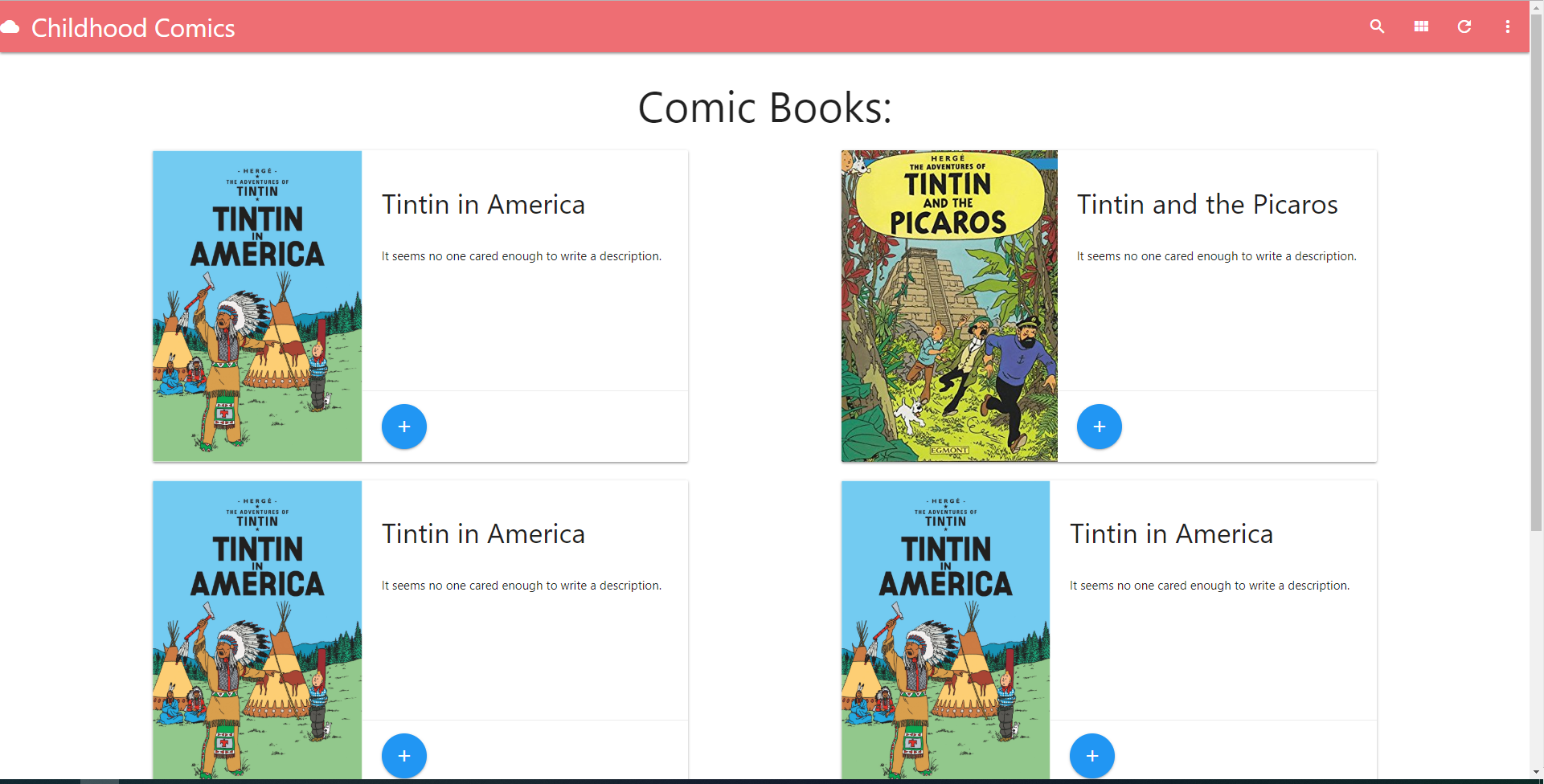
Task 0 : Explain what you are doing/ going to accomplish

*Make a store page that allows users to add books to cart after browsing*

Task 1: Sketch interface design



Task 2: Identify any classes required

*Users class, books class*

Task 3: Identify information to be displayed

*Each book with book name, description and stock*

Task 4: Identify user inputs

*Clicks on + icon*

Task 5: Identify any constants or existing data if required

Books

Task 6: Identify indexed data structures

Books - Stores all book objects

Users – stores all user objects

Task 7: Determine what calculations are necessary

*None*

Task 8: Develop a modular structure for your program

*Class Books:*

*Make appropriate constructor that created all the data needed*

*Def web page:*

*Passes and manages data that is required for given web page*

*Run server*

Task 9: Define the functions identified

*Import bottle functions*

*Import a counting function*

*Define user class:*

*Define variable books\_in\_cart*

*Create constructor*

*Pass object name*

*Define book class:*

*Define name*

*Define description*

*Define cost*

*Define stock*

*Define photo*

*Define constructor(name, cost, stock, photo):*

*Define object name variable*

*Define object description variable*

*Define object cost variable*

*Define object stock variable*

*Define object photo variable*

*Define books array:*

*Create book object passing required variables*

*Create book object passing required variables*

*Create book object passing required variables*

*Create book object passing required variables*

*Create book object passing required variables*

*Define users array:*

*Add user named moses*

*@route(/)*

*@view(index)*

*Define index page function:*

*Do nothing*

*@route(/store)*

*@view(store)*

*Define store function:*

*Data = dict of book\_list filled with books*

*Pass data back to website as book\_list*

*@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*

*In HTML:*

*Loop through books and create a card for each*

*If the book is in stock*

*Add cart button*

*Else*

*dont*

*Run server with port 6969*

Task 10: Address any relevant implications such as usability,

functionality, legal/ethical requirements.

*Make sure all the books are child friendly*

Task 11: Document test cases for testing the program

*Test the site by running it repeatedly and checking the buttons work correctly. Go back and forth between pages to make sure they update*

Task 12: Refine the plan

*.* *Added small bar next to + icon showing cost and stock levels. Button goes red when out of stock*

*Updated books, no more repeats.*

Task 13: Document testing

