CIT261: Ronda Hollander

Following are ways that I demonstrated professionalism:

1. During the semester, I contacted my professor many times to make sure I could be successful in this course. Early on I was struggling with coding topics and reached out to better understand how I could manage my assignments better.
2. I submitted code samples with communications and links to my github account as-well-as my links to my PowerPoint presentations and links each group recorded session.
3. I asked for assistance with localStorage as I didn’t realize that localStorage and LocalStorage are different.
4. When code recommendations were provided, I tried to extend my code to include the suggestions.
5. I tried to turn in code in a timely manner. As I completed code topics each week, I would upload my code and provide links with instructions.
6. I attended on-line team sessions every week and even on my birthday.
7. Included with this submission is a PowerPoint presentation that highlights the five (5) lessons I taught. Links to the following pages organize all my code submissions and presentation.

**Click on the links below to see all the details:**

* + Class Code Examples Submitted:

<http://rhollander777.github.io/index.html>

* + Team Session Recordings & PowerPoint Presentations

<http://rhollander777.github,io/index-teaching.html>

Professionalism PowerPoint Presentation

1. I used a combination of PowerPoint presentations as-well-as code examples to teach concepts to my fellow students.
2. I did teach every time that I was scheduled for a total of five (5) lessons.
3. Our team would communicate outside of the team meetings using google hangouts.
   1. Jacob was struggling with his code so we provided links to additional resources for him to study and solve his problems.
   2. I also did some code peer review and provided feedback to Mike and Jacob.

1. I started building a list this semester of good resources by topic. When Jacob would need help, I would send a link or share code examples to help extend learning. Mike shared a link to site with API to get ideas for pulling data. For example, the following links were shared for CSS:

|  |
| --- |
| https://tympanus.net/Tutorials/AnimatedButtons/index4.html |
| https://css-tricks.com/snippets/css/a-guide-to-flexbox/ |
| https://css-tricks.com/controlling-css-animations-transitions-javascript/ |
| https://www.w3schools.com/css/css\_positioning.asp |

1. I was not aware that I was supposed to post to the iLearn Discussion Board for this class. I was under the impression that we were to work within our groups and we did.
2. Yes I did help with code review. Mike Summers and I discussed and reviewed his code several times.
3. When I found out that I didn’t understand the difference between localStorage and LocalStorage, I didn’t stop there. I continued to work on my code to better understand and apply what I was learning. Even though I struggled, I have become more knowledgeable and will continue to learn.
4. I continued to write down what I have learned in my journal and reflect on how I can do my coding better. There are a million different ways to code a solution and I am always trying to get better, streamline my functions, and write complete code for both positive and negative use cases. What new programmers often forget to do is program for all the ways the user will make a mistake. When you write good code, you take everything into consideration and try to communicate in easy English with the user when they make a mistake or you need them to do something.
5. I think professionalism in the workplace is very important. If you have ever worked in an environment where co-workers, supervisors, or managers are unprofessional it causes undo stress. Unprofessionalism costs businesses millions of dollar each year in lost productivity due to problems, rework, miscommunication, and lack of teamwork. Companies that communicate, work together, and try to solve issues at the earliest possible time-frame are much more successful.
6. As for the ideal team configuration, it really depends upon the type of products and services your team are providing. In an Agile environment, a small close-knit team with a good band-width of front-end and back-end coding knowledge is needed. This team should be approximately 3-6 people who can work very closely together in a collaborative environment to produce functionality in a short time-frame (approximately 3-6 month).

If the team is working on a multi-phase, multi-year project, the dynamics will be completely different. There may need to be multiple teams using a combination of waterfall and agile. Some teams my work on requirements management, others on configuration, others, on migration, others on interfaces, while still other work on testing and implementation just to name a few. In this case the team may be a total of dozens to get a major software initiative across the finish line. In this case one or more strong project managers are needed to ensure all the moving parts are orchestrated well so that all the individual pieces come together in a logical and timely manner.

Ideal team meetings also vary depending upon whether you are using an Agile or Waterfall development method. Agile teams have daily scrum meeting that should be no longer than 15 minutes and cover hot tops, roadblock and progress. While waterfall team meeting may need to be broken up by the various team, groups, disciplines, or topic areas. No matter what meeting is held, it should have an agenda, participants should be prepared and engaged, the discussion should stay on topic, and someone should capture decisions and follow-up items.

SUMMARY:

Our team was professional in our interactions with each other. What was unprofessional is that several members dropped out without any notification which leaves the team holding the bag. We had to take on additional topics to cover those who did not follow through with their commitment.

At first it was a little weird for me to meet with our group but as we worked together it became easier. Instead of it feeling force, we were able to share more openly, ask questions, and rely on each other. I do believe that when you have to teach someone else you are more intentional about your learning.

When I would get stuck, many times my classmates did not have the extended knowledge to help. Many of the topics and tools were new to our group so it made it more challenging to extend your knowledge. The one thing that I didn’t like is that I felt we were teaching ourselves rather than being taught. I missed having the professor extend our knowledge by connecting some of the dots and helping us to not go down bunny rabbit trails. It would have been nice to have had our instructor visit our team meetings but I did appreciate the feedback I received on code topics submitted.