[10570628](mailto:1%30%35%370%36%32%38@%6d%79%64%62%73%2e%69%65) CA Report

For my CA information system, I decided to focus on books and novels as my domain of interest to focus on. With this in mind I designed a website around a library for its premise, with the database to search for books that might be found in the library.

Starting with my database, I coded it in Python using Visual Studio, with the database using three primary topics to organise and search for books. These three were the books title, the author of the book and the published release date of the book. I entered eight books in total, some as stand-alone books, where one author wrote one book, such as Bram Stokers Dracula. Others that had one author to several books such as J. R. R. Tolkien who has four books in my database. The reason for this is for testing purposes, to check that if a user searches for an author will all books appear correctly or to check if there’s an error with the database in case it threw up the incorrect information.

I did have some difficulty creating the database initially, I took several different approaches and experimented with my own code and some tutorials I found online. My current system works by allowing the user to search for the author to locate a book written by them. It can be changed in the database to search by the book title or by book release date, however for testing I preferred searching by author for a balanced and quick form of testing, as searching by title would mean typing lengthy titles to see one book, and date will be quick but also only reveal one book. Searching by author allowed me quicker and more efficient testing as I can see multiple books under one author.

The main Python script has an attachment code I made labelled “Library.py”, which allowed for a quicker, slightly automated way to create new entries into the database. The other main aspect on my database is under “sqlite3.connect” I set it to “Memory”. By setting this to memory, when the database stops running, it “forgets” what’s elements are in the database. Running the code again will put these elements back in. This was very helpful for testing as without it and hardcoding the database I would not be able to add the same book more than once without causing an error to occur, which would slow down my testing as I’d need to code it out or constantly add a new put into the database to test anything. Keeping it in this format allows for a user to easily add books into the database without them causing an error by entering a book that’s already there, which I believe would be ideal in a real-world situation as a database in a library could grow drastically. My database is also capable of updating and removing books from the database. While might not be needed with the memory style of the database I still think it could be important to keep if its needed. By default, however, the remove code in my database has been coded out to prevent coding out a book while trying to add it.

For the website aspect I created a site based on Clondalkin’s Library to match with the database. It has four pages for a user to access, “Home”, “Search”, “Contact” and “Sign in”. The home page acts as the websites index and hub page. The search page is designed to be where the database is, allowing the user to enter their desired book or author to locate the book. It also has a linked connection to Google Maps which will show the library’s location on the map. While this page is intended for the database, I had difficulty getting it to work correctly on the site itself as I couldn’t put the python code into the HTML coding, and despite building a python system in a program like Sublime, I met little success in this regard. In the contact part of the website, I have it set that a user can enter info which simulates a message sending before sending the user back to the home page. The sign in page simulates a user to sign into an account, with a JavaScript in place to stop and password less than eight characters and one capital letter to be accepted.

Other than that, I did edit the website in PuTTy as well, however it is primitive as I cannot get any CSS into it. I’ve also tried connecting my Azure website to my GitHub account via the GitHub workflow, however it doesn’t seem to work for the time being.