

Project Title: **Course Evaluations Fall 2023**

Courses Audience: **59**
Responses Received: **54**
Response Ratio: **91.5%**

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)

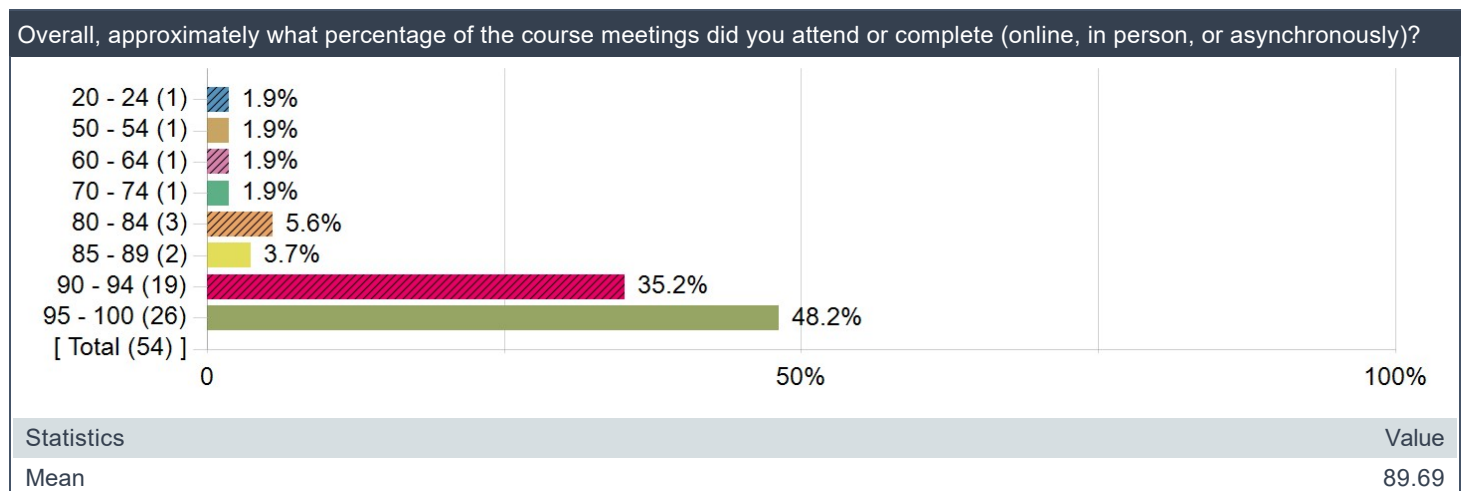
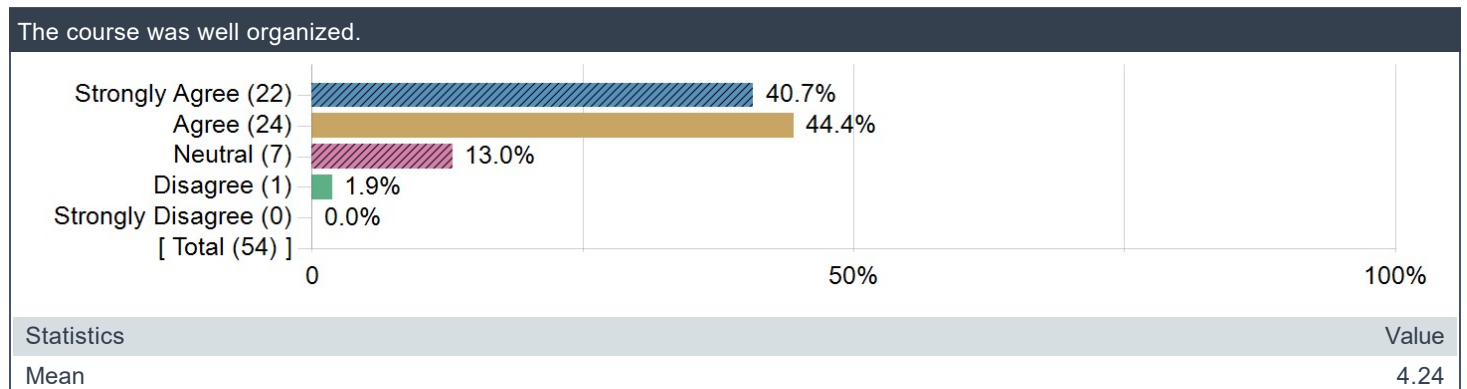
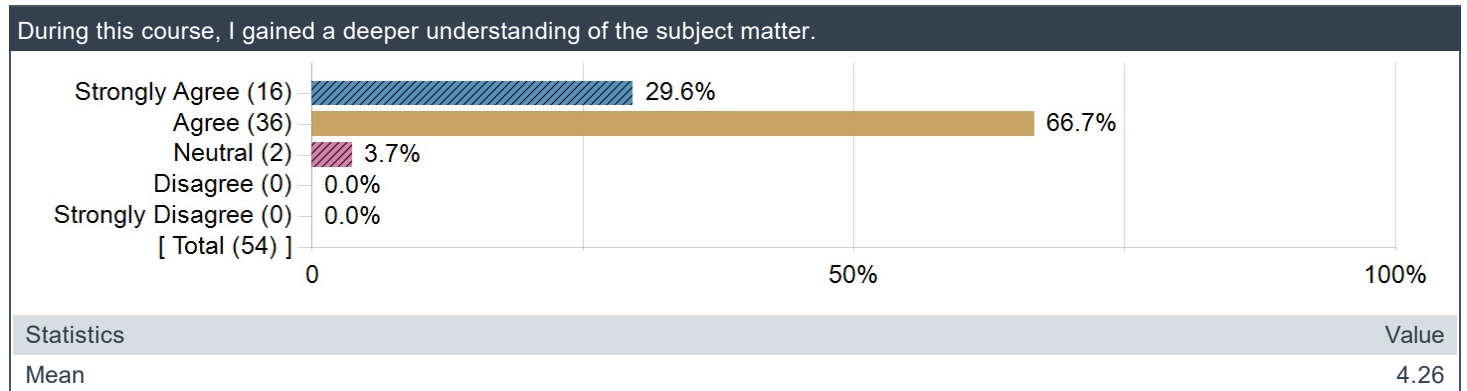
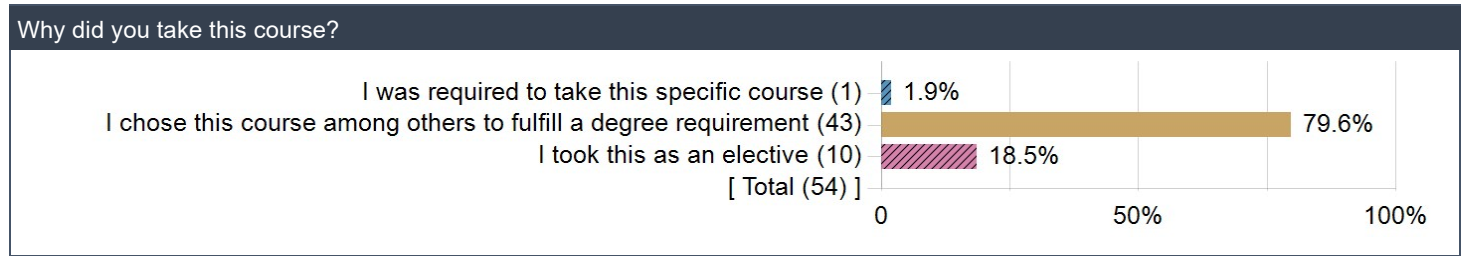
The Mean is calculated by adding all of the weights for a single question and dividing by the number of respondents. The course workload question is not averaged.

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.

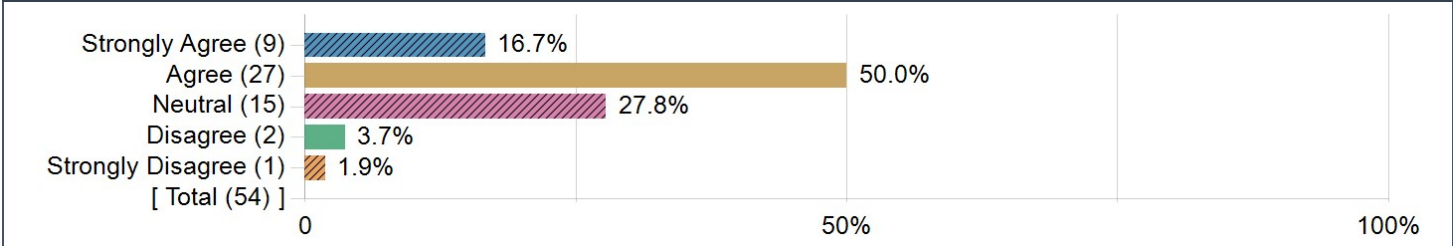
Creation Date: **Monday, December 18, 2023**

Course Questions



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.

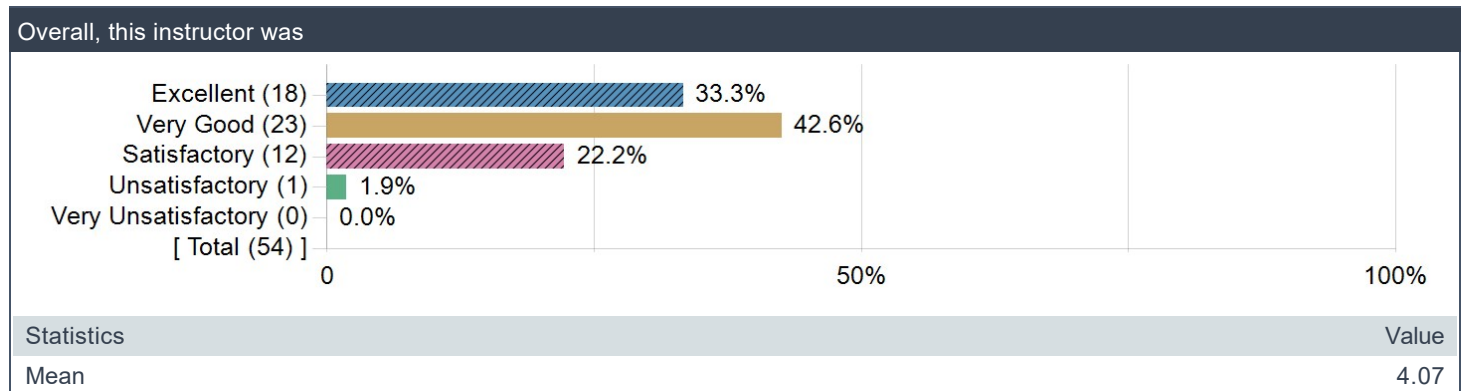
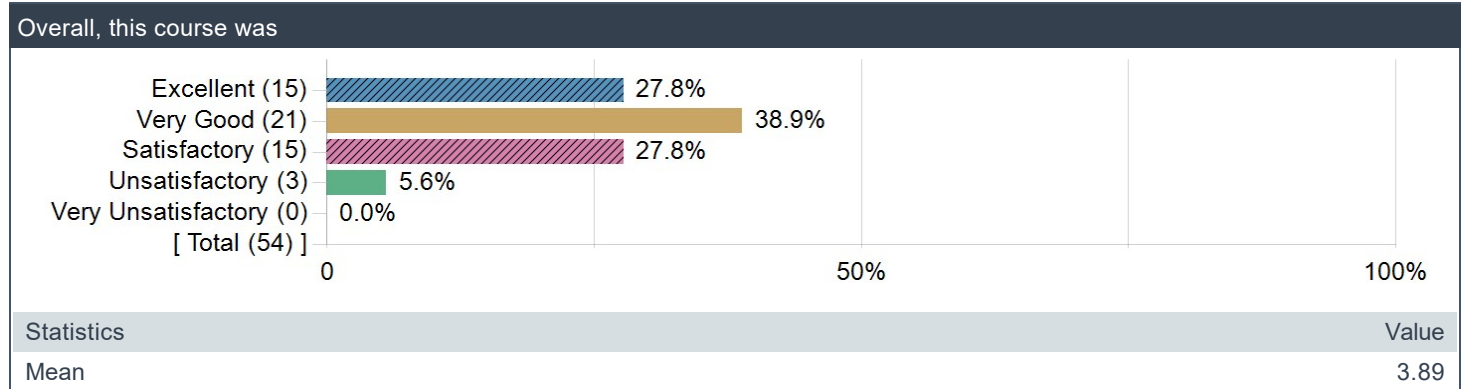


Statistics	Value
Mean	3.76

Instructor Questions

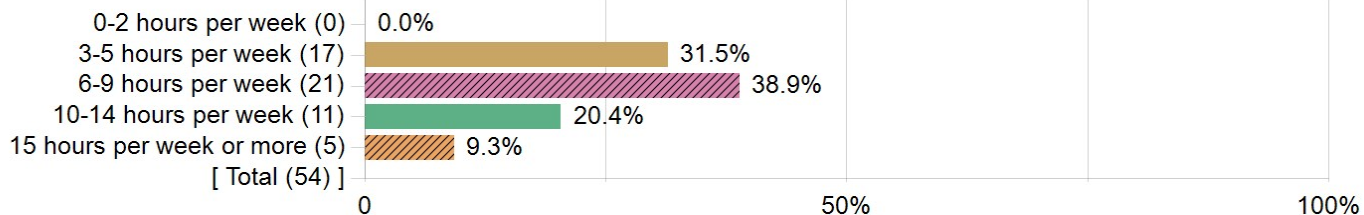
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	46.3%	50.0%	3.7%	0.0%	0.0%	54	4.43
The instructor fostered an inclusive learning environment.	46.3%	46.3%	5.6%	0.0%	1.9%	54	4.35
The instructor effectively explained the concepts and subject matter in this course.	37.0%	51.9%	5.6%	5.6%	0.0%	54	4.20
The instructional techniques kept me engaged in learning.	38.9%	44.4%	9.3%	5.6%	1.9%	54	4.13
The instructor checked for student understanding of the concepts presented in the course.	59.3%	37.0%	3.7%	0.0%	0.0%	54	4.56

Overall Questions



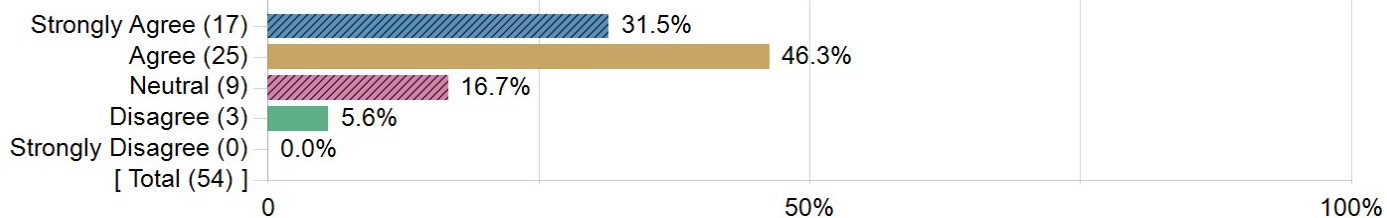
College, School, or Unit Questions

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	59
Response Count	54
Response Ratio	91.5%
Mean	3.07
Median	3.00
Semi-Interquartile Range	1.00
Mode	3
50th Percentile	3.00
Sum Total	0.00
Standard Deviation	0.95
Population Standard Deviation	0.94
Standard Error (base on SD)	0.13
Standard Error (base on PSD)	0.13
Aggregate Frequency 1	0.0%

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



Statistics	Value
Mean	4.04

Comment Questions

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

Comments
lecture recording and lecture notes
The exercises helped solidify new concepts.
The project gave me the best chance to improve and develop my understanding of full-stack development.
The daily quizzes ensured students were able to get frequent checks on their learning, which helped us understand what we need to improve on prior to the next class.
A lot of the OOP material was reused.
Cold calls kept me on my feet, and the projects taught me a lot about full-stack development and using tools that I've never encountered before.
The projects were really engaging and required a lot of outside learning. Overall, this allowed me to have a better understanding of software engineering and a great foundation for web development
cold calling
The lecture structure of walking through things, and the ability to use any resource for our projects
Having people around me to ask for help or having the TA with meetups to check on us helped us get organized and provided a source of help.
Figuring out how to work in our team and build out a full-stack app was the part that contributed most to be learning as a CS Major.
I thought that the daily quizzes were a nice way for me to ensure that I was following along properly. I also thought that the Socratic method of teaching was helpful since it forced me to acknowledge when I wasn't fully understanding the lecture — although the cold-calling did make me anxious at first.
The cold calling was, admittedly, a great motivator to show up and pay attention in class. Downing was also very effective in his lectures, and I never found myself bored or all that confused.
Projects
I found enforcing digital devices to be shut off and cold calling to boost interaction between the students and professor very helpful for learning. Other students being on digital devices can be distracting for other students and cold-calling allows students to think through the information being taught and apply the knowledge with guidance.
Ed discussion and TA meetings
class was well structured, learned a lot, projects were interesting
Self learning.
Love hate relationship with cold calling but it did help me pay attention.
projects, lectures
The quizzes helped motivate me to go to class every day and keep up with the content without falling behind.
The projects and readings were good
Group project gave me a much better understanding of React and backend development. Lectures were also very informative and interesting.
The lectures were very informative
The only thing I found to be really helpful to my ability to learn the material required to complete the projects are the guides put out by the TA's.
Having the projects be open-ended helped in making me be better at self-learning new topics. The TAs also gave references in case we needed help.
Projects mostly
Professor Downing explained class concepts very thoroughly and went over edge cases and such. This was very helpful because it gave me a much deeper understanding of the topics he went over.
I liked the first project
Professor Downing was able to explain the topics well.
I think Professor Downing's lectures were well-organized informative, and a valuable use of time. I learned a lot about Python semantics, which helped me complete the course assignments.
I think the projects really helped to gain a deeper understanding of SWE the practice, but not so much with the class
In-class quizzes keep me engaged since I would have to constantly revisit the material to ensure I get a good grade.

Comments
The teaching style was very effective at getting the points across to the students. I was able to learn and review until each I felt comfortable with each concept.
The recorded lectures and notes.
Professor Downing does a great job of teaching and keeping the class engaged. The course assignments are built around this philosophy as well.
The projects were realistic in simulating a working environment with expected results for us.
I thought I learned a lot from the hacker ranks.

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
grading system – taking the minimum of all graded categories still seems harsh. A weighted grading system would be ideal for this class.
That the website development involved many tools that we didn't learn in class.
The project and the quizzes were the most challenging aspects of the course. I believe the project is quite complex due to the number of components involved; however, it is rather rewarding. The quizzes aren't as valuable as I feel that they sometimes test material that isn't as relevant at times when the course revolves around a full-stack process.
I thought the in-class exercises were most challenging, as they required a complete understanding of the material, so it was easy to get stumped on some of them. To help students, chances to retake these exercises could be offered; this will ensure that students fully understand the topics before moving on to harder exercises.
Overcoming anxiety related to cold calling.
I wish we could learn more of the software engineering concepts in class. I understand the merit of having to learn new tools on our own, but I think it just encouraged a lot of stealing from other sites and less original implementations. I think if I learned these concepts in class I would've gotten a better understanding than what I have now.
Some of the niche things that we learned about the different coding languages were difficult to understand at times. I think it would be useful to have some more comments in the notes to describe in more detail what is going on in each example code block.
picking up new skills, but this was addressed by the TAs giving a list a resources on where to get strated
The most challenging aspect was quiz questions for me because it felt like we had little time and questions were very specific – maybe more time or more general questions
It was most challenging to be learning stuff on my own or blindly almost since everything was new to me.
95% of the material learned in class was not very applicable to the project, which spans a majority of the grades. Most of the quizzes were fine, but some of them tested incredibly minute things that were difficult to remember from previous lectures.
The projects were challenging because we were not being taught about the tools that we needed to use for them in class. I understand that it is supposed to mimic a real-world job, but I think any real-world job would have at least a bit of onboarding to allow new software engineers to ask questions about the tools to someone who is more experienced. At the very least, I think that all members of the course staff should be very familiar with all tools required for the project. Often, Canyon was the only one who seemed to be able to answer my questions — even the professor had to direct me to him.
The quizzes are a bit ruthless at times, as I think they hold a lot of weight in the final grade calculation. Also I do wish there was more partial credit to be gained from the exercises – even getting a one for some could help a lot and I feel that a lot of students do enough work on them to earn that grade.
Quizzes
I found the IDB group project most challenging for a multitude of factors. The project required a lot of self-learning and communication between group members, and there would be times when deadlines were sooner than expected. I think the system currently in place allowing slip days and resubmits is adequate enough for students to handle these challenges, as well as the TAs being helpful on Ed Discussion and in office hours.
The projects were challenging as they required learning many new technologies. Better explaining AWS could help future students.
group project groups were randomly assigned and having a good or bad group can make or break your experience and grade in the class
Attending lectures was challenging. They were unrelated to the assignments given outside of class.
The attendance policy
projects – some phases were much more difficult than others, it wasn't spaced out well. random groups make it so difficult to work, it should be an option but not a requirement. ed discussion is so difficult to communicate on, i would rather a text gc. projects

Comments
should have more relevance in the lectures, we shouldn't have to figure every single aspect out on our own
The disconnect between lectures and projects was challenging, and it would have been more helpful if even some of the project concepts were taught during lecture.
It was very challenging to learn all the skills required in the project – especially the frontend side. It would be very helpful to able to get a good set of resources to learn from.
The IDB project was very time consuming and required a great deal of effort. I do not think this challenge is necessarily a bad thing, as it helped me greatly improve my web development skills.
I struggled with the daily quizzes. There were a few where I think the time limit was a bit too restrictive
Almost nothing taught in class is relevant to the project and considering how large the project is and how much is required to complete it, having to learn all the tools needed to complete it on your own makes it take much longer than it would if the teacher spent any time at all covering them in class.
The projects being open-ended was challenging. I had to find out how to do things myself. I think how the TAs handled it by providing references is pretty good.
Nothing the class is not too demanding
There was a bit of a disconnect between the course homework and the course contents. I understand that it is intentional, but I do feel like I did not get to really apply what I learned in class because of this.
In class attendance quizzes – there are so many. Maybe once a week?
Having to plan and coordinate with teammates, as well as figuring out new technologies without guidance.
I didn't like the disconnect between lectures and the project
I felt that a lot of the information required for the SWE projects was self-learn, so it was difficult at first to figure out what I was doing.
I found the complete lack of lecture time on the course assignments very challenging. I understand this was done intentionally, but I found it overly challenging.
I did not like the disconnect between what was taught in class and what was asked for in the projects. That and the reading and blogs did not feel very useful either
The project was challenging but I enjoyed it.
I thought that the self-taught part of the class was a little difficult. I would love to just have a slight overview of some of the concepts for the project. The quizzes could sometimes be centered around very niche subjects we learned about, for the most part they were good but every once in a while there was a question about a brief topic we spent maybe 5 minutes on.
The daily quizzes were sometimes a bit difficult and the problem sets were a bit difficult with the given time.
The most challenging parts of the course were the projects. This is because we are using tools that are mostly new to us, and we need to learn them on our own. The instructors provide us with the resources, but there is no hand-holding involved. This is okay though, because it's similar to how things would go in the real world. I actually appreciated this aspect, as it forces us to learn on our own, and to learn how to find our own answers.
There is a disconnect between lectures and the projects at times making it hard to progress in the projects with no help unless we help hours are attended. For example, learning how to use specific tools or applications was not taught, leading to a learning curve. I feel that if there was a small lecture or demonstration on how to use the required tools in class it would greatly help students understanding.
I hated how much pressure and stress it was.