Backend Discussion

CS 485/698: AI-Assisted SE

In the News

CHECK YOUR WORK

After outages, Amazon to make senior engineers sign off on AI-assisted changes

AWS has suffered at least two incidents linked to the use of AI coding assistants.

Today’s Agenda

- Team meeting (~15 minutes)
  - Sprint retro for P4:
    - What is going well? What are you struggling with?
    - Did you over- or under-estimate any tasks?
  - Be ready to show us that you’ve made progress on P4
    - max 1 minute demo time
- A4 discussion (rest of class)

A4 discussion

- On the next slide, there is a list of groups. Sit with your group near the big number
- Each group has ***at most one*** person who picked each reflection question
- We'll discuss the questions in order:
  - Discuss with your group for ~5 minutes
  - Share anything particularly interesting with the whole class
    - **New today**: focus this part on sharing anything that will help with P4

A4 discussion: groups

Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Dhyani Soni	Declan Blanchard	Thomas Kolb	Balaji Shashipreeth Racherla	Marcus Hilario	Tirell Spence
James Mullins	Fardeen Iqbal	Salma Ghazi	Jossie Zamora	Nafisa Ahmed	Jonathan Martinez
Luke Hill	Ashraf Aldekaim	James Marciano	Elvis Valcarcel	Allen Cabrera	Isabel Patrisso
Avanish Kulkarni	Saanvi Elaty	Eric Perez	Engy Masoud	Krishi Shah	Aryan Modi
Swechcha Ambati	Xun Song	Zhirong Zhang	Mark Youssef	Justin Carreno	Haroon Aftab
Brandon Howe	Vishesh Raju	Aiden Barrera	Victor Jimenez	Paulo Bellame	Alexander Tochtchev
Roaa Elsayed	Youssef Masoud				

Question 1

Is LLM-generated backend code defensive enough, too defensive, or just defensive enough? ("Defensive" here refers to the amount of error handling logic. In other words, the question is asking about how much error-handling code LLMs generate.)

Question 2

Do you feel more or less confident in backend code compared to frontend code generated by LLMs? How do you test what you can't see?

Question 3

Is LLM-generated code secure? Does LLM-generated code need to be reviewed more or less thoroughly than human-written code? Are there special steps that we should take to make sure that LLM-generated code is secure?

Question 4

Are there any backend components that should always be handled by humans? E.g., should LLMs avoid directly modifying the database schema, or should API specifications be provided explicitly? In other words, what do you think is the ideal balance between human and LLM in backend development?

Question 5

Do LLMs make good architectural decisions? Do they use design patterns appropriately?

Question 6

Is LLM-generated code usually scalable to many users, or do you need to do something special to make sure that it will scale? How can you tell if LLM-generated code will create performance bottlenecks?

Question 7

How maintainable is LLM-written code? Is LLM-generated code overengineered more or less often than human-written code? Does it contain more or less technical debt?

Wrapup and Reminders

- P4 now due at the **end** of spring break (March 22, AoE)