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

Book – An object for each book holding its cost, description, name, stock and photo.

Task 3: Identify information to be displayed

A tile for each book will be displayed on the page displaying its image, name, description, stock and cost

Task 4: Identify user inputs

Simply clicking on the + icon will add the book to the cart, no other inputs needed

Task 5: Identify any constants or existing data if required

The data of each book

Task 6: Identify indexed data structures

One that holds each book object

Task 7: Determine what calculations are necessary

Stock – 1 for each book added to cart

Task 8: Develop a modular structure for your program

from bottle import run, route, view, get, post, request and static\_file

from itertools import count

Define a class called Book:

Set variable '\_ids' to count (0)

Set variable 'name' to "How the heck would I know"

Set variable 'description' to "It seems no one cared enough to write a description."

Set variable 'cost' to "At least $0"

Set variable 'stock' to "Probably none"

Set variable 'photo' to "Void"

Define object constructor function(Passing parameters: self and

Set variable 'name' to "How the heck would I know",

Set variable 'description' to "It seems no one cared enough to write a description." by default,

Set variable 'cost' to "At least $0" by default,

Set variable 'stock' to "Probably none" by default,

Set variable 'photo' to "Void" by default):

Set variable 'self.id' to next(self.\_ids)

Set variable 'self.name' to name

Set variable 'self.description' to description

Set variable 'self.cost' to cost

Set variable 'self.stock' to stock

Set variable 'self.photo' to photo

Set variable 'books' to [

an iteration of class Book(Passing parameters: name= "Tintin in America", cost= 25, stock= 7 and photo= "https://images-na.ssl-images-amazon.com/images/I/912vIXwG23L.jpg"),

an iteration of class Book(Passing parameters: name= "Tintin and the Picaros", cost= 25, stock= 5 and photo= "https://images-na.ssl-images-amazon.com/images/I/61%2Bp-97QRIL.jpg"),

an iteration of class Book(Passing parameters: name= "Tintin in America", cost= 25, stock= 7 and photo= "https://images-na.ssl-images-amazon.com/images/I/912vIXwG23L.jpg"),

an iteration of class Book(Passing parameters: name= "Tintin in America", cost= 25, stock= 7 and photo= "https://images-na.ssl-images-amazon.com/images/I/912vIXwG23L.jpg"),

an iteration of class Book(Passing parameters: name= "Tintin in America", cost= 25, stock= 7 and photo= "https://images-na.ssl-images-amazon.com/images/I/912vIXwG23L.jpg"),

an iteration of class Book(Passing parameters: name= "Tintin in America", cost= 25, stock= 7 and photo= "https://images-na.ssl-images-amazon.com/images/I/912vIXwG23L.jpg")

]

#Index page

Set route('/')

Set view('index')

Define a function called index():

pass

#Store page

Set route('/store')

Set view('store')

Define a function called store():

Set variable 'data' to dict(books\_list = books)

return data

#Cart updated page

Set route('/cart\_updated/ is less than book\_id is greater than ')

Set view('cart\_updated')

Define a function called cart\_updated(Passing parameters: book\_id):

Set variable 'book\_id' to int(book\_id)

Set variable 'found\_book' to None

for book in books:

if book.id is equal to book\_id:

Set variable 'found\_book' to book

break

found\_book.stock -= 1

Set variable 'data' to dict(book = found\_book)

loged\_user.books\_in\_cart.append(found\_book)

return data

run(host='0.0.0.0', Set variable 'port' to 6969, reloader = True and debug = True)

Task 9: Define the functions identified

Class book creates book objects that store all the necessary data.

Task 10: Address any relevant implications such as usability,

Functionality, legal/ethical requirements.

Task 11: Document test cases for testing the program

Task 12: Refine the plan

Task 13: Document testing