**Handling Errors: Hands-On Project 4-1**

Rachael Herman

Colorado State University Global

ITS340: Introduction to Programming with JavaScript

Professor Alex Lazo

March 6, 2022

**Handling Errors: Hands-On Project 4-1**

Handling errors in JavaScript is a common and essential practice for programmers. Debugging a program requires understanding the three main types of errors one will encounter, including syntax, run-time, and logic errors. A "bug" is defined as "any error in a program that causes it to function incorrectly" (Vodnik & Gosselin, 2015, pg. 213). This week, the project involves handling errors by using try, throw and catch blocks to make sure users choose at least one crop from the list. These statements allow the programmer to try a block of code, catch any errors, throw an error message, and finally do something regardless of the result (*JavaScript Errors,* n.d.). However, a finally statement was not used in this scenario. The code demonstrates successful completion of the error handling process but will need to address bugs in the future.

**The Process for Project 4-1**

Using data files provided by our textbook, Chapter 4 focused on creating a program that recommends specific farm equipment depending on acreage, crops, months of activity, and preferred fuel source. After adding authorship information and organizing the data files, the testFormCompleteness() function was removed from the verifyCrops() function and replaced with try, throw, and catch statements that check for marks in the Crops section of the form. If none are chosen, the system throws an error to select at least one crop from the list. The catch statement occurs outside the try block to display the throw message if the Crops section is incomplete or “false.” The catch statement is placed outside the try block but before the end of the function to allow the program to complete the try block before executing the catch and ending the function (*Try,* 2021). A finally statement was not added to this code but will likely be done later. Finally, I changed the values of the global variables for acresComplete, cropsComplete, and monthsComplete to false to accommodate the error handling code.

**Figure 1**

The JavaScript Code to Handle Error

Text

Description automatically generated

**Figure 2**

Changed Values

Text

Description automatically generated

**Figure 3**

Tuba Farm Equipment Error Handling Test

Graphical user interface, application

Description automatically generated

**Challenges and Experience**

I encountered no significant challenges in the course of this project. There were two typos in the instructions that could cause a few issues if directly copied. First, in the for statement of the try block, “I = 0” should be “i = 0” or it throws an error. Second, in the if statement of the try block, “cropscomplete” should be “cropsComplete." Overall, this chapter and the project were good starting points to understanding the syntax and need for bulletproofing code with good try and catch statements that help programmers (and users) avoid unintentional errors.

**References**

*JavaScript Errors* (n.d.). W3 Schools. Retrieved on March 5, 2022, from <https://www.w3schools.com/js/js_errors.asp>

*Try...catch…finally statement (visual basic).* (2021, November 5). Microsoft. Retrieved on March 5, 2022, from <https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/statements/try-catch-finally-statement>

Vodnik, S., & Gosselin, D. (2015). *JavaScript: The web warrior series.* (6th ed.). Cengage.