**Simplilearn Phase-2 Project**

**Developer Details:**

Developer Name: Vodthala Avinash Goud

Email: [avinashgoud98765@gmail.com](mailto:avinashgoud98765@gmail.com)

Date:4th May 2022

**Objective:**

To build an application where user can track classes, subjects, students, and teachers which has a back-office application with a single administrator login.

**Features:**

Using our application as a user you can add class details, student details, assign teacher and view report.

**User Interaction:**

1. User should login by using credentials provided

2. By selecting various options user will get the following options:

* Add Class
* Add Student
* Add Subject
* Add Teacher
* Assign Teacher
* Assign Subject

3. By selecting the option Admin can do all the operations and save details by clicking on submit button.

**Sprint Planning:**

Number of Sprints Planned: 3

Duration of 1 sprint = 7 days

The following are the sprint details:

1. Sprint 1:

* Collected requirements
* prepared flow chart diagram.
* Added Developer details and Main Menu

1. Sprint 2: Added Business level Operations
2. Sprint 3: Testing the Application by performing business level operations.

**Java Concepts used:**

Servlets, Hibernate

**Tools used:**

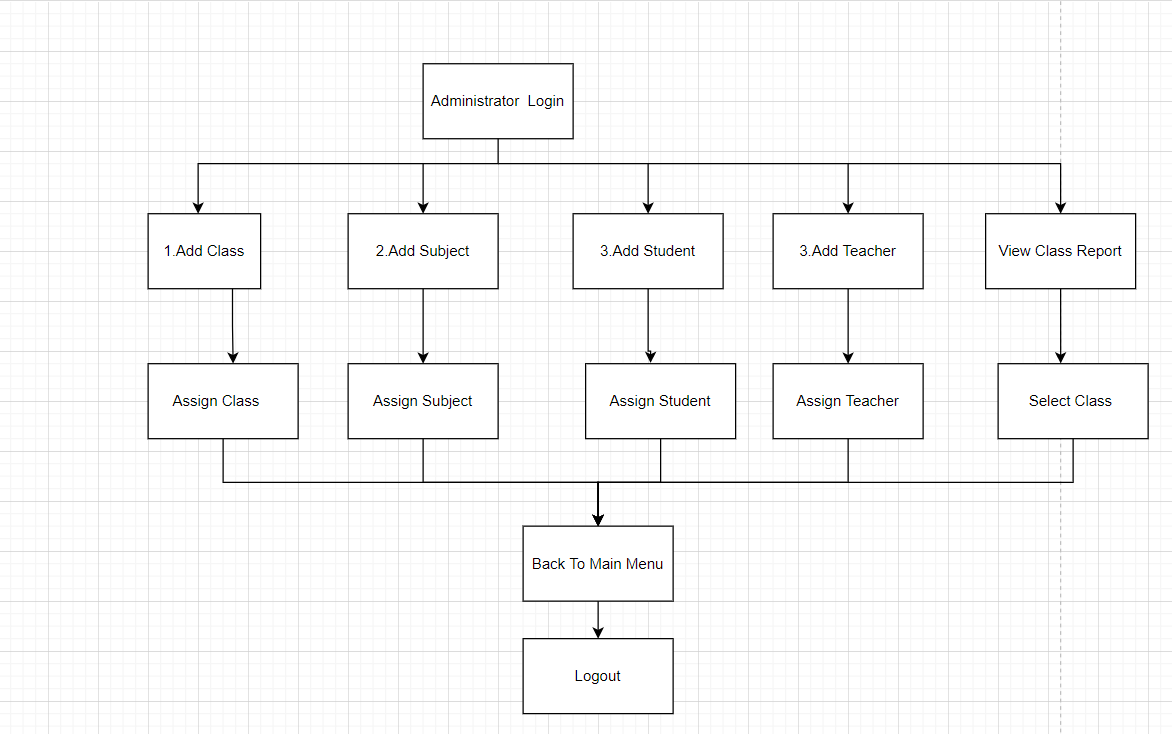
* Eclipse
* Git hub
* MySQL

**Programming language:**

* Java

**Diagrams:**

**1.Flow Chart:**



**Source code:**

**Login:**

package com.simplilearn.login;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class LoginServlet

\*/

@WebServlet("/login")

public class Login extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Login() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

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

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

if((username!=null && username.equals("Avinash")) &&

(password!=null && password.equals("12345"))){

HttpSession session = request.getSession();

session.setAttribute("isLoggedIn", true);

session.setAttribute("username", username);

response.sendRedirect("index.html");

}

PrintWriter out = response.getWriter();

out.print("<html> <body>");

out.print("Welcome to Administrative Portal..");

out.print("<form action='login' method='POST'>");

out.print("Username: <input type=\"text\" name=\"username\"/>");

out.print("<br> Password: <input type=\"text\" name=\"password\"/>");

out.print("<button type=\"submit\"> Login </button>");

out.print("</form>");

out.print("</body></html>");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Classes:**

package com.simplilearn.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

import java.util.List;

import javax.persistence.CascadeType;

@Entity

@Table(name="class\_table")

public class Classes {

@Id

@GeneratedValue

private int class\_id;

@Column(name="class\_name")

private String class\_name;

@OneToMany(cascade= CascadeType.ALL, mappedBy="classes")

private List<Student> students;

@OneToMany(cascade= CascadeType.ALL, mappedBy="classes")

private List<Subject> subjects;

public List<Student> getStudents() {

return students;

}

public void setStudents(List<Student> students) {

this.students = students;

}

public List<Subject> getSubjects() {

return subjects;

}

public void setSubjects(List<Subject> subjects) {

this.subjects = subjects;

}

public int getClass\_id() {

return class\_id;

}

public void setClass\_id(int class\_id) {

this.class\_id = class\_id;

}

public String getClass\_name() {

return class\_name;

}

public void setClass\_name(String class\_name) {

this.class\_name = class\_name;

}

}

**Students:**

package com.simplilearn.entity;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name="student\_table")

public class Student {

@Id

@GeneratedValue

private int student\_id;

@Column(name="name")

private String name;

@Column(name="fname")

private String fname;

@ManyToOne(cascade=CascadeType.ALL)

@JoinColumn(name="class\_id")

private Classes classes;

public int getStudent\_id() {

return student\_id;

}

public void setStudent\_id(int student\_id) {

this.student\_id = student\_id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public Classes getClasses() {

return classes;

}

public void setClasses(Classes classes) {

this.classes = classes;

}

}

**Subjects:**

package com.simplilearn.entity;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name="subject\_table")

public class Subject {

@Id

@GeneratedValue

private int subject\_id;

@Column(name="subject\_name")

private String name;

@ManyToOne(cascade= CascadeType.ALL)

@JoinColumn(name="teacher\_id")

private Teacher teacher;

@ManyToOne(cascade=CascadeType.ALL)

@JoinColumn(name="class\_id")

private Classes classes;

public Classes getClasses() {

return classes;

}

public void setClasses(Classes classes) {

this.classes = classes;

}

public int getSubject\_id() {

return subject\_id;

}

public void setSubject\_id(int subject\_id) {

this.subject\_id = subject\_id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Teacher getTeacher() {

return teacher;

}

public void setTeacher(Teacher teacher) {

this.teacher = teacher;

}

}

**Teachers:**

package com.simplilearn.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name="teacher\_table")

public class Teacher {

@Id

@GeneratedValue

private int teacher\_id;

@Column(name="first\_name")

private String name;

@Column(name="last\_name")

private String lname;

@OneToMany(cascade=CascadeType.ALL, mappedBy="teacher")

private List<Subject> subjects;

public int getTeacher\_id() {

return teacher\_id;

}

public void setTeacher\_id(int teacher\_id) {

this.teacher\_id = teacher\_id;

}

public String getName() {

return name;

}

public void setName(String fname) {

this.name = fname;

}

public String getLname() {

return lname;

}

public void setLname(String lname) {

this.lname = lname;

}

public List<Subject> getSubjects() {

return subjects;

}

public void setSubjects(List<Subject> subjects) {

this.subjects = subjects;

}

}

**Logout:**

package com.simplilearn.login;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class LogoutServlet

\*/

@WebServlet("/logout")

public class Logout extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Logout() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

HttpSession session = request.getSession(false);

if(session!=null) {

if(session.getAttribute("username")!=null) {

session.removeAttribute("username");

session.removeAttribute("isLoggedIn");

//response.sendRedirect("login");

}

}

response.sendRedirect("login");

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**Servlet Codes:**

**AddClasses:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.simplilearn.entity.Classes;

import com.simplilearn.entity.Subject;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AddClass

\*/

@WebServlet("/AddClass")

public class AddClass extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddClass() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

String name = request.getParameter("class\_name");

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

//Step4; Create persistent object and add Teacher

Classes classes = new Classes();

classes.setClass\_name(name);

session.save(classes);

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewClass.jsp");

dispatcher.forward(request, response);

}

}

**AddStudents:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.simplilearn.entity.Student;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AddStudent

\*/

@WebServlet("/addStudents")

public class AddStudent extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddStudent() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

String name = request.getParameter("fname");

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

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

//Step4; Create persistent object and add Teacher

Student student = new Student();

student.setName(name);

student.setFname(lname);

session.save(student);

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewStudent.jsp");

dispatcher.forward(request, response);

}

}

**AssignStudents:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.query.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.simplilearn.entity.Classes;

import com.simplilearn.entity.Student;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AssignStudent

\*/

@WebServlet("/assignStudent")

public class AssignStudent extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AssignStudent() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

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

String[] nameList = name.split(" ");

//System.out.println("NameList: " + nameList);

String classes = request.getParameter("class");

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

String hql\_classes= "from Classes where name='" + classes + "'";

List<Classes> clas = session.createQuery(hql\_classes).list();

String hql\_student = "update Student s set s.classes=:n where s.name=:sn and s.fname=:fn";

Query<Student> query = session.createQuery(hql\_student);

query.setParameter("n", clas.get(0));

query.setParameter("sn", nameList[0]);

query.setParameter("fn", nameList[1]);

query.executeUpdate();

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewStudent.jsp");

dispatcher.forward(request, response);

}

}

**AddSubject:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.simplilearn.entity.Subject;

import com.simplilearn.entity.Teacher;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AddSubject

\*/

@WebServlet("/addSubjects")

public class AddSubject extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddSubject() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

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

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

//Step4; Create persistent object and add Teacher

Subject subject = new Subject();

subject.setName(name);

session.save(subject);

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewSubject.jsp");

dispatcher.forward(request, response);

}

}

**AssignSubject:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import com.simplilearn.entity.Classes;

import com.simplilearn.entity.Subject;

import com.simplilearn.entity.Teacher;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AssignSubject

\*/

@WebServlet("/assignSubject")

public class AssignSubject extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AssignSubject() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

@SuppressWarnings("unchecked")

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

String clas = request.getParameter("class");

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

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

String hql\_clas= "FROM Classes where class\_name='" + clas + "'";

List<Classes> classes = session.createQuery(hql\_clas).list();

// if(classes.isEmpty()) {

// System.out.println("Classes set is emptuy");

// }

String hql\_subject = "update Subject s set s.classes=:n where s.name=:sn";

Query<Subject> query = session.createQuery(hql\_subject);

query.setParameter("n", classes.get(0));

query.setParameter("sn", subject);

query.executeUpdate();

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewClass.jsp");

dispatcher.forward(request, response);

}

}

**AddTeacher:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.simplilearn.entity.Teacher;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AddTeacher

\*/

@WebServlet("/addTeachers")

public class AddTeacher extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddTeacher() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

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

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

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

//Step4; Create persistent object and add Teacher

Teacher teacher = new Teacher();

teacher.setName(name);

teacher.setLname(lname);

session.save(teacher);

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewTeachers.jsp");

dispatcher.forward(request, response);

}

}

**AssignTeacher:**

package com.simplilearn.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import com.simplilearn.entity.Classes;

import com.simplilearn.entity.Student;

import com.simplilearn.entity.Subject;

import com.simplilearn.entity.Teacher;

import com.simplilearn.util.HibernateUtil;

/\*\*

\* Servlet implementation class AssignTeacher

\*/

@WebServlet("/assignTeacher")

public class AssignTeacher extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AssignTeacher() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Step 1: Get details , user has entered

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

String[] nameArray = name.split(" ");

//System.out.println("NameList: " + nameList);

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

// Step2: Create session

SessionFactory sf = HibernateUtil.buildSessionFactory();

Session session = sf.openSession();

// Step 3: Begin Transaction

Transaction tx = session.beginTransaction();

String hql\_teacher= "from Teacher where name='" + nameArray[0] + "'" + " and lname='" + nameArray[1] + "'";

List<Teacher> teachers = session.createQuery(hql\_teacher).list();

String hql\_subject = "update Subject s set s.teacher=:n where s.name=:sn";

Query<Subject> query = session.createQuery(hql\_subject);

query.setParameter("n", teachers.get(0));

query.setParameter("sn", subject);

query.executeUpdate();

// STep5: Commit transaction and close sessoin

tx.commit();

session.close();

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/viewTeachers.jsp");

dispatcher.forward(request, response);

}

}

MySQL Code:

Enter password: \*\*\*\*

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 28

Server version: 5.5.25a MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use simplilearn;

Database changed

Create table class\_table(class\_id bigint primary key auto\_increment, name varchar(200));

Create table student\_table(student\_id bigint primary key auto\_increment, name varchar(

), fname varchar(200));

Create table subject\_table(subject\_id bigint primary key auto\_increment, subject\_name varchar(200));

Create table teacher\_table(teacher\_id bigint primary key auto\_increment, first\_name varchar(200), last\_name varchar(200));

insert into class\_table(name) values('Java Full Stack');

insert into student\_table(name,fname,class\_id) values('Shyam Singha','Roy',1);

**Git Hub Details:**

Github – It is open source application to create remote repository and store our project details.

Git hub Token: ghp\_QDHuvM7LljbwdHHI1Z4nofs9AWW2De0C0qJu

**Git hub commands used:**

* Git init: To initialize the git repository in local
* Git status: To show the details like commited files, untracked files
* Git add . : To add the untracked files to stages area
* Git commit –m “Added Simple Project”: To commit in local area
* Git log: To display the number of commits you already did
* Git log –oneline : To display commit id which are in local
* git branch –M main:Rename master branch master to main
* Connect locale to remote repository.
* git push origin main / git push -u origin main: To push our commits to remote repository

To connect local and remote repository we can use the below link:

git remote add origin https://ghp\_QDHuvM7LljbwdHHI1Z4nofs9AWW2De0C0qJu@github.com/avinashgithub1/learnersacademy2.git

**The following commands were executed in git bash:**

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git init

Reinitialized existing Git repository in C:/Users/hp/eclipse-workspace/learnersacademy2/.git/

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git status

On branch main

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

.classpath

.project

.settings/

Documentation/

Screenshots/

build/

src/

nothing added to commit but untracked files present (use "git add" to track)

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git add.

git: 'add.' is not a git command. See 'git --help'.

The most similar command is

add

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git add .

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git commit -m "Added Phase2 project"

[main (root-commit) 12bf161] Added Phase2 project

91 files changed, 1758 insertions(+)

create mode 100644 .classpath

create mode 100644 .project

create mode 100644 .settings/.jsdtscope

create mode 100644 .settings/org.eclipse.jdt.core.prefs

create mode 100644 .settings/org.eclipse.wst.common.component

create mode 100644 .settings/org.eclipse.wst.common.project.facet.core.xml

create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container

create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name

create mode 100644 Documentation/phase 2 project final.docx

create mode 100644 Screenshots/1.PNG

create mode 100644 Screenshots/10.PNG

create mode 100644 Screenshots/11.PNG

create mode 100644 Screenshots/12.PNG

create mode 100644 Screenshots/13.PNG

create mode 100644 Screenshots/14.PNG

create mode 100644 Screenshots/2.PNG

create mode 100644 Screenshots/3.PNG

create mode 100644 Screenshots/4.PNG

create mode 100644 Screenshots/5.PNG

create mode 100644 Screenshots/6.PNG

create mode 100644 Screenshots/7.PNG

create mode 100644 Screenshots/8.PNG

create mode 100644 Screenshots/9.PNG

create mode 100644 Screenshots/flow.PNG

create mode 100644 build/classes/com/simplilearn/entity/Classes.class

create mode 100644 build/classes/com/simplilearn/entity/Student.class

create mode 100644 build/classes/com/simplilearn/entity/Subject.class

create mode 100644 build/classes/com/simplilearn/entity/Teacher.class

create mode 100644 build/classes/com/simplilearn/login/Login.class

create mode 100644 build/classes/com/simplilearn/login/Logout.class

create mode 100644 build/classes/com/simplilearn/servlet/AddClass.class

create mode 100644 build/classes/com/simplilearn/servlet/AddStudent.class

create mode 100644 build/classes/com/simplilearn/servlet/AddSubject.class

create mode 100644 build/classes/com/simplilearn/servlet/AddTeacher.class

create mode 100644 build/classes/com/simplilearn/servlet/AssignStudent.class

create mode 100644 build/classes/com/simplilearn/servlet/AssignSubject.class

create mode 100644 build/classes/com/simplilearn/servlet/AssignTeacher.class

create mode 100644 build/classes/com/simplilearn/util/HibernateUtil.class

create mode 100644 build/classes/hibernate.cfg.xml

create mode 100644 src/main/java/com/simplilearn/entity/Classes.java

create mode 100644 src/main/java/com/simplilearn/entity/Student.java

create mode 100644 src/main/java/com/simplilearn/entity/Subject.java

create mode 100644 src/main/java/com/simplilearn/entity/Teacher.java

create mode 100644 src/main/java/com/simplilearn/login/Login.java

create mode 100644 src/main/java/com/simplilearn/login/Logout.java

create mode 100644 src/main/java/com/simplilearn/servlet/AddClass.java

create mode 100644 src/main/java/com/simplilearn/servlet/AddStudent.java

create mode 100644 src/main/java/com/simplilearn/servlet/AddSubject.java

create mode 100644 src/main/java/com/simplilearn/servlet/AddTeacher.java

create mode 100644 src/main/java/com/simplilearn/servlet/AssignStudent.java

create mode 100644 src/main/java/com/simplilearn/servlet/AssignSubject.java

create mode 100644 src/main/java/com/simplilearn/servlet/AssignTeacher.java

create mode 100644 src/main/java/com/simplilearn/util/HibernateUtil.java

create mode 100644 src/main/java/hibernate.cfg.xml

create mode 100644 src/main/webapp/META-INF/MANIFEST.MF

create mode 100644 src/main/webapp/WEB-INF/config.properties

create mode 100644 src/main/webapp/WEB-INF/lib/FastInfoset-1.2.15.jar

create mode 100644 src/main/webapp/WEB-INF/lib/antlr-2.7.7.jar

create mode 100644 src/main/webapp/WEB-INF/lib/byte-buddy-1.12.8.jar

create mode 100644 src/main/webapp/WEB-INF/lib/classmate-1.5.1.jar

create mode 100644 src/main/webapp/WEB-INF/lib/hibernate-commons-annotations-5.1.2.Final.jar

create mode 100644 src/main/webapp/WEB-INF/lib/hibernate-core-5.6.6.Final.jar

create mode 100644 src/main/webapp/WEB-INF/lib/istack-commons-runtime-3.0.7.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jandex-2.4.2.Final.jar

create mode 100644 src/main/webapp/WEB-INF/lib/javax.activation-api-1.2.0.jar

create mode 100644 src/main/webapp/WEB-INF/lib/javax.persistence-api-2.2.jar

create mode 100644 src/main/webapp/WEB-INF/lib/javax.xml.bind.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jaxb-api-2.3.1.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jaxb-osgi.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jaxb-runtime-2.3.1.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jboss-logging-3.4.3.Final.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jboss-transaction-api\_1.2\_spec-1.1.1.Final.jar

create mode 100644 src/main/webapp/WEB-INF/lib/jta-1\_1.jar

create mode 100644 src/main/webapp/WEB-INF/lib/mysql-connector-java-8.0.20.jar

create mode 100644 src/main/webapp/WEB-INF/lib/stax-ex-1.8.jar

create mode 100644 src/main/webapp/WEB-INF/lib/txw2-2.3.1.jar

create mode 100644 src/main/webapp/WEB-INF/web.xml

create mode 100644 src/main/webapp/addClasses.html

create mode 100644 src/main/webapp/addStudent.html

create mode 100644 src/main/webapp/addSubject.html

create mode 100644 src/main/webapp/addTeacher.html

create mode 100644 src/main/webapp/assignStudentToClass.jsp

create mode 100644 src/main/webapp/assignSubjectToClass.jsp

create mode 100644 src/main/webapp/assignTeacherToSubject.jsp

create mode 100644 src/main/webapp/chooseClassReport.jsp

create mode 100644 src/main/webapp/classReport.jsp

create mode 100644 src/main/webapp/index.html

create mode 100644 src/main/webapp/viewClass.jsp

create mode 100644 src/main/webapp/viewStudent.jsp

create mode 100644 src/main/webapp/viewSubject.jsp

create mode 100644 src/main/webapp/viewTeachers.jsp

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git log -oneline

fatal: unrecognized argument: -oneline

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git log

commit 12bf161d9d89fa0ca93859a4761d3ba10d9e3060 (HEAD -> main)

Author: avinashv <avinash@gmail.com>

Date: Fri Jun 24 00:30:48 2022 +0530

Added Phase2 project

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git push -u origin main

fatal: 'origin' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git push -u origin main

fatal: 'origin' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

hp@Vyshu MINGW64 ~/eclipse-workspace/learnersacademy2 (main)

$ git remote add origin https://github.com/avinashgithub1/learnersacademy2.git

**Conclusion:** By using this application we can add/assign/view details of classes, subjects, students and teachers. We can enhance this application further by adding functionalities like Deleting details. We can design web pages to this application.