Library Management System

IIHT

Time To Complete: 3 hrs

CONTENTS

1 Problem Statement

A corporate organization has planned to setup a portal for managing library. Library managers can create/update/delete books.

2 Business Requirements:

o

Screen Name	Console input screen
Problem Statement	2. User needs to enter into the application. 3. The user should be able to do the particular operations 4. The console should display the menu i)add a subject ii)add a book iii)delete a subject iv)delete a book v)search a subject vi)search a book vii)exit

o

3 Implementation/Functional Requirements

3.1 Code Quality/Optimizations

1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

3.2 Template Code Structure

a. Package: com.assessment.hibernate

Resources

Class/Interface	Description	Status
Application.java(class)	This represent bootstrap class i.e class with Main method, that shall contain all console interaction with user.	**Not implemented**

B. Package: com.assessment.hibernate.model

Resources

Class/Interface	Description	Status
Book.java(class)	This represents entity class for Book	Already Implemented
Subject.java(class)	This represents entity class for Subject	Already Implemented

C. Package: com.assessment.hibernate.repository

Resources

Class/Interface	Description	Status
EntityDao.java(class)	This is an interface containing declaration of DAO method	Already Implemented
EntityDaoImpl.java	This is an implementation class for DAO methods. Contains empty method bodies, where logic needs to written by test taker	Not Implemented

D. Hibernate.cfg.xml

This is Hibernate configuration file, which contains all the configuration required for hibernate and to connect with database

4 Execution Steps to Follow

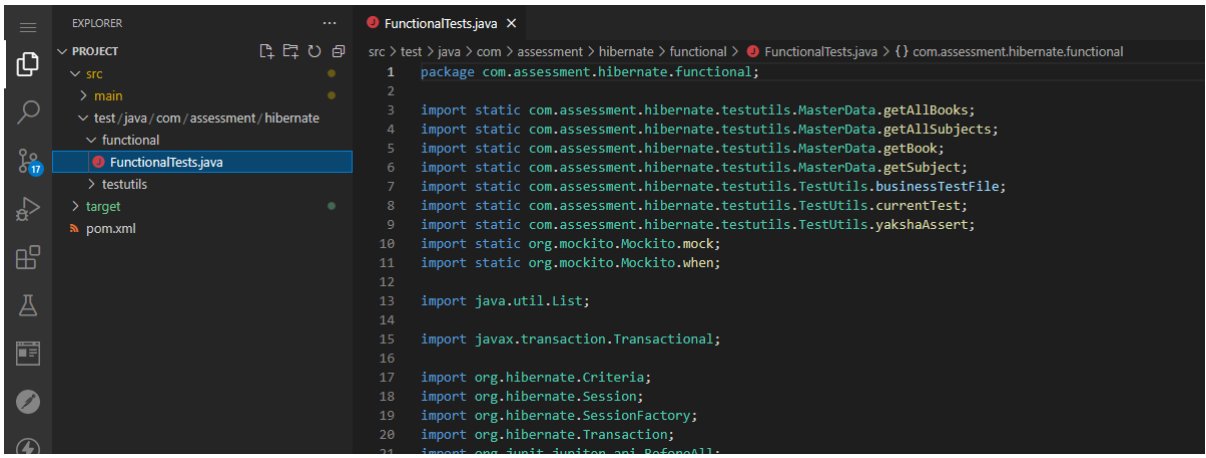
1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to the Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. To build your project use command:
   mvn clean package -Dmaven.test.skip
4. This editor Auto Saves the code.
5. 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.
6. 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.

7. Default credentials for MySQL:
   a. Username: root
   b. Password: pass@word1
8. To login to mysql instance: Open new terminal and use following command:
   a. sudo systemctl enable mysql
   b. mysql -u root -p
      The last command will ask for password which is 'pass@word1'

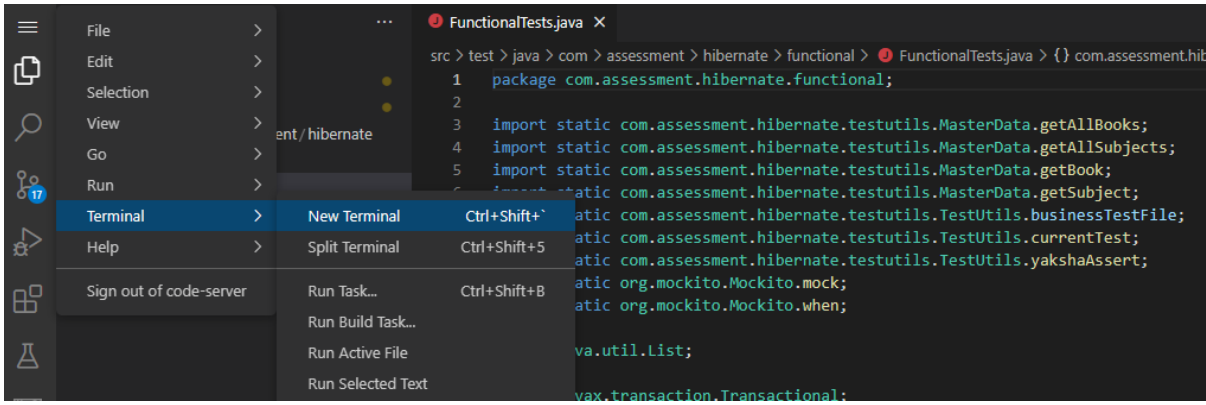
9. These are time bound assessments. The timer would stop if you logout (Save & Exit) and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
10. To run your project use command:
    mvn clean install exec:java
    -Dexec.mainClass="com.assessment.hibernate.Application"

11. To test your project, use the command

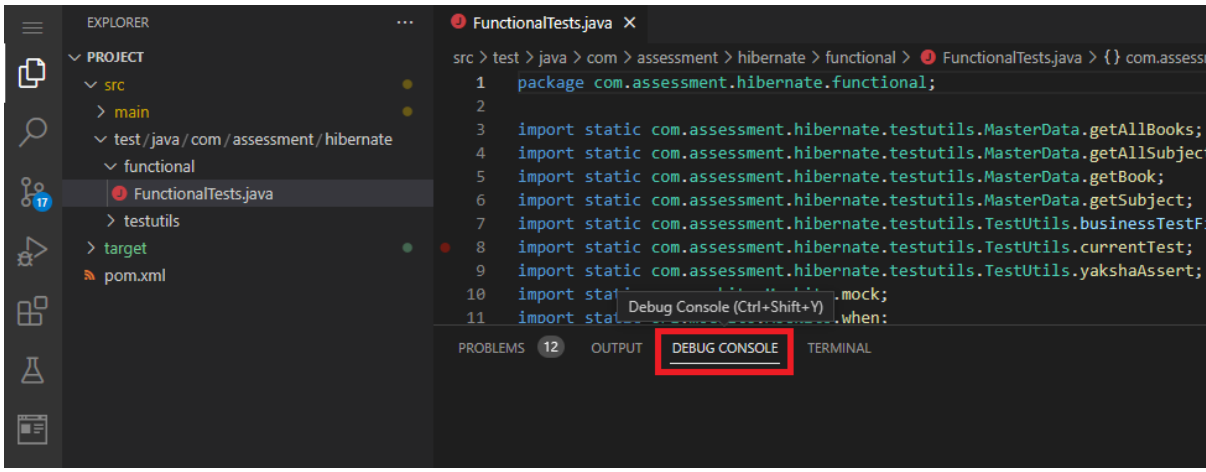
a. Open FunctionalTests.java file in editor



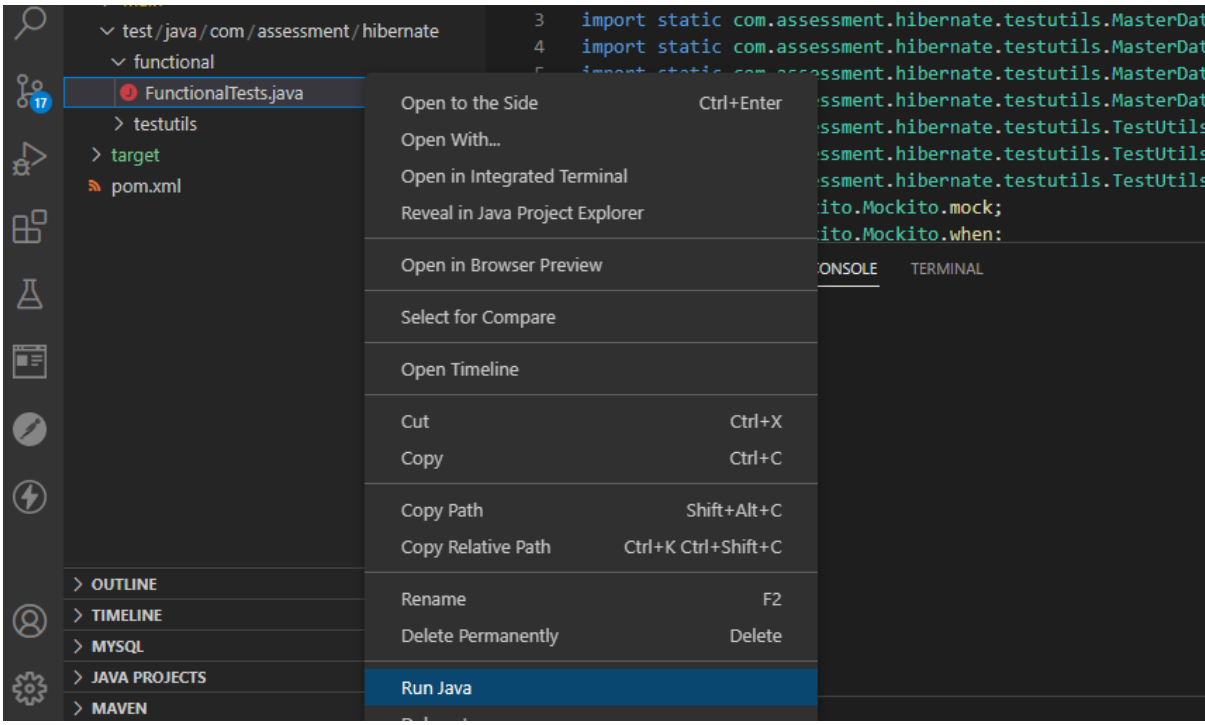
b. Open a new Terminal



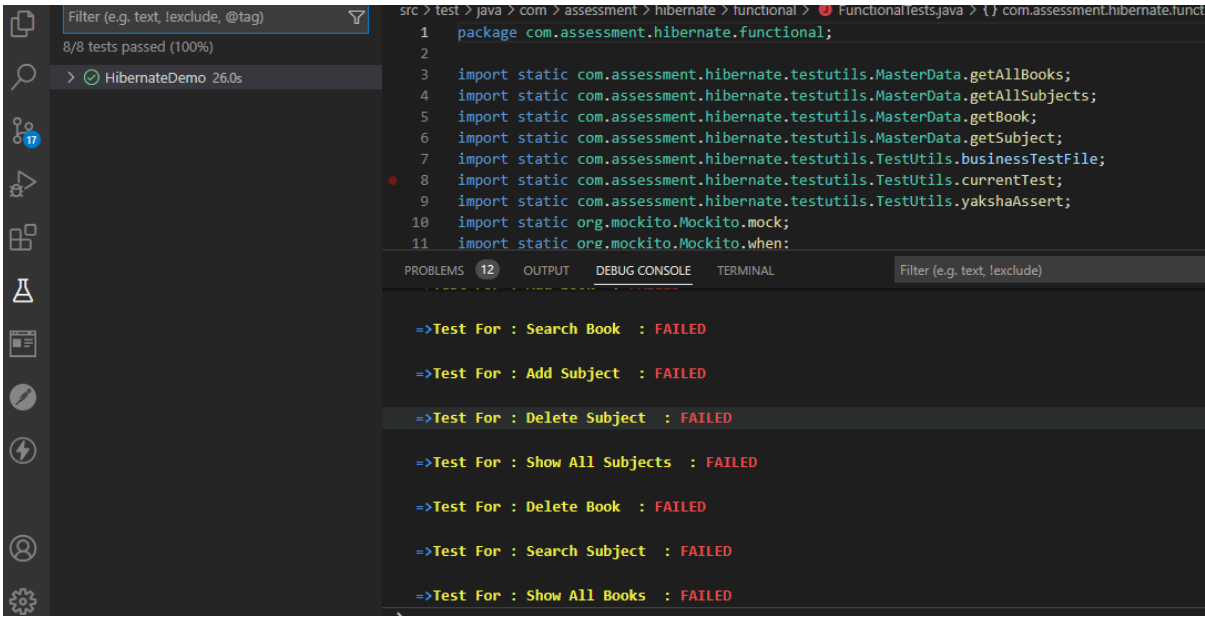
c. Go to Debug Console Tab



d. **Right click on FunctionalTests.java file and select option Run Java**



e. **This will launch the test cases and status of the same can be viewed in Debug Console**



12. **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.**