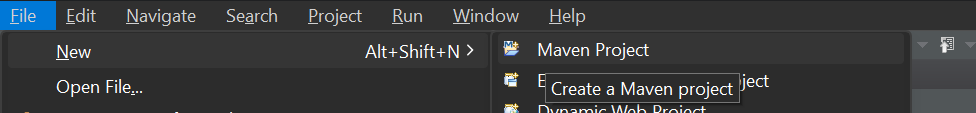
**Practical No : 05**

**Date:** 23/09/2024

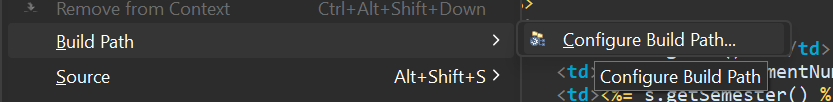
**Problem Statement:**

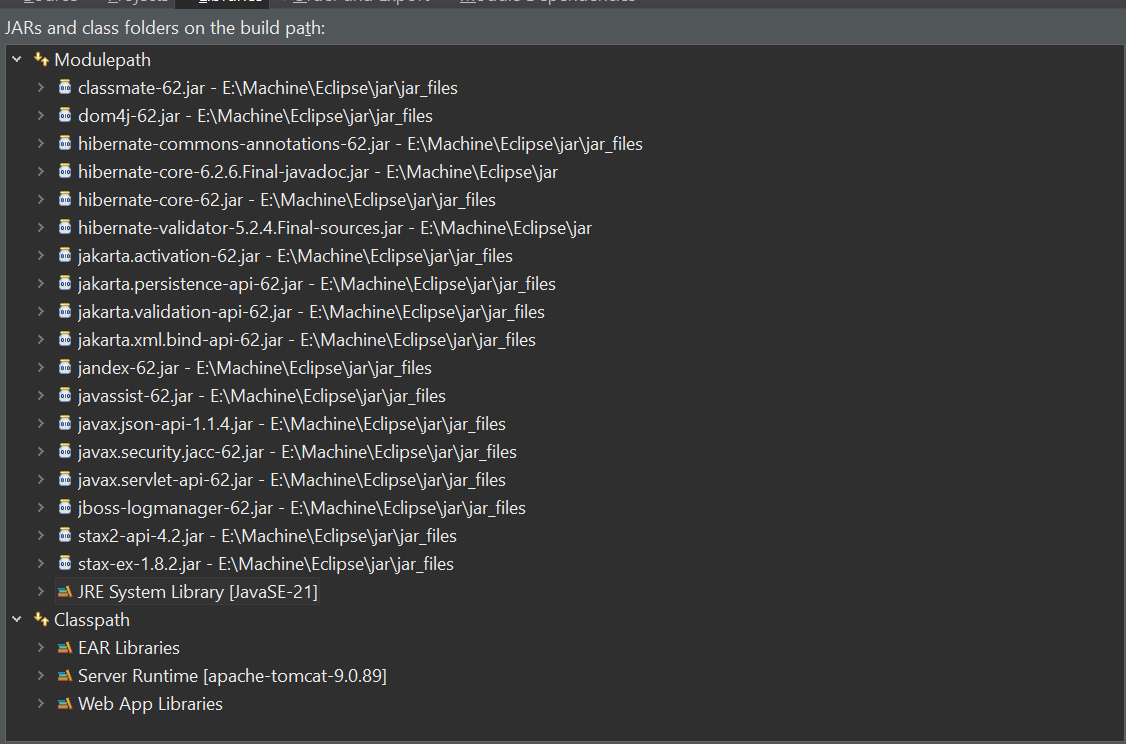
**2. Implement the Hibernate Application to store the student registration details.**

**Configuration:**

****

|  |  |
| --- | --- |

****

****

**Code:**

**StudentDAO.java**

package com.student.dao;

import com.student.model.Student;

import com.student.util.HibernateUtil;

import org.hibernate.Session;

import org.hibernate.Transaction;

import java.util.List;

public class StudentDAO {

public void addStudent(Student student) {

Transaction transaction = null;

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

transaction = session.beginTransaction();

session.save(student);

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public List<Student> getAllStudents() {

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

return session.createQuery("FROM Student").list();

}

}

}

**Student.java**

package com.student.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "students")

public class Student {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY) // Use IDENTITY for MySQL

private int id;

private String enrollmentNumber;

private int semester;

private double spi;

// Getters and Setters

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getEnrollmentNumber() {

return enrollmentNumber;

}

public void setEnrollmentNumber(String enrollmentNumber) {

this.enrollmentNumber = enrollmentNumber;

}

public int getSemester() {

return semester;

}

public void setSemester(int semester) {

this.semester = semester;

}

public double getSpi() {

return spi;

}

public void setSpi(double spi) {

this.spi = spi;

}

}

**HibernateUtil.java**

package com.student.util;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory sessionFactory = buildSessionFactory();

private static SessionFactory buildSessionFactory() {

try {

return new Configuration().configure().buildSessionFactory();

} catch (Throwable ex) {

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

}

**hibernate.cfg.xml**

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/student\_management</property>

<property name="hibernate.connection.username">your\_username</property>

<property name="hibernate.connection.password">your\_password</property>

<property name="hibernate.hbm2ddl.auto">update</property>

<mapping class="com.student.model.Student"/>

</session-factory>

</hibernate-configuration>

**index.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="com.student.dao.StudentDAO" %>

<%@ page import="com.student.model.Student" %>

<%@ page import="java.util.List" %>

<%

StudentDAO studentDAO = new StudentDAO();

// Handle adding a new student

if (request.getMethod().equalsIgnoreCase("POST")) {

String enrollmentNumber = request.getParameter("enrollmentNumber");

int semester = Integer.parseInt(request.getParameter("semester"));

double spi = Double.parseDouble(request.getParameter("spi"));

Student newStudent = new Student();

newStudent.setEnrollmentNumber(enrollmentNumber);

newStudent.setSemester(semester);

newStudent.setSpi(spi);

studentDAO.addStudent(newStudent);

}

// Retrieve all students

List<Student> studentList = studentDAO.getAllStudents();

%>

<!DOCTYPE html>

<html>

<head>

<title>Student Registration</title>

</head>

<body>

<h1>Student Registration</h1>

<form method="post">

Enrollment Number: <input type="text" name="enrollmentNumber" required/><br>

Semester: <input type="number" name="semester" required/><br>

SPI: <input type="number" step="0.01" name="spi" required/><br>

<input type="submit" value="Add Student" />

</form>

<h2>Registered Students</h2>

<table border="1">

<tr>

<th>Student ID</th>

<th>Enrollment Number</th>

<th>Semester</th>

<th>SPI</th>

</tr>

<%

for (Student s : studentList) {

%>

<tr>

<td><%= s.getId() %></td>

<td><%= s.getEnrollmentNumber() %></td>

<td><%= s.getSemester() %></td>

<td><%= s.getSpi() %></td>

</tr>

<%

}

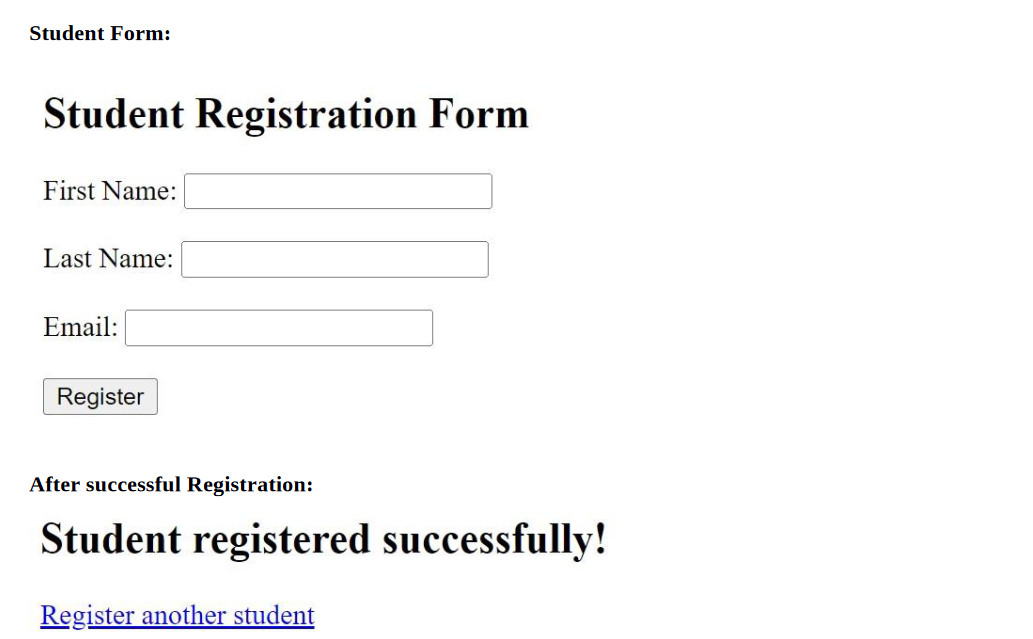
%>

</table>

</body>

</html>

**Output:**

****