System Requirements Specification Index

For

Python E-Library Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

E-Library ConsoleApplication

System Requirements Specification

1 Project Abstract

E-Library Console Application is a python application with python collection,where it allows users to manage the books and issue the books from the library.

2 Common Constraints

1. Take console input of number of books (n).
2. Take input of details of each book and store in a collection.
3. Take console input of number of books to be issued(m).
4. Take input of details of books to be issued.
5. Take input of details of book to be issued and store in a collection.
6. Show the books stock remained after issuing books.

3 Template Code Structure

Resources (Instances)

Class	Description	Status
Book	This class contains all the properties of the Bookin the constructor.	Already implemented.
Issue	This class contains all the properties of the Issuein the constructor.	Already implemented.

Resources (Business Logic)

Class	Description	Status
BookInventory	This class contains all the methods which are used to write the business logic for the application.	Partially implemented (available only template code).

Resources (Exceptions)

Class	Description	Status
ISBNAlreadyExistsError	Custom Exception to be thrown when trying toadd a book for which ISBN is already exists	Already created.
ISBNDoesNotExistsError	Custom Exception to be thrown when trying to issue a book for which the ISBN does not exists	Already created.
BookNotAvailableError	Custom Exception to be thrown when trying to issue a book which is already issued.	Already created.

Resources (Main class)

Class	Description	Status
MainClass	This is the class which is having methods to be used to read data from the user or display the data. All the business logic methods of the BookInventory class will be called from this class.	Partially implemented (available only template code).

4 Execution Steps to Follow

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.

3. The editor Auto Saves the code.
4. If you want to exit(logout) and to continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To run the application, use the following command

   python3 main.py

7. Mandatory: Before final submission run the following command

   python3 -m unittest

8. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
9. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

--------*--------