System Requirements Specification Index

For

Assessment Handling Application (Console)

Version 4.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
fullstack@iiht.com

ASSESSMENT HANDLING CONSOLE APPLICATION

System Requirements Specification

1.Business-Requirement:

Name of the Course : C#

Tool Stack : .net core console application

Problem Statement :

Build an application to work with Assessment handling like KBA handling, SBA etc.

Description :

1. Take console input from the user in the Main method of the program.

2. Take input of details of each assessment in the following format.
   Assessment Id, Assessment Name, Assessment Type, Assessment Date,Track
   Eg: 2, MyAssessment, MyAssessmentType,MyAssessmentDate,MyTrack

3. Create Assessment class(Entity) with
   a. AssessmentId as int
   b. AssessmentName as string
   c. AssessmentType as string
   d. Assessment Date as datetime
   e. Track as string
   f. Create one static method as below:
      i. static IEnumerable<Object> AddAssessment(Assessment assessment)
      which accepts Entity Object as input, add the input to the list

   g. Create another static method as below:
      i. static IEnumerable<Object> UpdateAssessment(Assessment assessment)
      which accepts Entity Object as input, check AssessmentId and update the details to the list

   h. Create another static method as below:
      i. static IEnumerable<Object> ShowAllAssessments()
      Display all the list objects in the console

   i. Create another static method as below:
      i. static IEnumerable<Object> GetAssessmentById(AssessmentId)
      Return assessment details by trackId

   j. Create another static method as below:
      i. static IEnumerable<Object> RemoveAssessmentById(AssessmentId)
      ii. Remove track details from list by trackId

2.Resources available:

2.1 Package: AssessmentHandling

Names	Resource	Remarks	Status
Package Structure/Project			
AssessmentHandling	Assessment.cs	This file contains all the properties related to assessment.	Already Implemented
	Program.cs	Program.cs is an important class that is the entry point of application and contains business logic of application.	Partially Implemented

2.2 Package: AssessmentHandling.Tests

Resources

Note: - Under the AssessmentHandling.Tests contain all test cases for code evaluation, please don't try to alter and edit it.

3. 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 the Application menu (Three horizontal lines at left top) Terminal → New Terminal.

3. On command prompt, cd into your project folder (cd <Your-Project-folder>).

4. To build your project use command:
   (AssessmentHandling /dotnet build)

5. To launch your application, Run the following command to run the application:
   (AssessmentHandling /dotnet run)

6. This editor Auto Saves the code.

7. To run the test cases in CMD, Run the following command to test the application:
   (AssessmentHandling.Tests/dotnet test --logger "console;verbosity=detailed")

   (You can run this command multiple times to identify the test case status,and refactor code to make maximum test cases passed before final submission).

8. If you want to exit(logout) and 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.

9. 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.

10. 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.