BOOK BANAZA

Final Reflection

Having never experienced using both java script and python prior to this unit and being more familiar with practical aspects of design through programs such as photoshop and illustrator I found the final project of using coding elements to tell the ‘story’ of design difficult.

As such, this project was by far the most difficult unit I have been involved in which required learning a range of coding parameters. However, it is clear the possibilities for which python could be used for is limitless. In time, and with me further exposure to Python, the limits to its ability to import certain information and display it in a way more readable to the user is endless. String objects and reading elements to output a final product is important.

Overall, I felt like my final design was successful. The code I was able to create using Python, scans the isbn.txt file and populates the ‘**Book Banaza’** homepage successfully. The automation which Python can create and generate content is something I am still having trouble to understand, using the correct terms to either read or write code.

My vision for ‘Book Banaza” was to provide a straight forward webpage which displayed the user with a visual representation of the books they had in their library based on the ISBN’s in the .txt file, a cover image for the book, a link to find further details of the book and a snapshot of book information (author & summary of the book). I was unsuccessful in being able to .write the information from Cover Library that would populate the author and summary of the book.

The other difficulty I was unable to overcome, was the ability for the .py to read and adapt to ISBN’s within the isbn.txt file which had alternative numbers or symbols and did not fit the standard input of “13” digits. My work around for this was to include instructions on the “About” page explaining the ISBN’s within the .txt should not have any space or “-“. This was more a policy approach which allowed me to define acceptable input as a warning to the user and if bad input was provided, then it is the fault of the user.