Online Library Management

IIHT

Time To Complete: 3 hrs

Contents

1 Problem Statement

The Online Library Management System empowers users to perform CRUD (Create, Read, Update, Delete) operations and search functionalities in different criterias on books and subjects. Users can create new book entries and subject categories, update existing book and subject information, delete books and subjects that are no longer relevant or available, and retrieve book and subject details for viewing.

2 Business Requirements:

Screen Name	Console input screen
Problem Statement	1. User needs to enter into the application. 2. The user should be able to do the particular operations 3. The console should display the menu 1) create subject 2) update subject 3) delete subject 4) view all subjects 5) create book 6) update book 7) delete book 8) view all books 9) search subject by name 10) list all issued books 11) search book by name 12) remove all subjects 13) remove all books 14) exit

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.onlinelibrarymanagement

Resources

Class/Interface	Description	Status
OnlineLibraryApplication.java(class)	This represents bootstrap class i.e class with Main method, that shall contain all console interaction with the user.	Partially implemented

B. Package: com.onlinelibrarymanagement.model

Resources

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

C. Package: com.onlinelibrarymanagement.repository

Resources

Class/Interface	Description	Status
BookDao.java(interface)	This is an interface containing declaration of DAO method	Already Implemented
BookDaoImpl.java(class)	This is an implementation class for DAO methods. Contains empty method bodies, where logic needs to written by test taker	Partially Implemented
SubjectDao.java(interface)	This is an interface containing declaration of DAO method	Already Implemented
SubjectDaoImpl.java(class)	This is an implementation class for DAO methods. Contains empty method bodies, where logic needs to written by test taker	Partially Implemented

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. sudo systemctl start mysql

NOTE: After typing the second sql command (sudo systemctl start mysql), you may encounter a warning message like :

System has not been booted with systemd as init system (PID 1). Can't operate. Failed to connect to bus: Host is down

>> Please note that this warning is expected and can be disregarded. Proceed to the next step.

c. mysql -u root -p

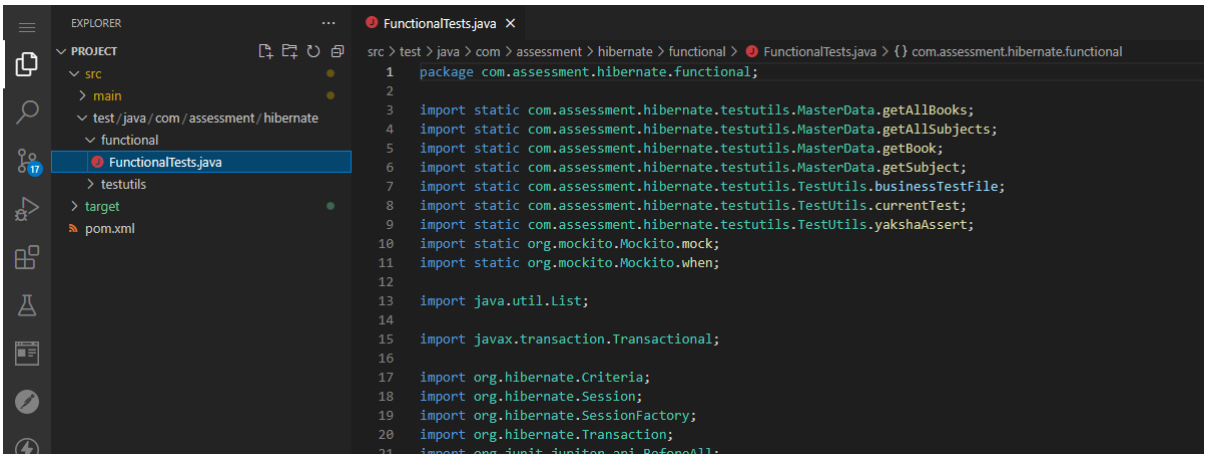
The last command will ask for password which is 'pass@word1'

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:**

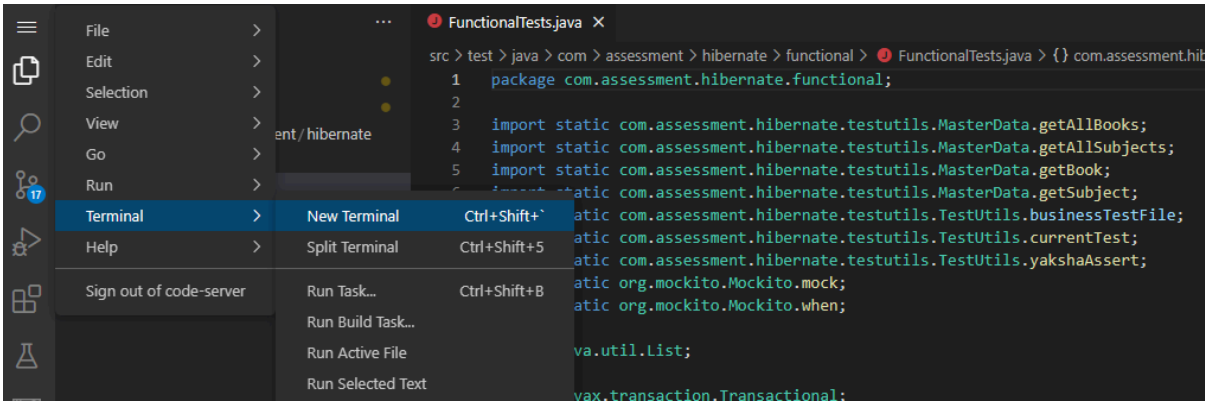
    mvn clean install exec:java -Dexec.mainClass="com.onlinelibrarymanagement.OnlineLibraryApplication"

11. **To test your project, use the command**

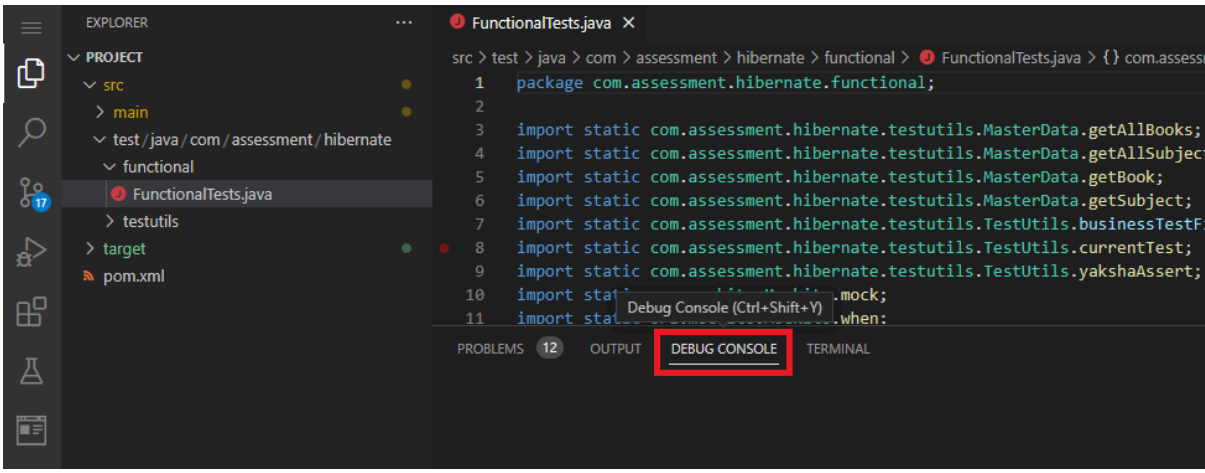
a. **Open FunctionalTests.java file in editor**



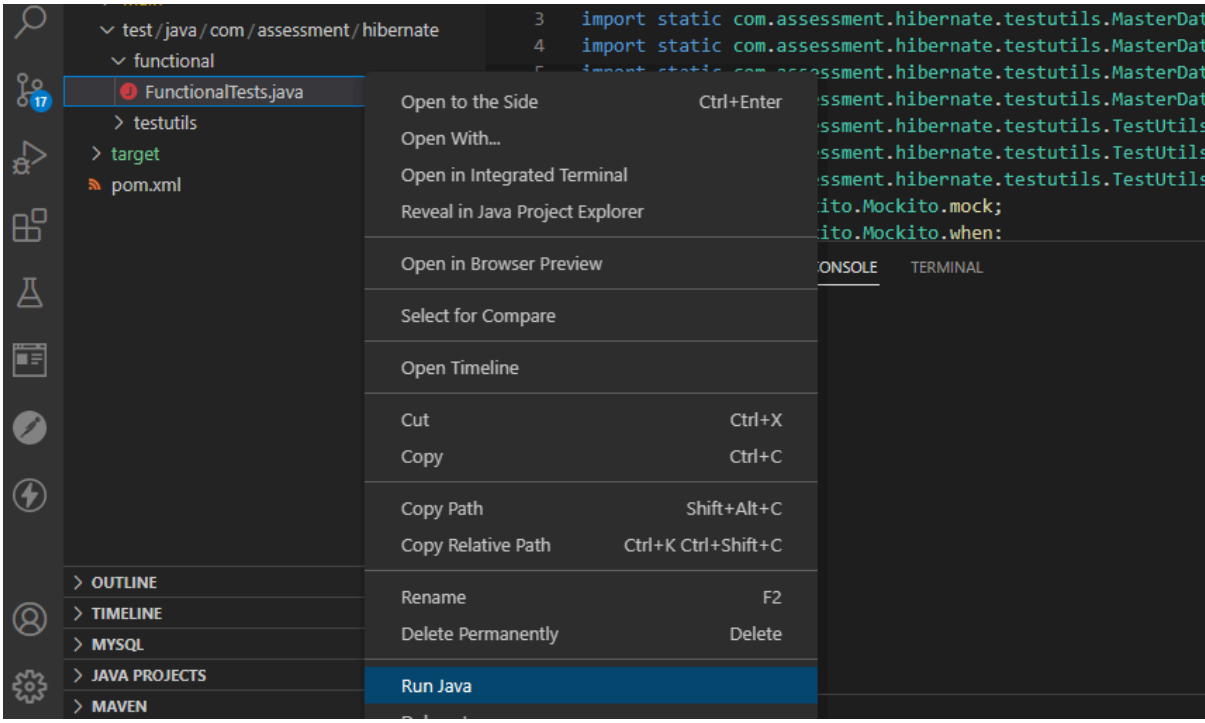
b. **Open a new Terminal**



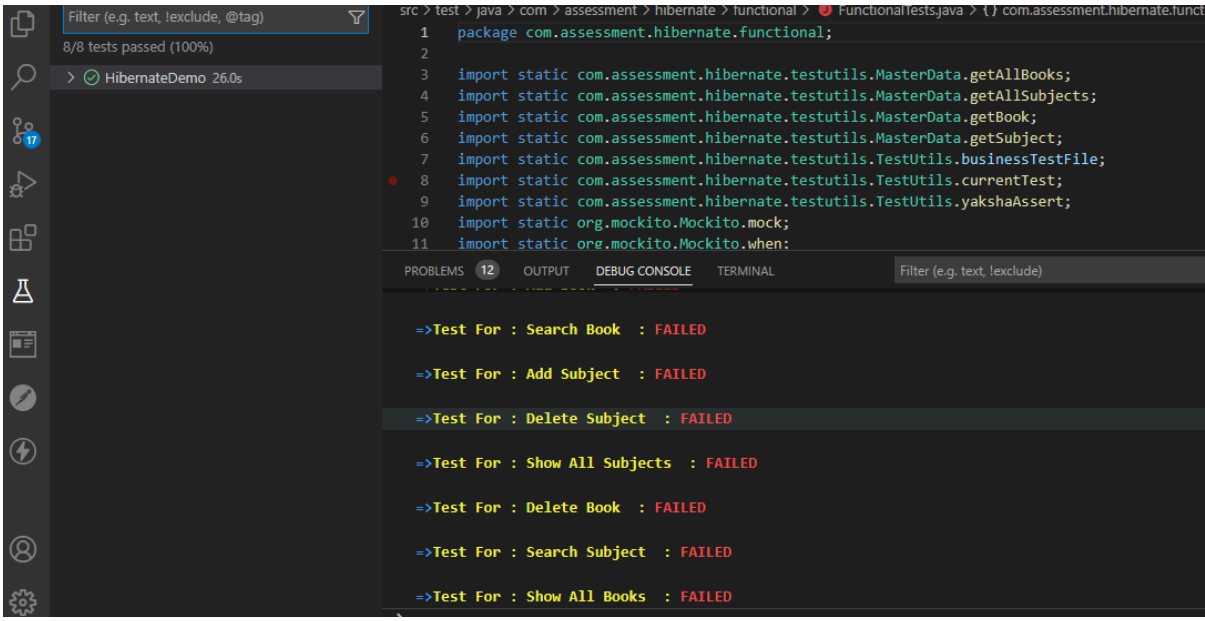
c. Go to Debug Console Tab



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.**