**Reflective Report**

**Phase 3 and 4**

Nouraldin Hassan

Contents

[What Was Learned About my Programming Language 3](#_Toc184140135)

[The Best Resources for Me to Learn Are 3](#_Toc184140136)

[WBS, Project Management Software, Reflection of Time Estimation 3](#_Toc184140137)

[Discussion Board Post Archive 4](#_Toc184140138)

[References 5](#_Toc184140139)

# What Was Learned About my Programming Language

The programming language that I have continued to learn is Server-Side JavaScript utilized with Node.JS, with the addition of MySQL. Node.JS itself is a runtime environment that allows software developers to run JavaScript code outside of a web browser. It is used primarily for Server-Side Applications. MySQL is a relational database management system (RDBMS) that allows users to store, manage, and retrieve structured data efficiently, often for use in server-side applications and programs. MySQL has been integrated with Node.js to manage and query structured data.

The following topics have been learned alongside the programming language:

* C.R.U.D. Functions for database tables.
* The use of imports/requirements from the mysql2 module and data configuration.
* Importing and exporting a CSV file to and from the database.
* How to better run Node.JS’ own unit testing service.
* How to better utilize callbacks.
* How to make the program run faster via async/await.
* The use of typeof for determining the type of a variable.

The strengths that I have continued to acknowledge in Node.JS and Server-Side Javascript is that it handles asynchronous activities very well, to where I/O operations are nearly instantaneous for simple command line operations, as well as being able to use it for both frontend and backend development, and having a large ecosystem of libraries and modules for as many purposes as required, all open source too [1] [2].

The strengths that I found in MySQL is that it can handle database operations well with good queries, especially with large server operations, as well as having priority over “data security with its access privilege system”. Given that it has existed for decades, as well as having a high reputation among other RDBMS softwares, it is considered an industry standard. [3].  
  
The weaknesses I have continued to acknowledge in Node.JS and Server-Side Javascript however called Callback Hell, where multiple nested callbacks result in very indented code that makes it hard to maintain and comprehend, and Node.JS being single-threaded meaning that CPU-Intensive tasks limit its capability and therefore affect its performance [4] [5].

The weaknesses that I found in MySQL are that debugging tools there are not considered to be good as other databases, as well as a limited amount of scalability for large-scale databases. Circumventing the scalability issue can be done via sharding – which is a process that involves separating the large-scale database into smaller and more manageable segments that are distributed across multiple servers [6] [7].

# The Best Resources for Me to Learn Are

The NodeJS documentation, as well as the MySQL documentation.

General resource types are listed in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Resource | Time Consumption | Usefulness | Rank |
| Books | High | Medium | Medium |
| Documentation Sites | Low | High | High |
| Online Videos | Medium | High | High |
| Training Sites | Low | High | Medium |
| Other Websites | Medium | Medium | Medium |

The most effective resources I found were the Web Documentation sites, due to the information displayed at a practical level and shows quick and simple detail on how a given function, method, or parameter works. The second most effective resource is online videos, since they show a visual way on how to perform a particular action or actions with the given programs, as well as some being either informative or quick-to-the-point so I can choose the learning style and curvature.

# WBS, Project Management Software, Reflection of Time Estimation

The timing continues to be miscalculated – albeit less – to where it was still difficult to find out what has been over-estimated and what has been under-estimated, due to the time-providing restriction where I could not use the dates and times that I pick to have it span the start date and the due date (i.e. 8:00 AM cannot be changed to 11:59 PM for the same date for starting and ending).  
  
That has at least better and further taught me to plan dates ahead of time and format them based better on actual estimated or intended completion, rather than on exact due dates of completion in reference to Brightspace assignments, so I can at least submit the assignments utilizing the charts before the due date ends and can have some room for improvement via quick fixes and resubmitting when it is needed.

# Discussion Board Post Archive

**Discussion Forum 01 Post:**

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.

**Discussion Forum 02 Post:**

In the past week, I have improved my knowledge on how to use Node.JS and MySQL for use with trying to connect with Amazon Cloud SQL services, and have attempted to connect the database with the problem, only to be met with constant connection issues and functions not applying like they should, as well as trying to deal with Amazon’s multiple services that cause complications and confusion.  
In spite of that, I decided to focus my efforts on applying the instruction material and references on making a horizontal bar graph and applying other options, with only the horizonal bar graph yielding success. Regardless, I had also learned that:

* Typeof <variable> helps with finding the given variable type (i.e. console.log(“Retrieved: “ + typeof records); outputs to “Retrieved: String”).
* Multiple readline prompts stacked aren’t the best uses, but they worked well in my case.

I have improved upon my knowledge in:

* try/catch being used more often helps with determining what errors happen and how the program can move on if instructed.
* Utilizing the CLI for advanced outputs.

I found that the most interesting part of Node.JS and server-side JavaScript alongside MySQL with Bar Generation was the use of asynchronous programming made to speed up the process of making a bar chart, especially with how much data is used to make the program’s bar chart generation run at a fast yet efficient speed. What I also found interesting was the use of “typeof”, which can be helpful in situations where values need to actually be part of specific types in a given program, whether for advanced program development or otherwise.  
I tried to use two module libraries for making the graph but it ended up causing too much outputs of a graph to be displayed, as well as repeated information being printed, so I decided not to use module libraries at all.

When it came to the original project that was intended to be made, website resources once 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.  
When it came to the bar graphs, I figured it out on my own but took a painfully long time to get together.  
Forums yet again 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; though as always, improvements can be done.  
The Gantt chart yet again 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.

# References

|  |  |
| --- | --- |
| [1] | Node.JS, "JavaScript Asynchronous Programming and Callbacks," *Node.JS*, 03 July 2024. [Online]. Available: https://nodejs.org/en/learn/asynchronous-work/javascript-asynchronous-programming-and-callbacks. [Accessed 16 October 2024]. |
| [2] | E. Miquelito, "10 Powerful Node.js Libraries Every Developer Should Know About," *DEV Community*, 27 April 2023. [Online]. Available: https://dev.to/emiquelito/10-powerful-nodejs-libraries-every-developer-should-know-about-2c5g. [Accessed 16 October 2024]. |
| [3] | K. Vyas, "8 Major Advantages of Using MySQL," *Datamation*, 3 February 2023. [Online]. Available: https://www.datamation.com/storage/8-major-advantages-of-using-mysql/. [Accessed 6 December 2024]. |
| [4] | GeeksForGeeks, "Understanding Callbacks and Callback Hell in JavaScript," *GeeksForGeeks*, 13 August 2024. [Online]. Available: https://www.geeksforgeeks.org/what-to-understand-callback-and-callback-hell-in-javascript/. [Accessed 16 October 2024]. |
| [5] | S. Ghosh, "Node.js Architecture: The Single-Threaded Event Loop," *Sudipta.dev*, 21 April 2024. [Online]. Available: https://sudipta.dev/why-is-node-js-single-threaded/. [Accessed 16 October 2024]. |
| [6] | D. Raymond, "MySQL: 10 Cons & Disadvantages with 5 Alternatives," *ProjectManagers.net*, 25 January 2024. [Online]. Available: https://projectmanagers.net/mysql-10-cons-disadvantages-with-5-alternatives/. [Accessed 6 December 2024]. |
| [7] | GeeksForGeeks, "Disadvantages of Using MySQL," *GeeksForGeeks*, 9 November 2022. [Online]. Available: https://www.geeksforgeeks.org/disadvantages-of-using-mysql/. [Accessed 6 December 2024]. |