Task 0 : Explain what you are doing/ going to accomplish

*Make user log in accounts and an updateable cart*

Task 1: Sketch interface design

Task 2: Identify any classes required

*User class, Book class*

Task 3: Identify information to be displayed

*Each book with book name, description and stock*

*User name and password and books stored in cart*

Task 4: Identify user inputs

*Clicks on + icon to add to cart*

*Inputs user name*

*Inputs user password*

Task 5: Identify any constants or existing data if required

Books

Users

Task 6: Identify indexed data structures

Books - Stores all book objects and data

Users – stores all user objects and data

Task 7: Determine what calculations are necessary

*Stock - 1*

Task 8: Develop a modular structure for your program

*@route(/cart\_updated/passed book ID)*

*@view(cart\_updated)*

*Define cart\_updated function:*

*Convert book\_id into int*

*Creat variable found\_book*

*Loop through book in books:*

*If book.id is = to book\_id*

*Set found book to book*

*Remove 1 from book stock*

*Append book to users cart*

*Return book as a dictionary*

Task 9: Define the functions identified

Task 10: Address any relevant implications such as usability,

functionality, legal/ethical requirements.

*Make sure all the books are child friendly*

*Be sure to let the users know that the password is not secure.*

*Hide the password as they are typing it in, for this testing do not permantently store the password.*

*The usability is smooth and each icon is clearly labelled as a representation of the pages functionality.*

Task 11: Document test cases for testing the program

*Try putting in the wrong name and password.*

*Try putting in both accounts details.*

Task 12: Refine the plan

Task 13: Document testing

*Putting in false data crashes the page.*

*Both accounts work fine*

*.*