

# ONLINE FACULTY MANAGEMENT

IIHT

Time To Complete: 3 hrs

## CONTENTS

---

1 Problem Statement	3
2 Business Requirements:	3
3 Implementation/Functional Requirements	3
3.1 Code Quality/Optimizations	3
3.2 Template Code Structure	4
a. Package: com.onlinefacultyapplication	4
b. Package: com.onlinefacultyapplication.model	4
c. Package: com.onlinefacultyapplication.repository	4
4 Execution Steps to Follow	5

## 1 PROBLEM STATEMENT

---

The Online Faculty Management System enables users to perform CRUD (Create, Read, Update, Delete) operations and search functionalities in different criterias on teachers and departments. Users can create new teacher and department profiles, update existing teacher and department information, delete teachers and departments that are no longer active or needed, and retrieve teacher and department details for viewing.

## 2 BUSINESS REQUIREMENTS:

---

Screen Name	Console input screen
Problem Statement	<ol style="list-style-type: none"><li>1. User needs to enter into the application.</li><li>2. The user should be able to do the particular operations</li><li>3. The console should display the menu<ol style="list-style-type: none"><li>1) create department</li><li>2) create teacher</li><li>3) get department by id</li><li>4) get teacher by id</li><li>5) get all departments</li><li>6) get all teachers</li><li>7) update department</li><li>8) update teacher</li><li>9) delete department</li><li>10) delete teacher</li><li>11) search teachers by name</li><li>12) search teachers by subject</li><li>13) search department by name</li><li>14) exit</li></ol></li></ol>

## 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.ONLINEFACULTYAPPLICATION

#### Resources

Class/Interface	Description	Status
OnlineFacultyApplication.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.ONLINEFACULTYAPPLICATION.MODEL

#### Resources

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

### C. PACKAGE: COM.ONLINEFACULTYAPPLICATION.REPOSITORY

#### Resources

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

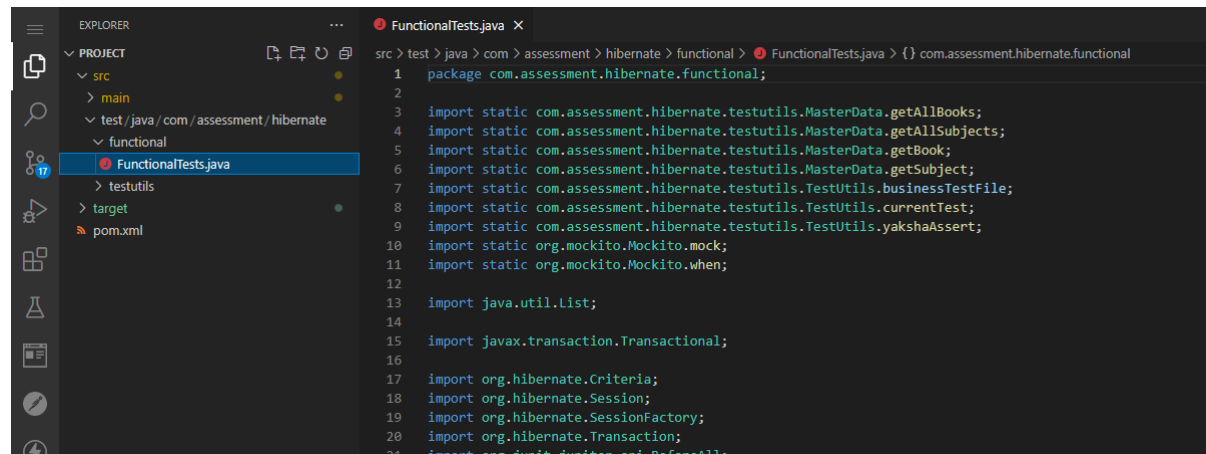
10. To run your project use command:

**mvn clean install exec:java**

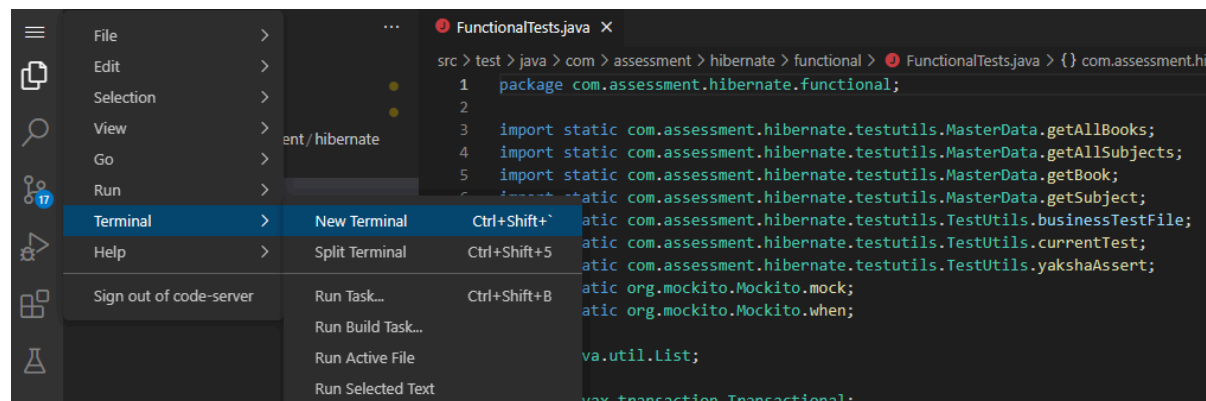
**-Dexec.mainClass="com.onlinefacultyapplication.OnlineFacultyApplication"**

11. To test your project, use the command

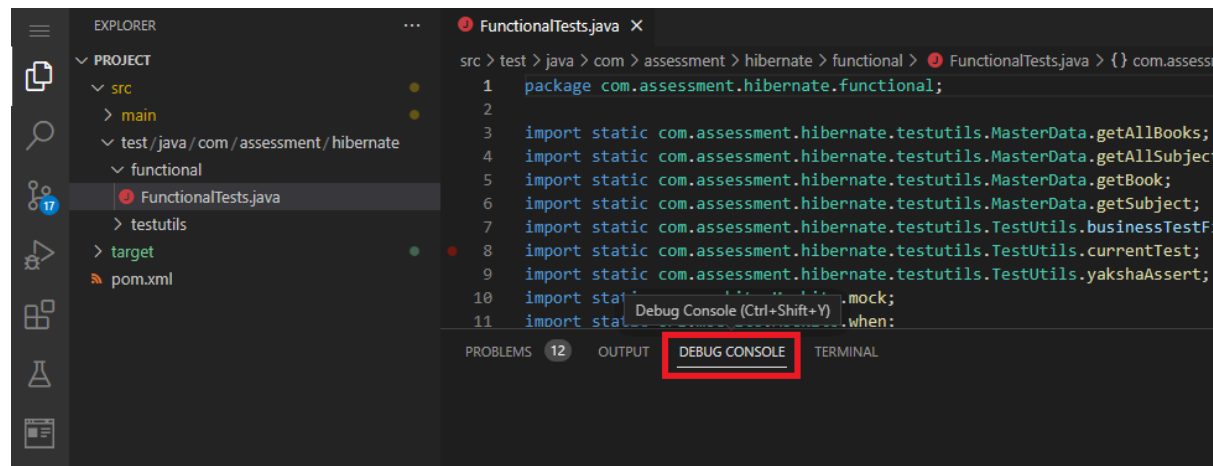
a. Open FunctionalTests.java file in editor



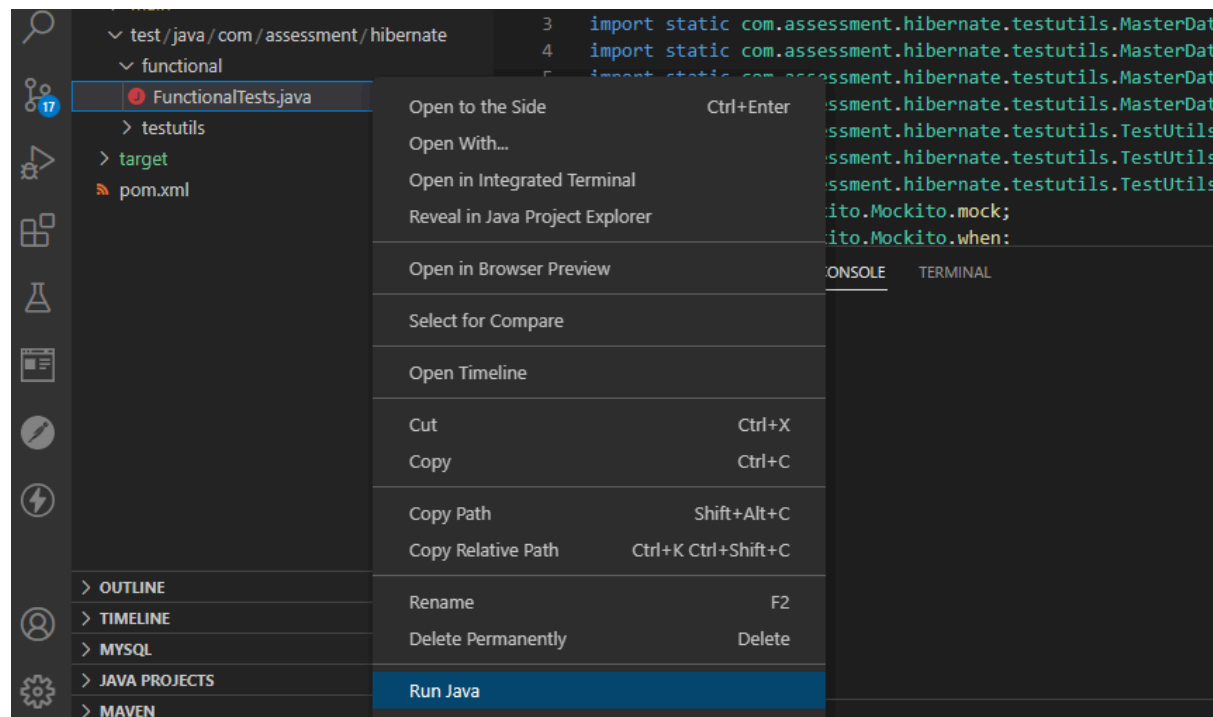
b. Open a new Terminal



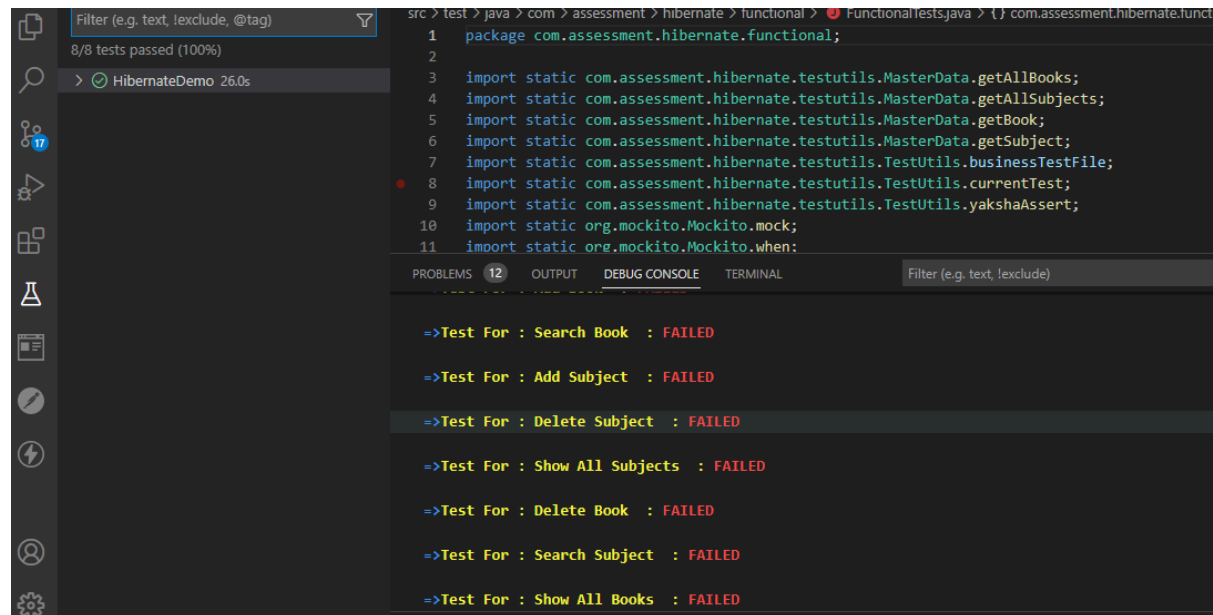
c. Go to Debug Console Tab



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.