Individual Instructor Report Fall 2023 Version A for C S 373 - SOFTWARE ENGINEERING (52985) (Glenn Downing)

Project Title: **Course Evaluations Fall 2023**

Courses Audience: **52**
Responses Received: **49**
Response Ratio: **94.2%**

Report Comments

Guide to the Interpretation of Course Evaluations at UT Austin
The goal of course evaluation process at UT Austin is to drive teaching excellence and to support continuous improvement in teaching and learning experiences. The two sets of scales used for core evaluation questions and the associated weights are:

Strongly Agree (5)
Agree (4)
Neutral (3)
Disagree (2)
Strongly Disagree (1)

Excellent (5)
Very Good (4)
Satisfactory (3)
Unsatisfactory (2)
Very Unsatisfactory (1)

The Mean is calculated by adding all of the weights for a single question and dividing by the number of respondents. The course workload question is not averaged.

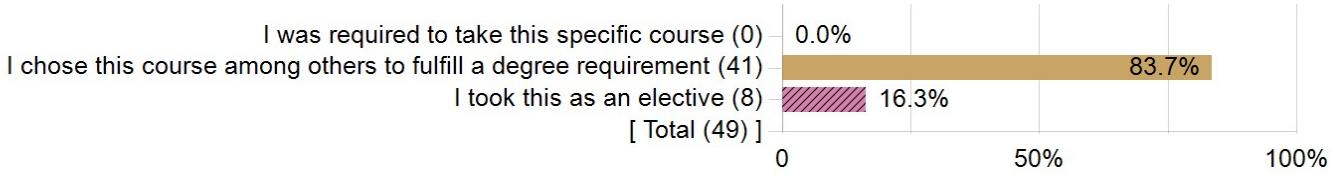
The number of students (e.g. respondents) marking each option is reported for each of the items. These frequency distributions provide information about the level of student ratings and the spread and shape of the class distribution of responses. The distributions thus provide a picture of student perception of a course.

Course evaluations provide snapshots of student perspectives on their course-level learning experiences. Most experts on teaching evaluation advise that no individual method gives the complete picture of an instructor's teaching effectiveness; multiple and diverse measures, on multiple occasions, are advised to give a full picture of the teaching effectiveness of a particular instructor. Moreover, other factors, such as size of class, level of the class, and content of the course, can cause small variations in the ratings. Therefore, student perspectives for a particular instructor or course should be interpreted as a snapshot, and not as providing complete information on the teaching effectiveness of that instructor.

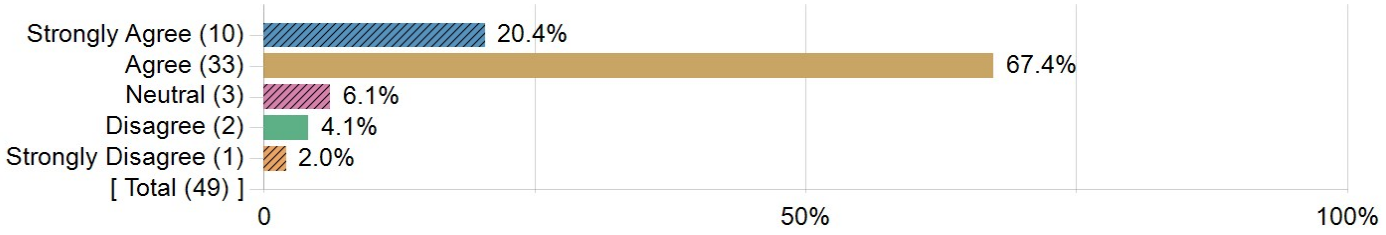
Creation Date: **Tuesday, December 19, 2023**

Course Questions

Why did you take this course?

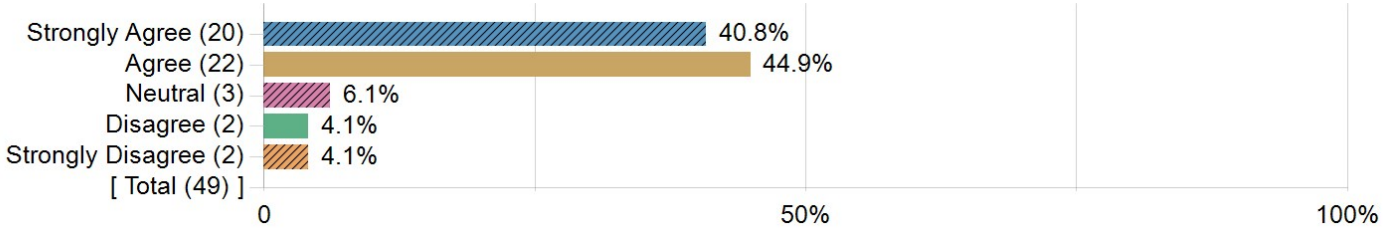


During this course, I gained a deeper understanding of the subject matter.



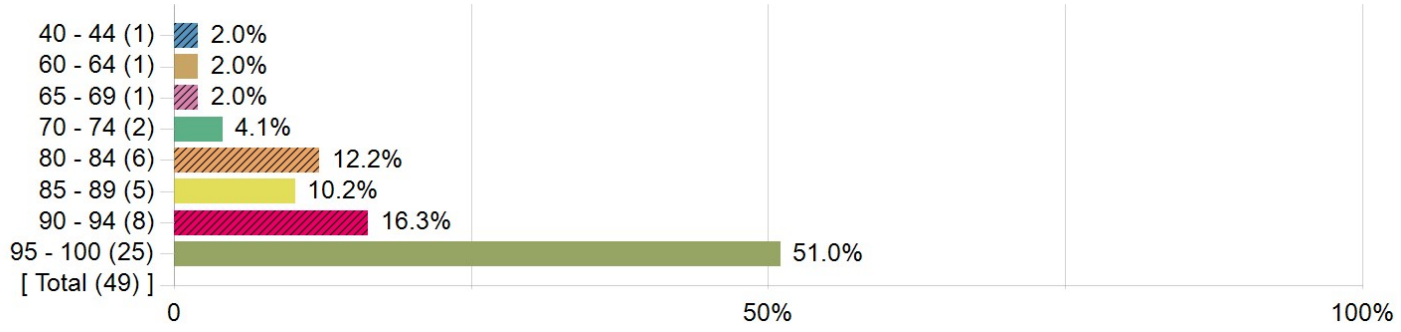
Statistics	Value
Mean	4.00

The course was well organized.



Statistics	Value
Mean	4.14

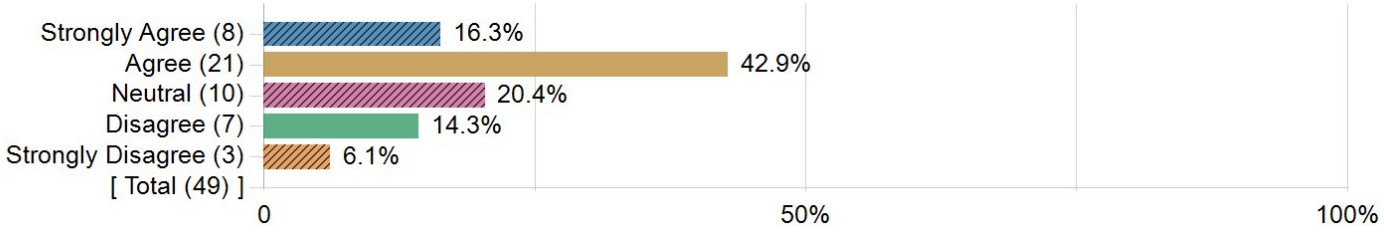
Overall, approximately what percentage of the course meetings did you attend or complete (online, in person, or asynchronously)?



Statistics	Value
Mean	88.90

The course assignments improved my ability to express my ideas in writing. (Flag Question)

The course assignments improved my ability to express my ideas in writing.



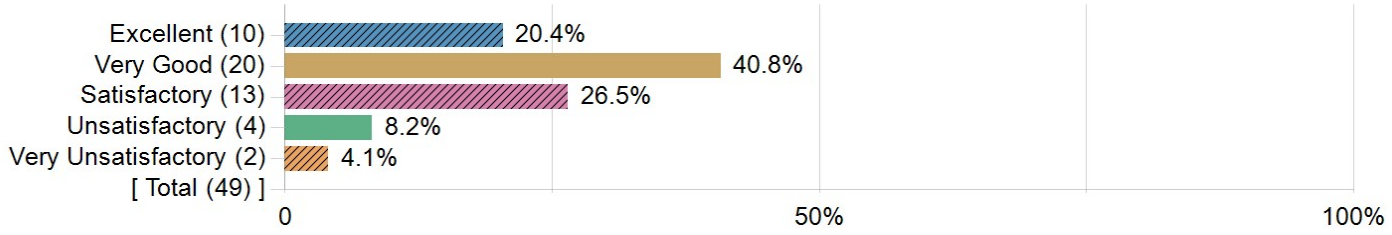
Statistics	Value
Mean	3.49

Instructor Questions

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	44.9%	42.9%	4.1%	4.1%	4.1%	49	4.20
The instructor fostered an inclusive learning environment.	49.0%	36.7%	6.1%	4.1%	4.1%	49	4.22
The instructor effectively explained the concepts and subject matter in this course.	40.8%	36.7%	12.2%	4.1%	6.1%	49	4.02
The instructional techniques kept me engaged in learning.	38.8%	38.8%	16.3%	2.0%	4.1%	49	4.06
The instructor checked for student understanding of the concepts presented in the course.	49.0%	38.8%	6.1%	0.0%	6.1%	49	4.24

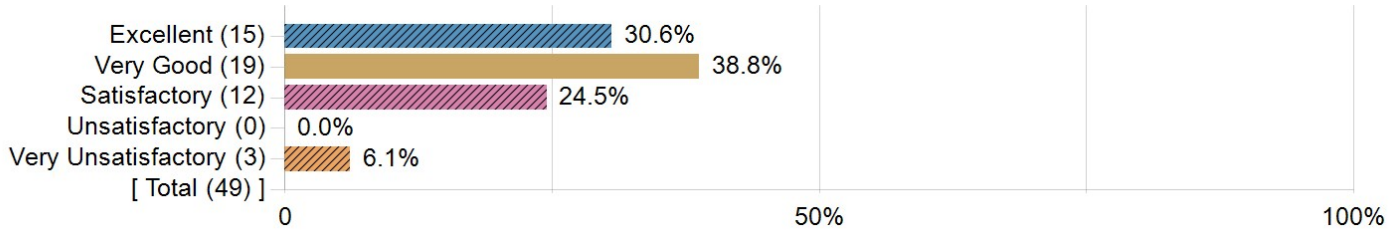
Overall Questions

Overall, this course was



Statistics	Value
Mean	3.65

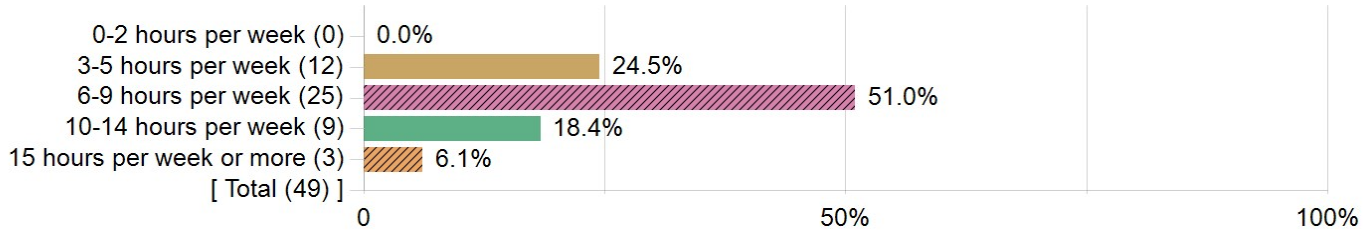
Overall, this instructor was



Statistics	Value
Mean	3.88

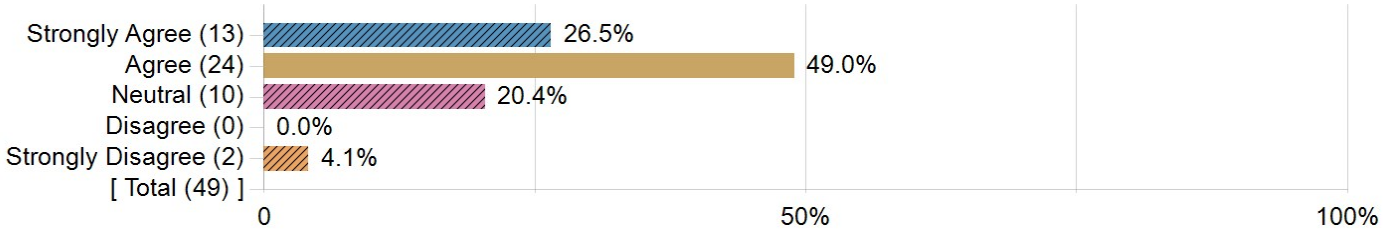
College, School, or Unit Questions

On average, approximately how many hours per week did you spend working outside of the course? Include time on homework, reading, reviewing, papers, projects, etc.



Statistics	Value
Invited Count	52
Response Count	49
Response Ratio	94.2%
Mean	3.06
Median	3.00
Semi-Interquartile Range	0.50
Mode	3
50th Percentile	3.00
Sum Total	0.00
Standard Deviation	0.83
Population Standard Deviation	0.82
Standard Error (base on SD)	0.12
Standard Error (base on PSD)	0.12
Aggregate Frequency 1	0.0%

The course format (online, hybrid, face-to-face) helped me to learn.



Statistics	Value
Mean	3.94

Comment Questions

Identify aspects of the course that were the most effective in helping your learning.

Comments
The lectures enhanced my learning of different concepts related to Python, SQL, and refactoring.
The projects were the most effective at teaching.
The lectures were insightful
I liked the deep dive into python and sql
the projects forced us to learn aspects of software engineering on our own.
The projects enabled me to learn a bit because we had to do tasks that I had never done before.
I really liked the idea of cold calling, it always made me pay more attention to the course material.
cold calls
Required attendance with quizzes helped to ensure learning
The daily quizzes made sure I was up to date with the material
I enjoy Prof. Downing's cold call approach that keep students on their toes in case they get called on. With the cold calling approach you also get to understand problems better, and don't need to be afraid of asking the "stupid" questions.
The exercises were very helpful, as well as the projects.
I think I learned a lot in class, and the quizzes helped me further strengthen my understanding of the concepts and the ability to remember what we learned in the previous lecture. I also learned a lot from the web dev project. The exercises were also pretty helpful for in–depth understanding.
The IDB project was an effective way of teaching software engineering and web development tools and procedures.
The Python and SQL lectures were helpful in understanding the languages at a deeper level. It was useful to use what was learned when programming outside of the course.
the group project in general was a huge learning experience and I thought that the weekly check in meetings were helpful since we were able to get feedback.
Having weekly TA feedback on our projects was super helpful in guiding us to reaching our goals.
The Socratic questioning method during the lecture kept me engaged and thinking about how I would answer the questions my peers were answering. The project allowed me to explore very relevant tools and languages in web development as well as refine my team leadership skills, which I found to be both intriguing and challenging.
I thought the projects were great in regards to helping us dive head first into a SWE–esque environment.
The Socratic style of teaching meant that I had to constantly pay attention in class in order to be ready to answer the professor's questions. That meant I internalized a lot more of the information presented. Also, the daily quizzes about the previous lecture's content also reinforced the information.
Doing the projects and learning about things on my own or with my team members was the most helpful.
The group work and structure of the teams was very close to a real experience of working in the industry, which I thought was a great learning experience.
Lectures were very well organized and were easy to follow along as our instructor made everyone participate.
The conversation style lectures.
Ed Discussion had helpful tips and resources for working on the project. Projects from past semesters were also helpful to serve as a reference.
The way the professor lectured with talking to specific students.
The project and nothing else was relevant to my learning. I found classes extremely unrelated to what we needed to do and was instead 'cool things you can do in python' that are extremely obtuse and unneeded.
The projects, while messy at first and requiring one to do their own learning, fostered many important technical and team skills that no doubt will be important outside of class.
I liked cold calling
Hands–On Learning
CS 373: Software Engineering was exceptionally well–structured, fostering an effective learning environment throughout the semester. The course began with clear and concise objectives, providing a roadmap for the topics to be covered. The progression of topics was logical, building upon foundational concepts before delving into more advanced material. What set this course apart was its emphasis on project–based learning and practical coding assignments. These hands–on experiences allowed for the

Comments
direct application of theoretical knowledge, reinforcing concepts and enhancing problem–solving skills. Collaborative learning opportunities through group projects facilitated teamwork, mirroring real–world software development scenarios. The lectures were engaging, well–prepared, and augmented with real–world examples, making complex concepts more accessible. Instructors were not only responsive and supportive but also provided regular and constructive feedback on assignments, aiding in continuous improvement. The integration of industry–relevant tools and practices, coupled with real–world case studies, ensured that the course content remained up–to–date and aligned with the ever–evolving field of software engineering. Overall, the course succeeded in providing a holistic understanding of the software development life cycle and preparing students for the challenges of the industry.
I appreciate the lectures because Dr. Downing provides detailed explanations for each topic, which greatly helps me in developing a deeper understanding of the materials.
The lectures were extremely well taught, and that helped my learning a lot.
The project was kind of like a deep dive into content, which I learned a lot about but I felt like the course content could align with project more.
TA guides were extremely helpful
The projects were okay.
The IDB project was an excellent way to provide students with hands–on experience.
I thought the in class quizzes and cold calling helped ensure that I really learned the material and stayed engaged during class.
I think the project itself taught me the most. Also the TAs are sooooo good.
Our professor employed cold–calling during lectures which was personally beneficial in my learning because it kept me on my toes. The live coding we did was also pretty helpful in keeping me engaged and on task.
It seemed like the professor put a lot of time and effort into planning the class material and structure.
The TAs.
Very clear instructions on grading and expectations for assignments

Identify the aspect of the course that you found most challenging, why you found it was challenging, and suggest one thing that could be done to help future students meet that challenge more effectively.

Comments
The most challenging aspect was the team project, because we had to teach ourselves how to use the various tools to complete the project. In the future, I suggest that students should try and resolve any communication issues they may have with their team as early as possible, so the rest of the project will go smoothly.
Also the projects, but I felt that the difficulty was reasonable.
The project deadlines were challenging, as were some of the quizzes. More time on the quizzes would be great.
A lot of the material, especially for IDB, is self–taught. I liked the deep dive into python and sql, but would have preferred we cover more topics during lecture (js, react, APIs, deployment, etc)
The projects were the most challenging part of the course
I found the projects to be most challenging because there was no instruction or guidance given. We weren't even taught about how to do some of the foundational techniques needed for the project. I feel like class time was useless for majority of the class and we really should have spent the time learning how to do more applicable tasks, like things related to our website.
I think the biggest challenge for me and my group was to get started on the project early that's why we always kind of struggled close to the deadline. Nothing on the course or professor though.
quizzes but i don't really have any suggestions for changes
Required attendance also meant less ability to miss class if something were to come up. This wouldn't be a problem if you passed all the quizzes, but thats not a luxury for everyone. I would've liked a tad bit more quiz drops. The SQL exercises weren't easy to do because of the need to comment/uncomment in order to see the entire output. The time crunch and many questions on the exercises combined with the extra need to comment/uncomment/compare–to–solution–output really made those exercises very stressful.
The projects were a bad experience for me because my group was not helpful. We only had 4 people at the start and out of the 4 only me and one other person did all the work and we still might get a B in the class even though we did more than our fair share of work.
I think the daily quizzes can be challenging, if you are not staying up to speed on lectures. The grading scheme also makes it so you have to be doing well in all areas of the course, as your grade in the course is the lowest of any of the categories.

Comments
The quizzes were the most challenging part, but I think they were a good difficulty (i.e. they weren't too hard). It is helpful to review your notes from last class before taking the quiz. The papers were engaging and useful; I more about software engineering from them than anything else in the class. The blog posts felt like total busy work. I don't think they add anything to the class. Since there's nowhere else to put this, I'll put it here: Professor Downing is a really good teacher and he explains the concepts that he teaches very well. However, CS373 was not a class about software engineering. Only in the last few weeks (and in the papers) did we talk about SWE concepts. I think the course should be more about SWE topics (or else not be named "Software Engineering").
I found some of the quizzes to be a bit challenging because sometimes they tested certain parts of the lecture that I felt was kind of glossed over. I think it would be nice if the quizzes can really only focus on the important parts of the lecture.
The daily quizzes and the grading scheme of the course worked against my ability to perform at my best for the course. The specifications grading approach in this class is something I would like to see change in future versions of this course.
I found the project most challenging because initially my group members were not very active and I had to take on a lot of the work to get the project completed. I understand the premise of the class is to self learn the topics for the project but it would also be nice if we could get an introduction to those topics in lectures as well.
The group project was challenging in general since communication was not consistent but I think we got better toward the end of the semester once we switched to Slack.
The daily quizzes were very difficult, especially since they were only 2 minutes. I think the point of learning is to be able to reason your way to an answer rather than memorizing the answer, but since the time for quizzes is so short and strict, you don't get that opportunity and are instead rewarded based on memorization rather than true learning.
The daily quizzes could sometimes be tricky to master given the time constraint. Although the grading system was very flexible with quizzes, I still found them to be somewhat stressful. For the most part, as long as you review the lecture material (especially the online lecture recordings) before the next quiz, you should be able to demonstrate mastery on enough of them to earn an A.
I found that learning concepts such as AWS or new tools such as Alchemy could be quite cumbersome at times.
The first part of the semester is spent learning the quirks of Python, a language that we barely even use for the main assignments of the course. I would prefer if that part of the course were dedicated to Javascript instead since regardless of the framework that you use for the website, you will have to write at least some Javascript. That would more nicely lead into the section on SQL, which is definitely used on the assignments.
Having to learn every single aspect of the technologies we used was difficult. Personally, I struggled a lot with Gitlab CI at the beginning of the semester. If Gitlab's CI specifically was talked about in lecture (like Docker was), that would have saved me a lot of frustration.
Time management, the deadlines were a bit tight at times.
The main challenge was the disconnect between class content and assignments. While I thought it was an important part of the course to allow students to discover new tools/languages in their own time, it felt strange that (almost) everything learned in class had nothing to do with our assignments. I originally envisioned this class as learning about software design in lectures and applying that to a project, but the reality is that the class is a weird advanced python tutorial/class while any design concepts are relegated to readings that are never discussed in class, at least until the end of the semester.
Quizzes
Hard to find times to meet with such a large random group.
I suggest giving more time for the quizzes since 2 minutes isn't enough time, and making it open notes. Also, it would be nice to cover concepts related to the project in class rather than have to learn everything on our own. I would expect to cover some web dev programming languages like Javascript, CSS, or HTML in a software engineering class where we're expected to make a website.
Giving us very little help on the website itself. It would be nice if he gave us a few more tutorials here and there like the docker one. The grading system is also kind of dumb since it encourages mediocrity. If someone does bad in one category and will get a B because of that category, then why would they complete all the assignments required to get an A in another category, when they can just do enough to get a B since they are getting one anyways.
I found it challenging to deal with the quizzes as instead of testing your understanding of already unrelated content, they were about the most niche of scenarios instead of the real main content of the lecture.
The projects were difficult, especially with AWS difficulties in a certain project phase, and were easy to get lost in. The quizzes before lecture were also a bit unforgiving.
I found some of the technical aspects of the quizzes challenging
Website creation from scratch

Comments
The most challenging aspect of CS 373: Software Engineering was the project–based learning approach that leaned heavily on self–directed learning. While the course content itself was comprehensive and well–structured, the projects required a significant amount of independent exploration and application of concepts not explicitly covered in class. This proved challenging as it demanded a high level of self–motivation and a proactive approach to seek out resources beyond the provided materials. To address this challenge more effectively for future students, the course could incorporate structured guidance or tutorials that align with the specific requirements of the projects. Additionally, offering regular check–ins or workshops where students can discuss challenges and receive guidance on their individual projects could provide a balance between autonomy and the support needed for successful completion. This way, students would still benefit from the hands–on, project–based learning approach while having more targeted resources to aid their self–directed efforts.
I find it challenging to work on my group projects due to some team members' lack of responsibility that impacted the progress of the whole team. Unfortunately, we cannot choose our own groups, which makes it difficult to address this issue directly. One suggestion I have to help other students overcome this challenge more effectively is to encourage them to prepare in advance for both the front–end and back–end aspects before taking this course. This way, they will have a solid foundation and be better equipped to contribute effectively to group projects
I think the grading system is unnecessarily complicated. It takes a long time to understand how it works. I also think the way cold call are done were unnecessary. He focused on one student for too long and did not let them go even if they clearly had no idea about the topic at hand. It was hard to watch at times.
I thought the projects were challenging. One thing that might help in the future is to give more resources on how to do some of the things required from the projects since the lectures were so different from what the projects were.
It was hard to keep track of course content and project knowledge to be able to do well on both the quizzes and the projects
Lack of explanation for projects and hard daily quizzes, fast pace of projects
The group design was poorly implemented: if one or more of your group members dropped, you were forced to do more work than you should have. Furthermore, you had no control over the group you were assigned, making your experience in this class luck–based or incredibly time–consuming. In the future, wait until after the add/drop period to assign groups to avoid the imbalanced workloads or let students choose their own groups to make it not luck–based. Besides the groups, the professor was rather rude towards students. Even if some remarks were meant to be jokes, calling students lazy or indirectly calling them stupid should not be taken lightly. Cold calling undoubtedly kept students engaged, and while it slowed down the lecture, it was mostly fine. The lecture ordering seemed to be less than ideal: after we had already used SQL for a while on the projects, the professor started to teach the SQL basics. This order made us learn SQL on our own and then wasted our time by repeating what we already had to teach ourselves. Overall, the professor and the class were not respectful of students' time or the students themselves.
The IDB project was the most challenging part of this course. Though it was an effective learning method, the instructions were vague and none of the skills necessary were taught in class. As a result, students were mostly on their own. Providing better project resources and covering project tools in class would make the learning experience easier and more effective.
I didn't like the lack of guidance for the project. Since the material we learned in class was unrelated, it felt like we had to figure a lot of stuff out on our own. In particular, AWS was very confusing for a lot of us, so I think an extra tutorial for that in the future would be useful.
I think the daily attendance quizzes were grueling. Also the divorce between class content and project content is a little awkward.
I believe some of the quizzes had too short of a time limit for their difficulty. There was definitely a large disparity in the overall difficulty of all the quizzes, some were super easy, and some pretty difficult.
The projects had very specific requirements that kind of hinder room for more creative design, but it's understandable why it's structured that way from a grading perspective.
The lectures were completely unrelated to the topics that we needed for the project. The way that the professor asked questions during class were condescending and annoying without really furthering the students' understanding. We were put in random groups which made it so there was only one or two people doing the main contribution for the project. It would be much better if we got to choose our own groups going into this.
Opportunities to improve quiz grades