**Bookstore Application – Report**

The following online web application system for a bookstore is designed using ASP.NET MVC (C#). It has preregistered three books into the bookstore as a requirement for the web app. This web app consists of three View pages, including Login View, Index View (homepage) and Reserve View. Two users are preregistered for the Login View of this web app. The credentials of both users are listed below.

User 1:

Username: TC

Password: 1234

User 2:

Username: SQ

Password: 1234

The functions of this bookstore web app are listed as follow:

1. This application allows the user to list all the books in the bookstore.
2. This application allows the user to search for a specific book in the bookstore using the name of the book as the keyword for searching.
3. This application allows the user to reserve a book in the bookstore and once the reservation is successful, a booking number will be generated.
4. This application will not allow the user to reserve a book if the book has been reserved by another user.

There are three versions in the GitHub repository I have submitted.

Version 1: A working application with two different methods of storing data (Database and In Memory Repo).

Version 2: Splitting Command and Query repos in the controller (first step of CQRS design pattern).

Version 3: Splitting Database Context into Command and Query by using two Databases (second step of CQRS design pattern).

Version 1:

This is a working bookstore web app with two different methods of storing data. First method of storing data is by using database (sqlite3) while the other method is using the in-memory repo. All of the data of the books will be stored in a class called GlobalData.cs. The user is able to swap between these two repos by changing the scope in Program.cs.

Version 2:

The implementation of CQRS design pattern of this web app is split into two steps. In this version, the interfaces and repos are split into both command and query.

Version 3:

In this version, the database is split into two different databases which are command and query database. The data in both databases are synced by as the command repos in the controller will update the command and query database if a command is initiated.