CS499

Sheel Patel

October 6, 2024

Databases Narrative

The artifact I’ve chosen for my Databases category is the text-based game which I wrote in Python. The premise of the project was to create an application in which the user was required to obtain a number of items before being able to enter the final room. This was the final project for the IT-140: Intro to Scripting course. I took this course during my first term in 2022 at Southern New Hampshire University.

I chose this artifact because it was the very first code that I wrote. I thought it would be another good showcase of how I have evolved as a developer. When I first wrote this code, I was only thinking about the end goal and put no consideration into things like security. For my enhancement, I have chosen to add a database for storing high scores. The original game didn’t have a scoring system, so I also had to add that. Additionally, due to storing the high scores in a json file, I added encryption and decryption using a key, so the scores are safely stored. This also required the creation of a main menu so user could navigate to the high scores or to the game. The high scores themselves are calculated as the time left after the user has won the game.

**A screen shot of a computer program

Description automatically generated**

The code displayed above shows how the high score database and encryption has been implemented. The code uses a key that I created to encrypt and decrypt the high scores. The high scores are stored in a json file because I found this to be the best solution for a game of this type and scale. If I were to put this game online, I might use MongoDB instead. When I was implementing these changes, I had some difficulties. I initially overcomplicated the scoring system and time system by trying to create a more elaborate score. However, after stepping away and coming back to the code, I realized that I could take the simpler approach of using the time remaining as the score. Another issue I ran into was the save location for the high scores json and secret key. The files were originally being stored and looked for in the windows user directory so, I had to add some additional logic to store the files in the same directory as the python script. If I were to publish this game, I might change the location where the files are stored to a more hidden location so they can’t be tampered with. Finally, the most difficult addition to my code was adding the encryption and decryption for the high score json. As I mentioned before, I had trouble with where the files were stored. In addition to this, I had not implemented encryption and decryption in Python so I had to research how this worked. The method I used in the code above is the best and simplest method which I found.

The changes I implemented in my enhancement directly apply to the course outcome regarding approaching coding with a security first mindset. I believe I did good job of this by implementing encryption and decryption of the high scores. Additionally, because I used a json file to store the scores rather than a SQL database, I didn’t have to implement SQL injection prevention. Overall, the adjustments made to the original code created a game that was more robust and secure.