**Discussion Board Forum Assignment:**

In the past two weeks, I have improved my knowledge on how to use Node.JS and its own test runner for use with assert and test cases, and have applied the tests to the program code as part of the assignments. I had also learned that:

* async/await is best used in my case with promises for seamless and fast database integration alongside queries.
* try/catch being used more often helps with determining what errors happen and how the program can move on if instructed.
* The mysql2 library can work by itself or importing the promises part of it.

I have improved upon my knowledge in:

* How to utilize MySQL with programming languages in general.
* Preparing queries to use before executing them.
* The practices in assertion for testing.
* Handling the program architecture (that being N-Layered).

I found that the most interesting part of Node.JS and server-side JavaScript alongside MySQL was the use of asynchronous programming made to better connect them together, especially with how much more improved the program runs as well as the increased efficiency in the code and database operations to where other tasks would not be blocked nor prevented.

Website resources yet again worked best for me, specifically the API explanation from the Postman blog of using Postman, Node.JS and MySQL together. This is because it covers the aspect I was looking for where database connectivity can be used, and on a web/cloud server for instance. Online videos also help still, but not as much as sources, and there aren’t as many videos of Node.JS with MySQL and Postman used together.  
Forums continue to take very long for me, due to having to also go through non-relevant and incorrect information provided.  
Documentations took the least time, for the reason that they have concise information.  
Resources: <https://blog.postman.com/build-an-api-with-postman-node-js-and-mysql/>, <https://www.linkedin.com/pulse/database-integration-postman-xmysql-umme-habiba/>, <https://medium.com/@vishnu_squareshift/simplifying-database-connections-with-cloud-sql-node-js-connector-31fd2c85bbce>, <https://stackoverflow.com/questions/51955252/using-mysql2-npm-package-on-google-cloud-sql>.

The Work Breakdown Structure that I use remains fair and I find has sufficient detail. The process I had for creating one was good enough to use; as always, improvements can be done.  
The Gantt chart still uses due dates that are intended to be close by to the due date, but that still resulted in making it neither overestimated nor underestimated. I still kept getting denied to use the intended dates to match closely to what I want.

**Student Feedback Part:**

Star Rating: 5  
  
I enjoyed reading your post! I had taken up and been through the database connectivity experience where CRUD was used and test cases were used. I haven’t used the same sort of resources, but I had been looking for something like them when I did my research, like documentation. YouTube hasn’t worked the best for me, and isn’t something that I had used much here, but it still has helped me with my project and I can for sure relate to that. You did well on your project, and I wish you all the best in improving it further.