Narrative for Enhancement 3

Cheyenne Nave

CS-499

SNHU

Dr. Maciosek

June 4, 2025

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.For Enhancement 3, I am continuing to use the same Calculator program that I have been using throughout the course. So far, I have added keyboard functionality and improved operations by including more complex mathematical concepts such as Logarithms, Cosine, Sine, Tangent, and more. To build on that, I created a login page and a register page that users can sign in through to access the calculator.

This was a major learning curve for me, as it has been months since I have taken a class here at SNHU about APIs or Databases in general. I had to do a lot of research and take a day off work to make sure it was working and functional.

A screenshot of a computer

AI-generated content may be incorrect.To start, I created a database to store the usernames, passwords, and user IDs through MySQL. In my previous narratives and plan, I wanted to use MongoDB to store historical calculations. When I changed to a login page, I switched to MySQL since it is relational. Because this is a small program, I’m not too worried about scalability. I used MySQL Workbench to build my database and it took me a few tries to get the database saved correctly. I had to review the guidebook in regards to schemas and how they are used in database creation, and basic terminal commands for connection when I was testing it in the Windows Command Line.

A screenshot of a computer

AI-generated content may be incorrect.Once I had my database together, I created a folder called ‘db-login’ to store all the necessary files to run the program. I used Node.js to create the API, and required Express. I created my ‘dbServer,js’ file to build and run my connections and routes. I added fail-safe errors to pinpoint exactly where errors occurred as I developed, which led me to a very clear understanding of API response codes. Using response codes instead of creating my own error message to display was so much easier because other software that I used (like POSTMAN) was able to throw the code and describe the error instead of displaying my input which could be misinterpreted by someone else. Then, I decided my security was lacking so I implemented the use of ‘bcrypt’ to hash user passwords. This way, if they were ever leaked, it would only be a hash sequence.

A screenshot of a computer

AI-generated content may be incorrect.

This enhancement has been the most difficult for me, but I’m proud of the work I have accomplished. I have met all course outcomes. Other than updating the style of my portfolio and adding my other enhancements and code review, I have no updates to provide.

Cites

* Maurya, A. (2022, September 22). mysql.createConnection vs mysql.createPool in Node JS. *Medium*. https://adi22maurya.medium.com/mysql-createconnection-vs-mysql-createpool-in-node-js-42a5274626e7
* *REST API response codes*. (n.d.). MailerSend. https://www.mailersend.com/help/rest-api-response-codes#:~:text=201%20%2D%20Created,webhooks%20and%20inbound%20routes%20endpoints.
* *MySQL :: MySQL Workbench Manual :: 8 Database Development*. (n.d.). https://dev.mysql.com/doc/workbench/en/wb-develop.html
* GeeksforGeeks. (2024, February 5). *Postman Tutorial*. GeeksforGeeks. https://www.geeksforgeeks.org/postman-tutorial/#sending-api-requests