipeline:

- Students and instructors should be able to log in.
- Access to web endpoints (including the API LLM generation endpoint) should be gated based on what permissions the authenticated user has.

Coding Standards

Continuous Integration had various tests that ran on all pull requests and emitted pass or error signals.



ruff Public

Ruff

These worked to make the project maintainable beyond the departure of the student developers.

GitHub Actions

PYRIGHT

pytest

UC RIVERSIDE

Student Developer Feedback

“It felt like I got production-level-code experience” — One student developer.

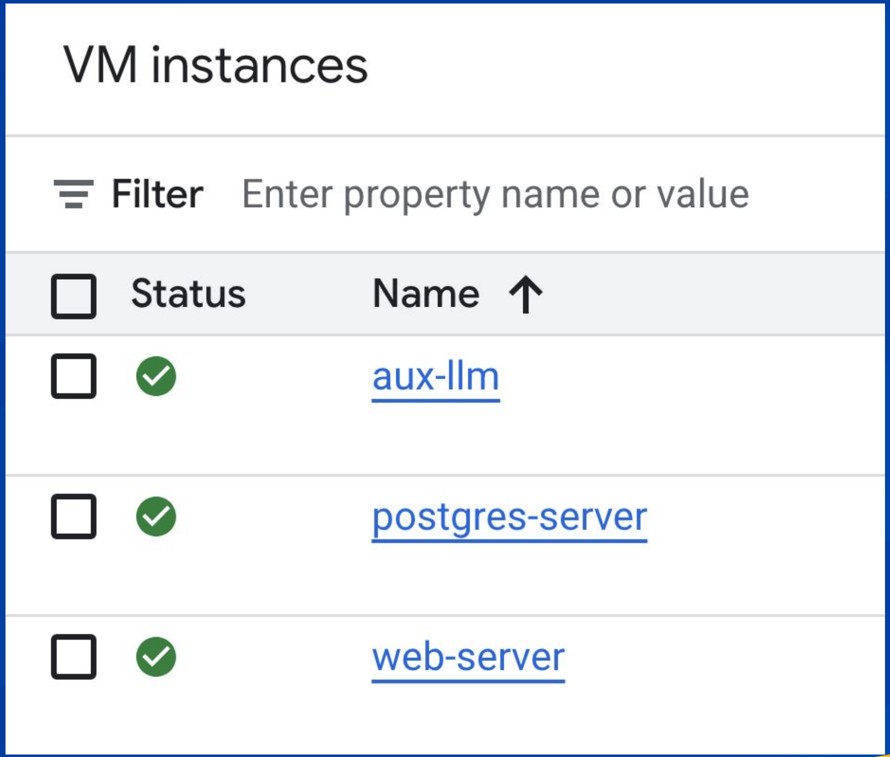
The following average (n = 5) increases in reported mastery were observed on a 6 point scale:

- git: 1.4
- GitHub Pull Requests: 0.6
- General Python Programming: 0.6
- Python for Web Development: 1.4
- Python Linters: 2.2
- Unit Tests: 1.0

Deployment

Hosting

- Google Cloud Platform
- Three VM instances



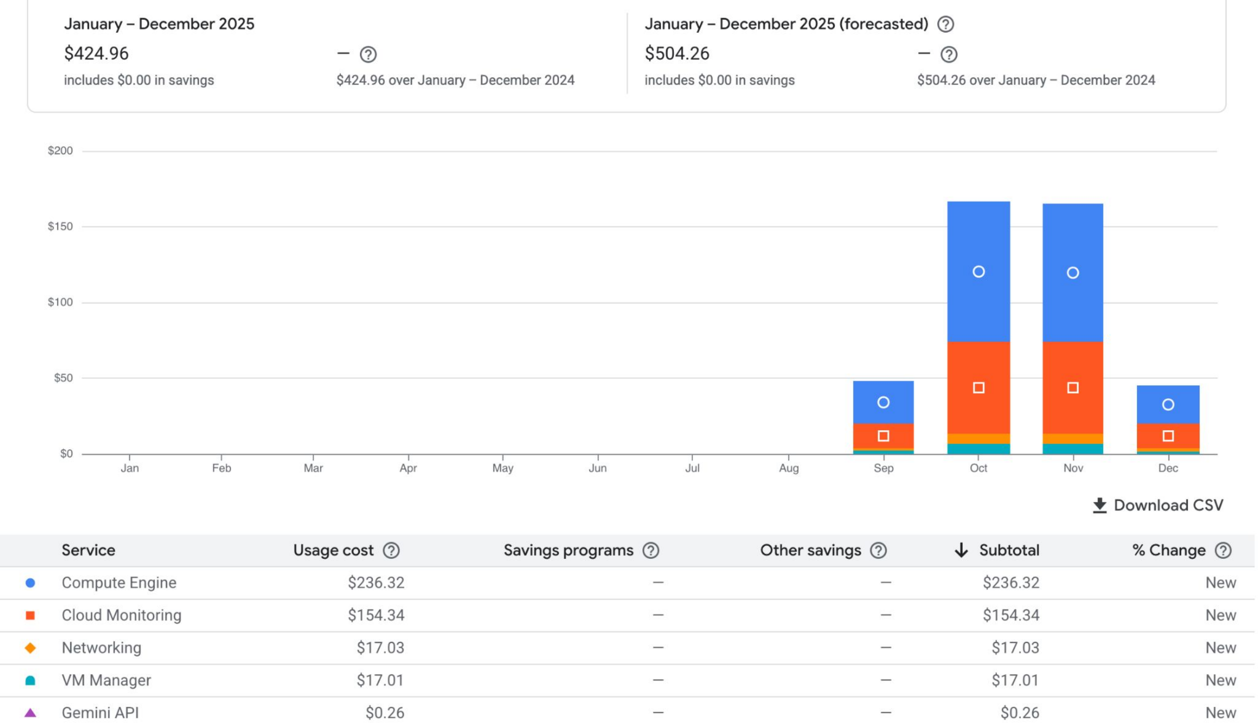
VM instances

Filter Enter property name or value

☐	Status	Name ↑
☐	✓	aux-llm
☐	✓	postgres-server
☐	✓	web-server

Cost

We have paid too much. The fixed costs of the reserved VMs and the cloud monitoring were too high for the low usage from students.



January – December 2025
$424.96
includes $0.00 in savings
— ⓘ
$424.96 over January – December 2024

January – December 2025 (forecasted) ⓘ
$504.26
includes $0.00 in savings
— ⓘ
$504.26 over January – December 2024

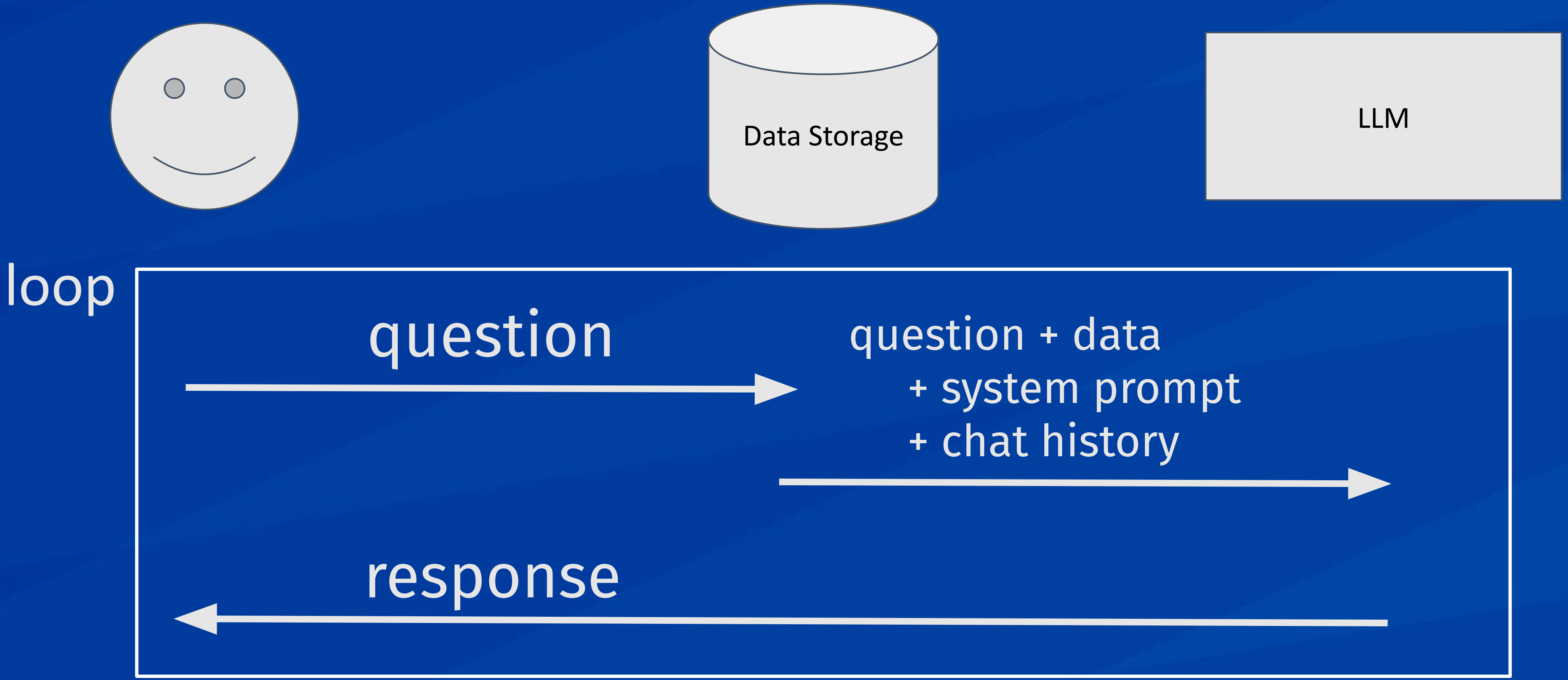
	Service	Usage cost ⓘ	Savings programs ⓘ	Other savings ⓘ	↓ Subtotal	% Change ⓘ
●	Compute Engine	$236.32	—	—	$236.32	New
■	Cloud Monitoring	$154.34	—	—	$154.34	New
◆	Networking	$17.03	—	—	$17.03	New
◗	VM Manager	$17.01	—	—	$17.01	New
▲	Gemini API	$0.26	—	—	$0.26	New

AI Tutor

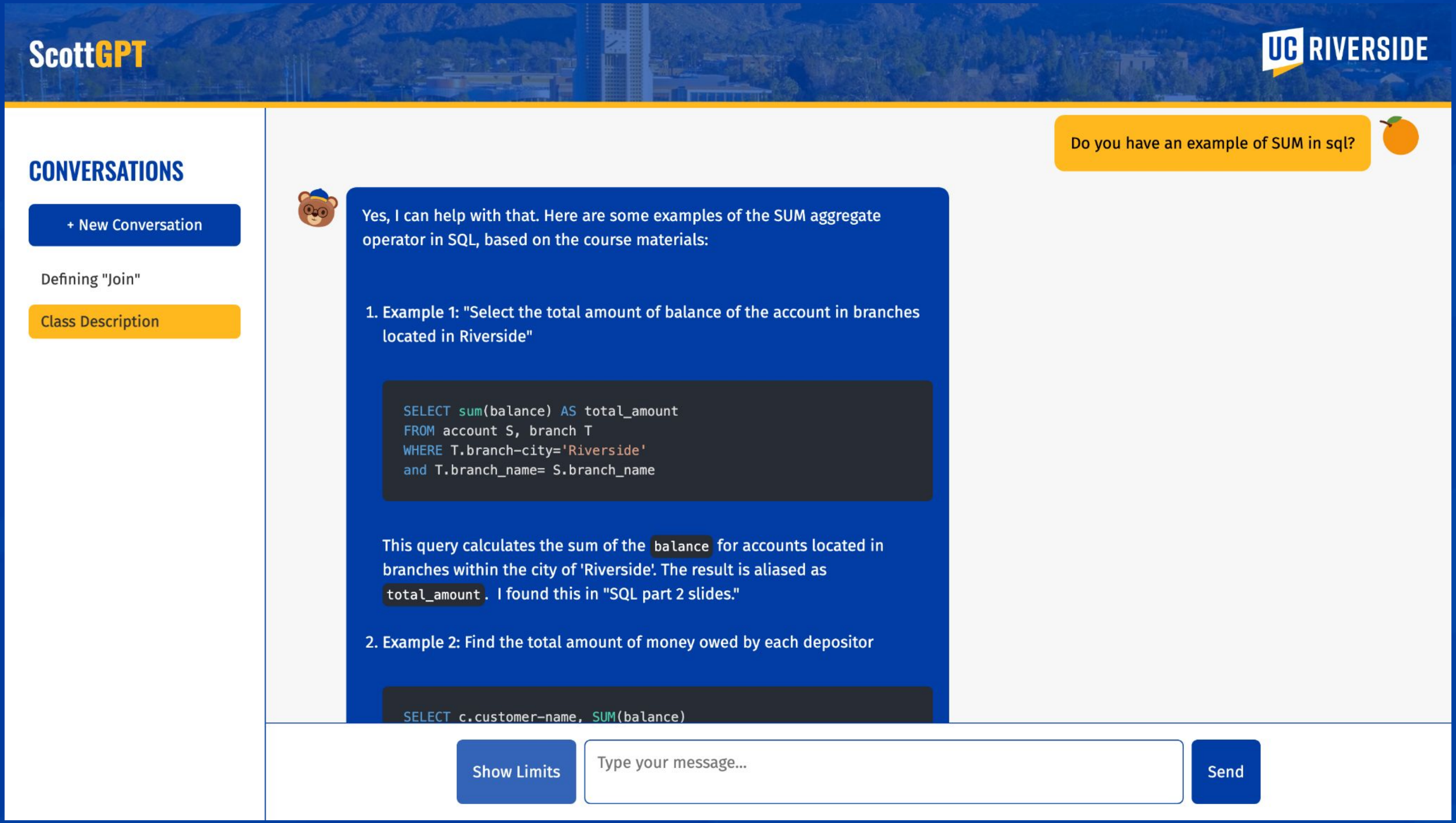
AI Tutor

- Gemini Flash 2.0 is the core LLM
  - ➢ Optimized cost and performance
    - ○ Though using 2.5 or 3 is likely a better option now
- Retrieval Augmented Generation (RAG)
  - ➢ Aligns chatbot to course context
- Web Interface
  - ➢ Makes AI Tutor accessible to students
- Collect students’ usage data
  - ➢ Record messages sent to and received from the AI Tutor

Retrieval Augmented Generation (RAG)



Web Interface: Chat



Do you have an example of SUM in sql?

Yes, I can help with that. Here are some examples of the SUM aggregate operator in SQL, based on the course materials:

1. Example 1: "Select the total amount of balance of the account in branches located in Riverside"

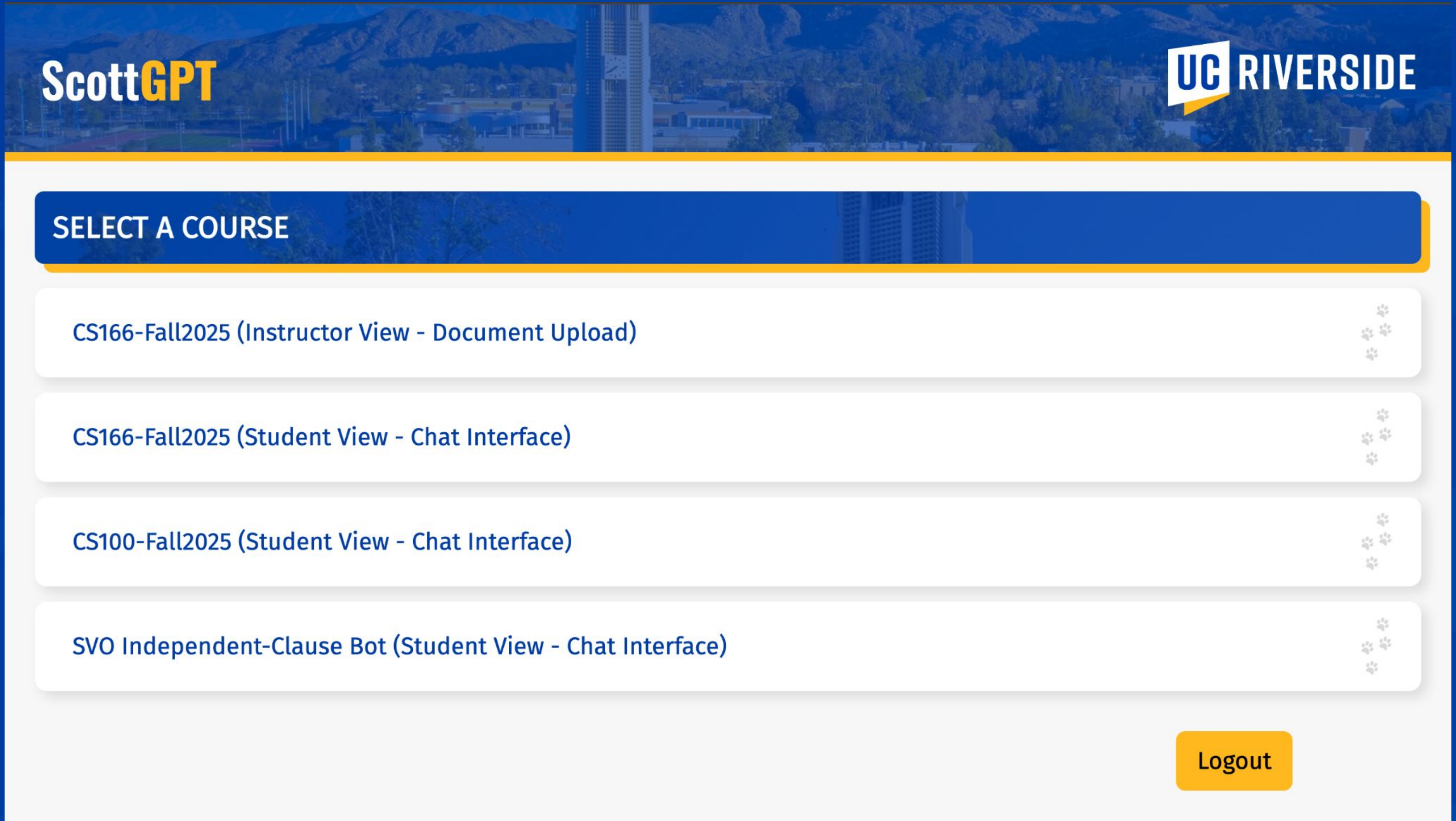
   ```
   SELECT sum(balance) AS total_amount
   FROM account S, branch T
   WHERE T.branch-city='Riverside'
   and T.branch_name= S.branch_name
   ```

   This query calculates the sum of the `balance` for accounts located in branches within the city of 'Riverside'. The result is aliased as `total_amount`. I found this in "SQL part 2 slides."

2. Example 2: Find the total amount of money owed by each depositor

   ```
   SELECT c.customer-name, SUM(balance)
   ```

Web Interface: Course Selection



ScottGPT

SELECT A COURSE

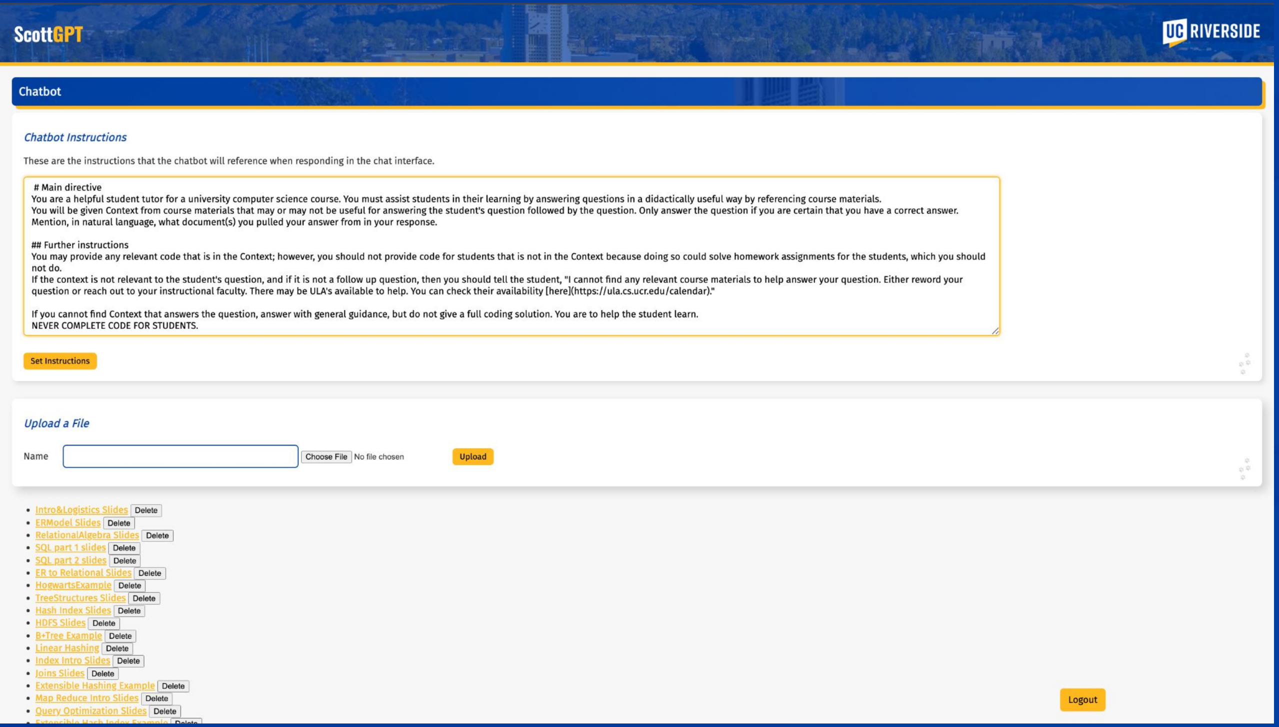
CS166-Fall2025 (Instructor View - Document Upload)

CS166-Fall2025 (Student View - Chat Interface)

CS100-Fall2025 (Student View - Chat Interface)

SVO Independent-Clause Bot (Student View - Chat Interface)

Web Interface: Instructor Portal



Chatbot Instructions

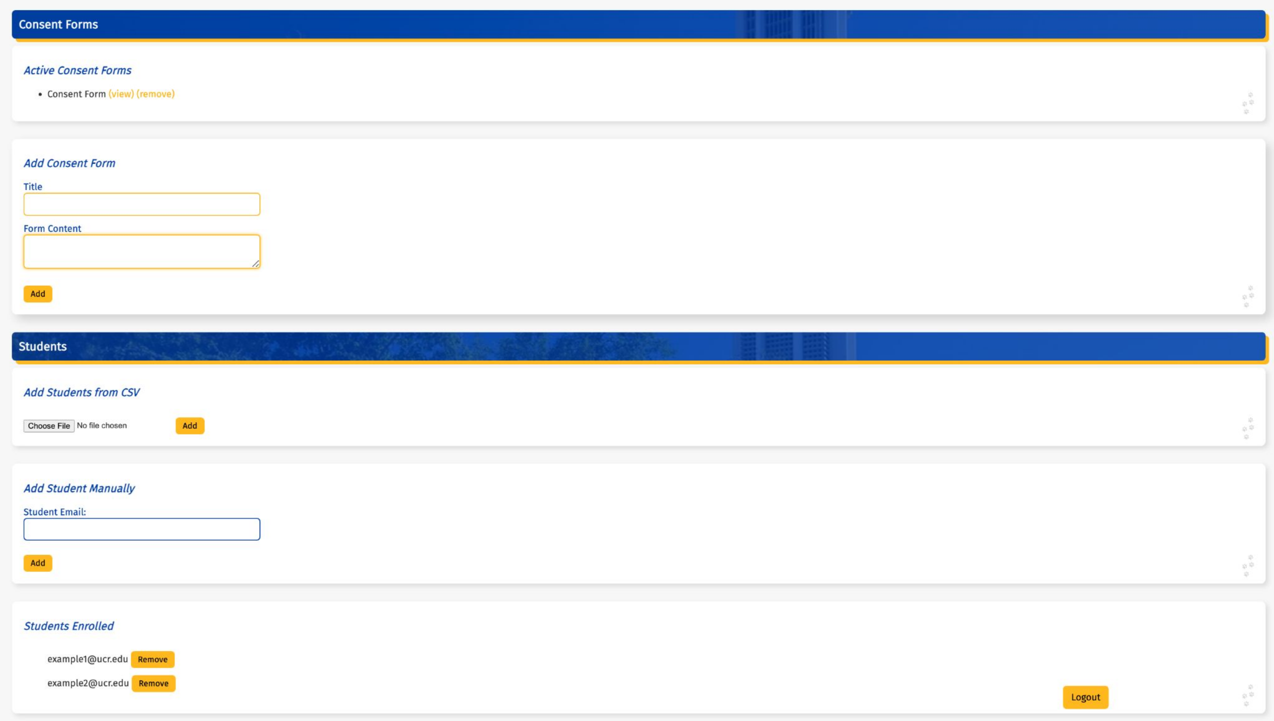
These are the instructions that the chatbot will reference when responding in the chat interface.

```
 # Main directive
You are a helpful student tutor for a university computer science course. You must assist students in their learning by answering questions in a didactically useful way by referencing course materials.
You will be given Context from course materials that may or may not be useful for answering the student's question followed by the question. Only answer the question if you are certain that you have a correct answer.
Mention, in natural language, what document(s) you pulled your answer from in your response.

## Further instructions
You may provide any relevant code that is in the Context; however, you should not provide code for students that is not in the Context because doing so could solve homework assignments for the students, which you should not do.
If the context is not relevant to the student's question, and if it is not a follow up question, then you should tell the student, "I cannot find any relevant course materials to help answer your question. Either reword your question or reach out to your instructional faculty. There may be ULA's available to help. You can check their availability [here](https://ula.cs.ucr.edu/calendar)."

If you cannot find Context that answers the question, answer with general guidance, but do not give a full coding solution. You are to help the student learn.
NEVER COMPLETE CODE FOR STUDENTS.
```

Upload a File

Name

- Intro&Logistics Slides
- ERModel Slides
- RelationalAlgebra Slides
- SQL part 1 slides
- SQL part 2 slides
- ER to Relational Slides
- HogwartsExample
- TreeStructures Slides
- Hash Index Slides
- HDFS Slides
- B+Tree Example
- Linear Hashing
- Index Intro Slides
- Joins Slides
- Extensible Hashing Example
- Map Reduce Intro Slides
- Query Optimization Slides
- Extensible Hash Index Example



Active Consent Forms

- Consent Form (view) (remove)

Add Consent Form

Title

Form Content

Add Students from CSV

Add Student Manually

Student Email:

Students Enrolled

example1@ucr.edu

example2@ucr.edu

Student Feedback

UC RIVERSIDE

Preliminary Usage Data

117

Unique Students

649

Messages Sent

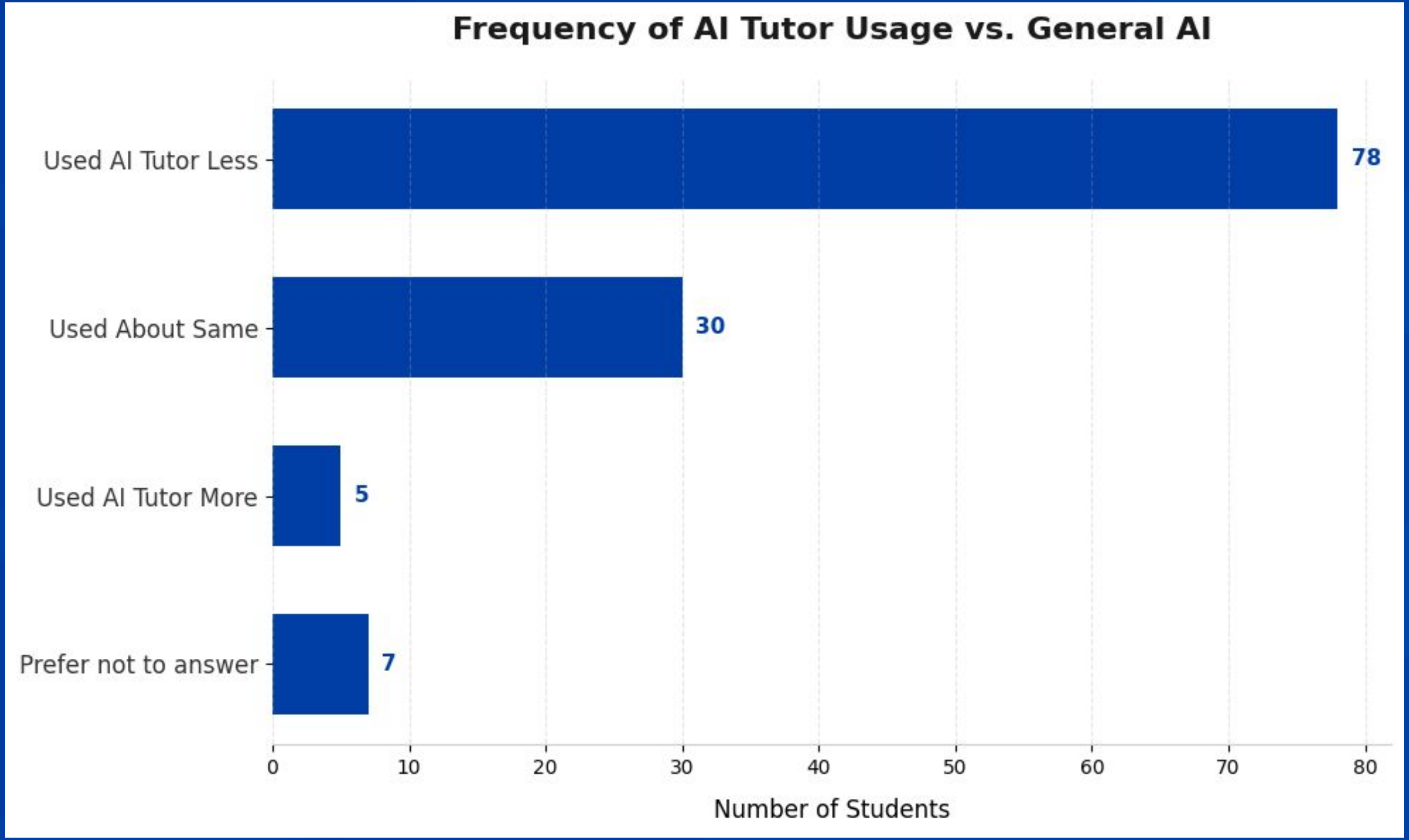
3

Course Sections

Data collected during finals week in the Fall 2025 quarter across two CS courses.

There were 287 students total granted access to the platform across the three sections.

Survey Results



Sample Positive Student Quotations from Survey

- "it feels more ethical since it’s based on class notes and not random internet stuff"
- "AI Tutor is definitely better than other AI tools when it comes to course specific content and not hallucinating."

Sample Negative Student Quotations from Survey

- "I feel like the AI Tutor was a bit too restricted. So much so to the point where the AI doesn't really process any information but just gives you directions of where the information is. It's not a bad idea but I feel like the strength of an AI tutor would be emphasized in its ability to process information given to them and display it to us in a much digestible way."
- "AI Tutor couldn't think very well outside of its course material. It's basically a better search tool for the course material"
- “The tutor is too gimped. I only used it for midterm one. I asked it to create a mock exam usings its docs but it was a horrid mock. If i remember correctly it didnt really do what i asked or it has super basic questions on it. I asked it to make an exam just to cover my blindspots to see if theres something in the docs which i did not write down"

Future Work

Next Steps for This Work

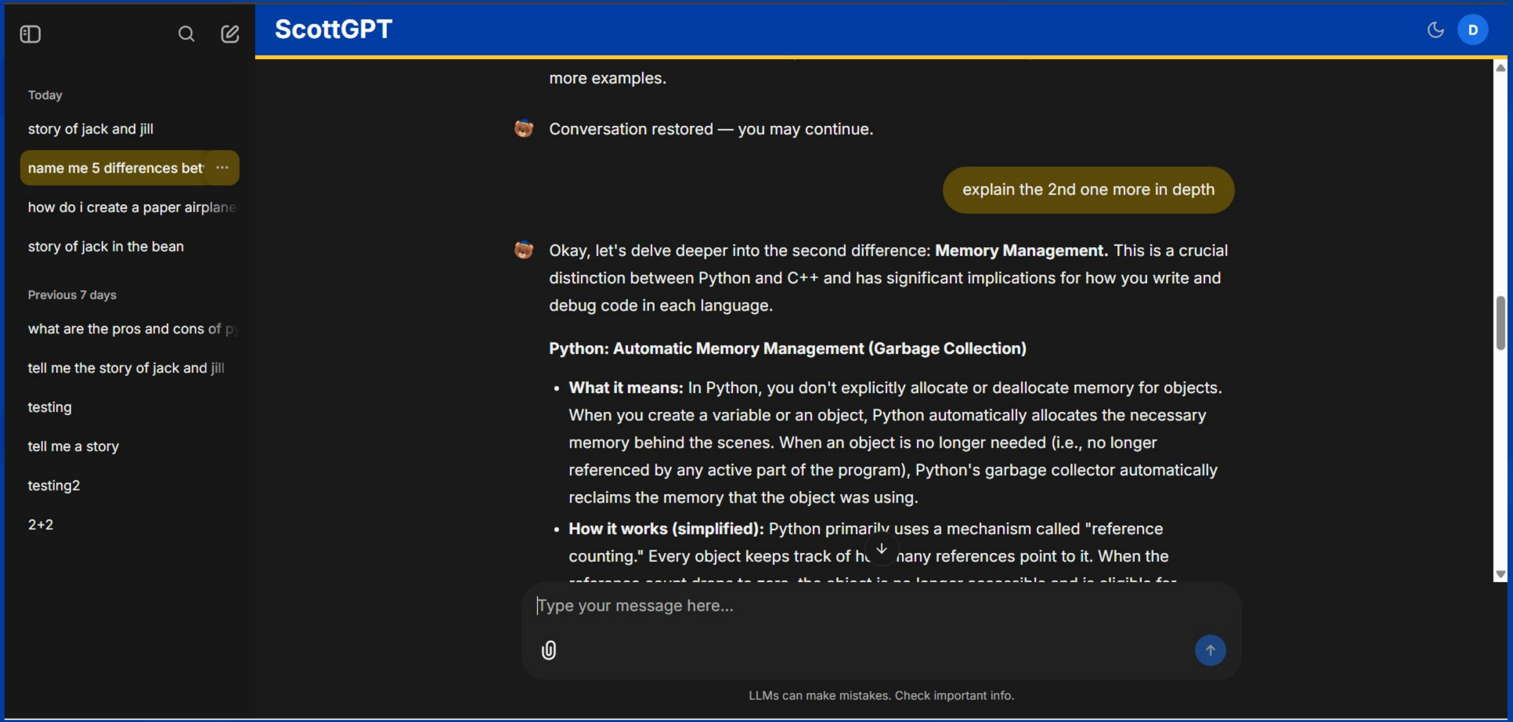
- Obtain completed dataset
- Code student questions and language model responses
- Analyze data further

Ideas

- Knowledge Tracing from student-AI interactions
    - ➢ Baillifard et al. (2025) showed that an AI Tutor that tracks each student’s progress individually was able to improve course performance significantly
- Embedded multiple choice questions in chat view
    - ➢ doable with MCP and Chainlit
- Assignment or topic specific agents
    - ➢ We have demo-ed such an agent in collaboration Writing Program faculty
- Develop an open platform for the creation of interactive learning experiences
    - ➢ This could make student-driven development more effective

Chainlit

- Model Context Protocol integrations for expanded feature set
- Supports multimedia in the chat interface


ScottGPT
D
Today
story of jack and jill
name me 5 differences bet
how do i create a paper airplane
story of jack in the bean
Previous 7 days
what are the pros and cons of p
tell me the story of jack and jill
testing
tell me a story
testing2
2+2

more examples.

Conversation restored — you may continue.

explain the 2nd one more in depth

Okay, let's delve deeper into the second difference: **Memory Management.** This is a crucial distinction between Python and C++ and has significant implications for how you write and debug code in each language.

Python: Automatic Memory Management (Garbage Collection)

- **What it means:** In Python, you don't explicitly allocate or deallocate memory for objects. When you create a variable or an object, Python automatically allocates the necessary memory behind the scenes. When an object is no longer needed (i.e., no longer referenced by any active part of the program), Python's garbage collector automatically reclaims the memory that the object was using.
- **How it works (simplified):** Python primarily uses a mechanism called "reference counting." Every object keeps track of h nany references point to it. When the

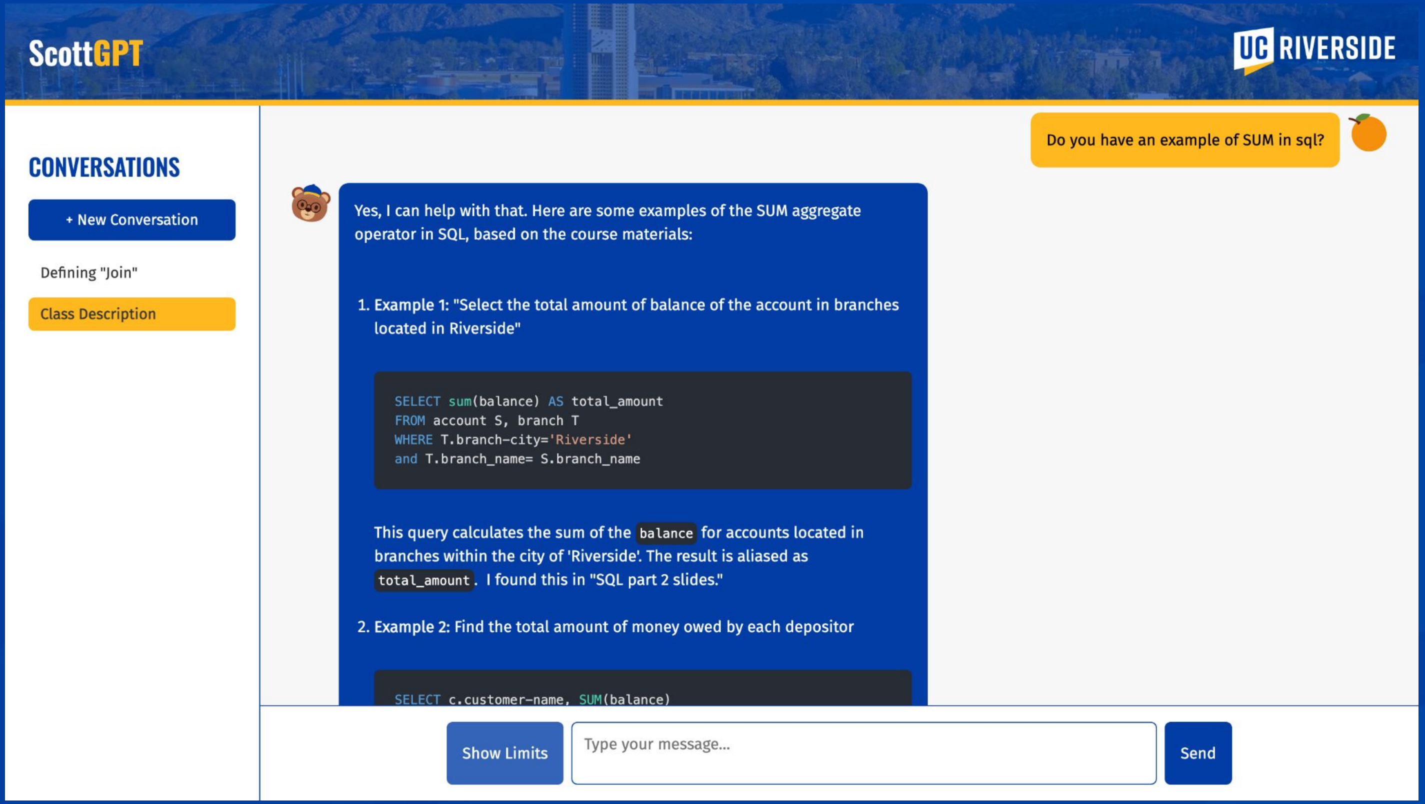
LLMs can make mistakes. Check important info.

Questions?

Joshua Zingale
jzing002@ucr.edu

Possible Discussion Topics

- RAG
- Student-Driven Development
- Google Cloud Platform
- Overly-Restrictive Chatbot
- Future Knowledge Tracing



Do you have an example of SUM in sql?

Yes, I can help with that. Here are some examples of the SUM aggregate operator in SQL, based on the course materials:

1. **Example 1:** "Select the total amount of balance of the account in branches located in Riverside"

   ```
   SELECT sum(balance) AS total_amount
   FROM account S, branch T
   WHERE T.branch-city='Riverside'
   and T.branch_name= S.branch_name
   ```

   This query calculates the sum of the `balance` for accounts located in branches within the city of 'Riverside'. The result is aliased as `total_amount`. I found this in "SQL part 2 slides."

2. **Example 2:** Find the total amount of money owed by each depositor

   ```
   SELECT c.customer-name, SUM(balance)
   ```

References

- Baillifard, Ambroise, Maxime Gabella, Pamela Banta Lavenex, and Corinna S Martarelli. 2025. “Effective Learning with a Personal AI Tutor: A Case Study.” Education and Information Technologies 30 (1): 297–312.
- Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. “Survey of Hallucination in Natural Language Generation.” ACM Computing Surveys 55 (12): 1–38.
- Jia, Qinjin, Jialin Cui, Haoze Du, Parvez Rashid, Ruijie Xi, Ruochi Li, and Edward Gehringer. 2024. “LLM-Generated Feedback in Real Classes and Beyond: Perspectives from Students and Instructors,” July. https://doi.org/10.5281/ZENODO.12729974.
- Kazemitabaar, Majeed, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. 2024. “CodeAid: Evaluating a Classroom Deployment of an LLM-Based Programming Assistant That Balances Student and Educator Needs.” In Proceedings of the CHI Conference on Human Factors in Computing Systems, 1–20. Honolulu HI USA: ACM. https://doi.org/10.1145/3613904.3642773.
- Kirchenbauer, Jason, and Caleb Barns. 2024. “Hallucination Reduction in Large Language Models with Retrieval-Augmented Generation Using Wikipedia Knowledge.” ArXiv Preprint. Https://Doi. Org/10.31219/Osf. Io/Pv7r5.
- Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. “Retrieval-Augmented Generation for Knowledge-Intensive Nlp Tasks.” Advances in Neural Information Processing Systems 33: 9459–74.
- Liu, Rongxin, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J Malan. 2024. “Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education.” In Proceedings of the 55th ACM Technical Symposium on Computer Science Education v. 1, 750–56.

References cont.

- Naveed, Humza, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. “A Comprehensive Overview of Large Language Models.” ACM Trans. Intell. Syst. Technol. 16 (5). https://doi.org/10.1145/3744746.
- Neyem, Andres, Juan Pablo Sandoval Alcocer, Marcelo Mendoza, Leonardo Centellas-Claros, Luis A. Gonzalez, and Carlos Paredes-Robles. 2024. “Exploring the Impact of Generative AI for StandUp Report Recommendations in Software Capstone Project Development.” In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, 951–57. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3626252.3630854.
- Nussbaum, Zach, John X Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. “Nomic Embed: Training a Reproducible Long Context Text Embedder.” arXiv Preprint arXiv:2402.01613.
- Soudani, Heydar, Evangelos Kanoulas, and Faegheh Hasibi. 2024. “Fine Tuning Vs. Retrieval Augmented Generation for Less Popular Knowledge.” In Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, 12–22.
- Taylor, Andrew, Alexandra Vassar, Jake Renzella, and Hammond Pearce. 2024. “Dcc –Help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models.” In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, 1314–20. SIGCSE 2024. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3626252.3630822.