SIMPLILEARN

ASSESSMENT -2

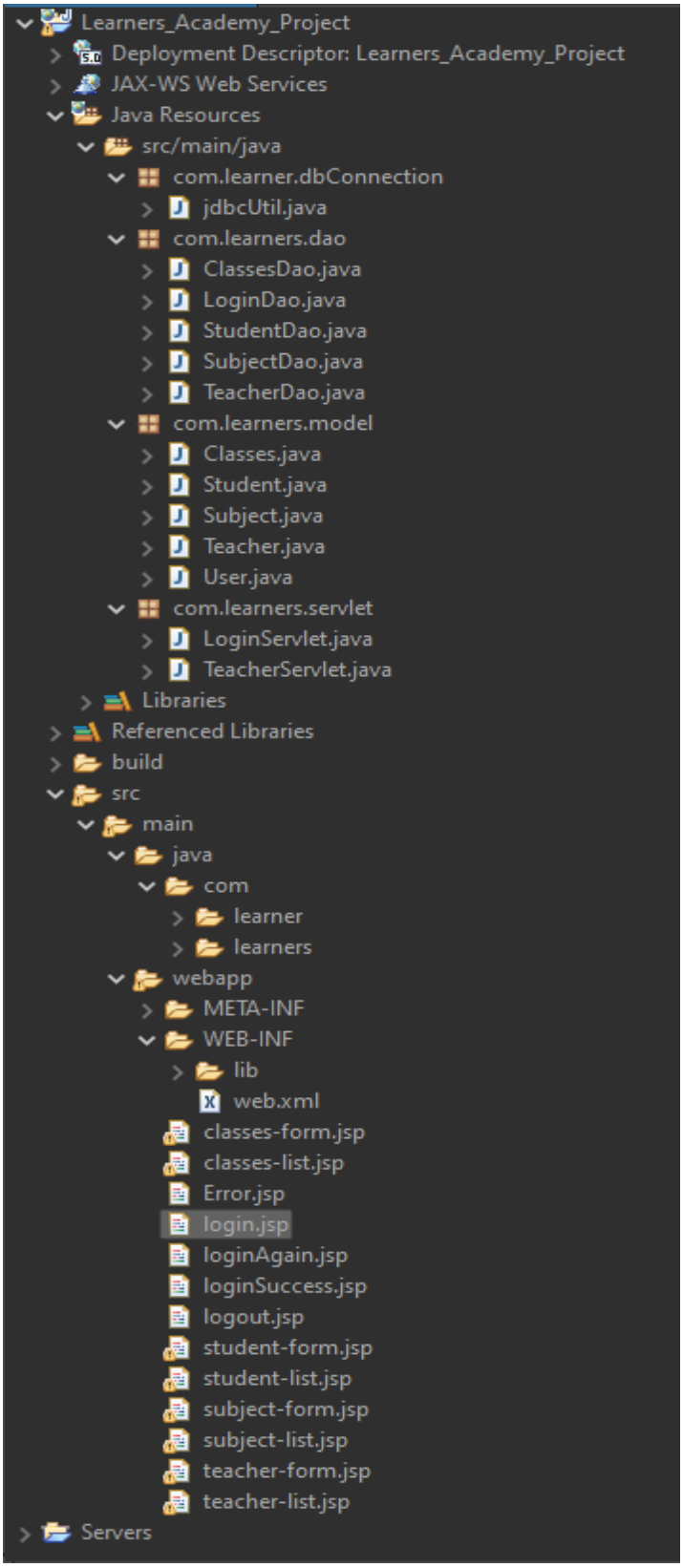
LEARNERS ACADEMY

In this file you can see the source code of Learners Academy which is written in java with Eclipse IDE.

Dipak Sinh

Source Code

- PROJECT STRUCTURE



- ## JDBC.UTIL.JAVA

```
package com.learner.dbConnection;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class jdbcUtil {
	public static String dbUrl = "jdbc:mysql://localhost:3306/learner?useSSL=false";
	public static String dbUname = "root";
	public static String dbPassword = "root12345";
	public static String dbDriver = "com.mysql.cj.jdbc.Driver";

	public static Connection getConnection() {
		System.out.println("Calling getConnection");
		Connection conn = null;
		try {
			Class.forName(dbDriver);
		} catch (ClassNotFoundException e1) {
			e1.printStackTrace();
		}
		try {

			conn = DriverManager.getConnection(dbUrl, dbUname, dbPassword);
		} catch (SQLException e) {

			e.printStackTrace();
		}
		System.out.println(conn);
		return conn;
	}
}
```

- ## USER.JAVA

```
package com.learners.model;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="user_data")
public class User implements Serializable {
	private static final long serialVersionUID = 1L;
```

```
	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	@Column(name="user_name")
	private String username;

	@Column(name="user_pwd")
	private String password;

	public String getUsername() {
		return username;
	}

	public void setUsername(String username) {
		this.username = username;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}

	public User(String username, String password) {
		super();
		this.username = username;
		this.password = password;
	}

	public User() {
		super();

	}

	@Override
	public String toString() {
		return "User [username=" + username + ", password=" + password + "]";
	}
}
```

- TEACHER.JAVA

```
package com.learners.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "teachers")
public class Teacher {
	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	@Column(name = "teacher_id")
	private int teacher_id;

	@Column(name = "teacher_name")
	private String teacher_name;

	@Column(name = "teacher_email")
	private String teacher_email;

	@Column(name = "teacher_class")
	private int teacher_class;

	public Teacher() {
		super();
	}

	public Teacher(String teacher_name, String teacher_email, int teacher_class) {
		super();
		this.teacher_name = teacher_name;
		this.teacher_email = teacher_email;
		this.teacher_class = teacher_class;
	}

	public Teacher(int teacher_id, String teacher_name, String teacher_email, int teacher_class) {
		super();
		this.teacher_id = teacher_id;
		this.teacher_name = teacher_name;
		this.teacher_email = teacher_email;
		this.teacher_class = teacher_class;
	}

	public int getTeacher_id() {
		return teacher_id;
	}

	public void setTeacher_id(int teacher_id) {
		this.teacher_id = teacher_id;
	}

	public String getTeacher_name() {
		return teacher_name;
	}

	public void setTeacher_name(String teacher_name) {
```

```
		this.teacher_name = teacher_name;
	}

	public String getTeacher_email() {
		return teacher_email;
	}

	public void setTeacher_email(String teacher_email) {
		this.teacher_email = teacher_email;
	}

	public int getTeacher_class() {
		return teacher_class;
	}

	public void setTeacher_class(int teacher_class) {
		this.teacher_class = teacher_class;
	}
}
```

- STUDENT.JAVA

```
package com.learners.model;

import javax.persistence.Column;
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.AUTO)
	@Column(name = "student_id")
	private int student_id;

	@Column(name = "student_name")
	private String student_name;

	@Column(name = "class_id")
	private int class_id;

	@Column(name = "student_address")
	private String student_address;

	@Column(name = "student_mobile")
```

```
	private String student_mobile;

	public Student() {
		super();
	}

	public Student(String student_name, int class_id, String student_address, String student_mobile) {
		super();
		this.student_name = student_name;
		this.class_id = class_id;
		this.student_address = student_address;
		this.student_mobile = student_mobile;
	}

	public Student(int student_id, String student_name, int class_id, String student_address, String
student_mobile) {
		super();
		this.student_id = student_id;
		this.student_name = student_name;
		this.class_id = class_id;
		this.student_address = student_address;
		this.student_mobile = student_mobile;
	}

	public int getStudent_id() {
		return student_id;
	}

	public void setStudent_id(int student_id) {
		this.student_id = student_id;
	}

	public String getStudent_name() {
		return student_name;
	}

	public void setStudent_name(String student_name) {
		this.student_name = student_name;
	}

	public int getClass_id() {
		return class_id;
	}

	public void setClass_name(int class_id) {
		this.class_id = class_id;
	}

	public String getStudent_address() {
		return student_address;
	}
```

```
	public void setStudent_address(String student_address) {
		this.student_address = student_address;
	}

	public String getStudent_mobile() {
		return student_mobile;
	}

	public void setStudent_mobile(String student_mobile) {
		this.student_mobile = student_mobile;
	}
}
```

- SUBJECT.JAVA

```
package com.learners.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="subjects")
public class Subject {
	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	@Column(name="subject_id")
	private int subject_id;

	@Column(name="teacher_id")
	private int teacher_id;

	@Column(name="subject_name")
	private String subject_name;


	public Subject() {
		super();
		// TODO Auto-generated constructor stub
	}

	public Subject(int teacher_id, String subject_name) {
		super();
		this.teacher_id = teacher_id;
		this.subject_name = subject_name;
```

```
	}

	public Subject(int subject_id, int teacher_id, String subject_name) {
		super();
		this.subject_id = subject_id;
		this.teacher_id = teacher_id;
		this.subject_name = subject_name;
	}

	public int getSubject_id() {
		return subject_id;
	}

	public void setSubject_id(int subject_id) {
		this.subject_id = subject_id;
	}

	public int getTeacher_id() {
		return teacher_id;
	}

	public void setTeacher_id(int teacher_id) {
		this.teacher_id = teacher_id;
	}

	public String getSubject_name() {
		return subject_name;
	}

	public void setSubject_name(String subject_name) {
		this.subject_name = subject_name;
	}
}
```

- CLASSES.JAVA

```
package com.learners.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="classes")
public class Classes {
```

```
	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	@Column(name="class_id")
	private int class_id;

	@Column(name="class_name")
	private String class_name;

	public Classes() {
		super();
	}

	public Classes(String class_name) {
		super();
		this.class_name = class_name;
	}

	public Classes(int class_id, String class_name) {
		super();
		this.class_id = class_id;
		this.class_name = class_name;
	}

	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;
	}
}
```

- TEACHERDAO.JAVA

```
package com.learners.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

```
import com.learner.dbConnection.jdbcUtil;
import com.learners.model.Teacher;

//Provide operation for teacher
public class TeacherDao {
	Connection conn=null;

	private static final String INSERT_TEACHERS = "INSERT INTO teachers "
			+ "(teacher_name,teacher_email,teacher_class) values" + "(?,?,?);";

	private static final String SELECT_TEACHER_BY_ID = "SELECT
teacher_id,teacher_name,teacher_email,teacher_class "
			+ "from teachers where teacher_id=?;";
	private static final String SELECT_ALL_TEACHERS = "SELECT * FROM teachers;";
	private static final String DELETE_TEACHERS = "DELETE FROM teachers where teacher_id=?;";
	private static final String UPDATE_TEACHERS = "UPDATE teachers SET
teacher_name=?,teacher_email=?,teacher_class=? WHERE teacher_id=?;";

	// Insert teachers
	public void insertTeacher(Teacher teacher) throws Exception {

		conn=jdbcUtil.getConnection();
		PreparedStatement ps = conn.prepareStatement(INSERT_TEACHERS);
		ps.setString(1, teacher.getTeacher_name());
		ps.setString(2, teacher.getTeacher_email());
		ps.setInt(3, teacher.getTeacher_class());
		ps.executeUpdate();
		System.out.println("Data inserted");
		conn.close();
	}

	// Update teachers
	public boolean updateTeacher(Teacher teacher) throws Exception {
		boolean rowUpdated;
		conn=jdbcUtil.getConnection();
		PreparedStatement pstmnt = conn.prepareStatement(UPDATE_TEACHERS);
		pstmnt.setString(1, teacher.getTeacher_name());
		pstmnt.setString(2, teacher.getTeacher_email());
		pstmnt.setInt(3, teacher.getTeacher_class());
		pstmnt.setInt(4, teacher.getTeacher_id());
		rowUpdated = pstmnt.executeUpdate() > 0;
		System.out.println("Data updated");
		conn.close();
		return rowUpdated;

	}

	// Select teachers by id
	public Teacher selectTeacher(int teacher_id) throws SQLException {
		Teacher teacher = null;
		conn=jdbcUtil.getConnection();
		PreparedStatement stmnt = conn.prepareStatement(SELECT_TEACHER_BY_ID);
```

```
                stmnt.setInt(1, teacher_id);
                System.out.println(stmnt);

                ResultSet rs = stmnt.executeQuery();
                while (rs.next()) {
                        String teacher_name = rs.getString("teacher_name");
                        String teacher_email = rs.getString("teacher_email");
                        int teacher_class = rs.getInt("teacher_class");

                        System.out.println(teacher_name + teacher_email + teacher_class);


                        teacher = new Teacher(teacher_id, teacher_name, teacher_email, teacher_class);
                }
                conn.close();
                return teacher;
        }

        // select all teachers
        public List<Teacher> selectAllTeachers() throws SQLException,Exception {
                List<Teacher> teacher = new ArrayList<Teacher>();
                conn=jdbcUtil.getConnection();
                PreparedStatement stmnt = conn.prepareStatement(SELECT_ALL_TEACHERS);

                ResultSet rs = stmnt.executeQuery();
                while (rs.next()) {
                        int teacher_id = rs.getInt("teacher_id");
                        String teacher_name = rs.getString("teacher_name");
                        String teacher_email = rs.getString("teacher_email");
                        int teacher_class =rs.getInt("teacher_class");

                        System.out.println(teacher_id + teacher_name + teacher_email + teacher_class +" -
"+"This data from selectAllTeacher");
                        teacher.add(new Teacher(teacher_id,teacher_name,teacher_email,teacher_class));
                        System.out.println(teacher);
                }
                conn.close();
                return teacher;
        }

        // delete teachers
        public boolean deleteTeacher(int teacher_id) throws SQLException {
                boolean rowDeleted;
                conn=jdbcUtil.getConnection();
                PreparedStatement stmnt = conn.prepareStatement(DELETE_TEACHERS);
                stmnt.setInt(1, teacher_id);
                rowDeleted = stmnt.executeUpdate() > 0;
                conn.close();
                return rowDeleted;

        }
}
```

- SUBJECTDAO.JAVA

```
package com.learners.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.learner.dbConnection.jdbcUtil;
import com.learners.model.Subject;

public class SubjectDao {

	Connection conn = null;

	private static final String INSERT_SUBJECT = "INSERT INTO subjects " +
"(teacher_id,subject_name) values"
			+ "(?,?);";

	private static final String SELECT_SUBJECT_BY_ID = "SELECT subject_id,teacher_id,subject_name
"
			+ "from subjects where subject_id=?;";
	private static final String SELECT_ALL_SUBJECTS = "SELECT * FROM subjects;";
	private static final String DELETE_SUBJECT = "DELETE FROM subjects where subject_id=?;";
	private static final String UPDATE_SUBJECT = "UPDATE subjects SET
teacher_id=?,subject_name=? WHERE subject_id=?;";

	// Insert Subject
	public void insertSubject(Subject subject) throws Exception {

		conn = jdbcUtil.getConnection();
		System.out.println("Insert query passed from insertSubject");
		PreparedStatement ps = conn.prepareStatement(INSERT_SUBJECT);
		ps.setInt(1, subject.getTeacher_id());
		ps.setString(2, subject.getSubject_name());
		ps.executeUpdate();
		System.out.println("Data inserted");
		conn.close();
	}

	// Update Subject
	public boolean updateSubject(Subject subject) throws Exception {
		boolean rowUpdated;
		conn = jdbcUtil.getConnection();
		PreparedStatement ps = conn.prepareStatement(UPDATE_SUBJECT);
		ps.setInt(1, subject.getTeacher_id());
		ps.setString(2, subject.getSubject_name());
		ps.setInt(3, subject.getSubject_id());
```

```
            rowUpdated = ps.executeUpdate() > 0;
            System.out.println("Data updated");
            conn.close();
            return rowUpdated;
        }

        // Select Subject by id
        public Subject selectSubject(int subject_id) throws SQLException {
            Subject subject = null;
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(SELECT_SUBJECT_BY_ID);
            stmnt.setInt(1, subject_id);
            System.out.println(stmnt);

            ResultSet rs = stmnt.executeQuery();
            while (rs.next()) {
                int teacher_id = rs.getInt("teacher_id");
                String subject_name = rs.getString("subject_name");
                System.out.println(teacher_id + subject_name);
                subject = new Subject(subject_id, teacher_id, subject_name);
            }
            conn.close();
            return subject;
        }

        // select all Students
        public List<Subject> selectAllSubjects() throws SQLException, Exception {
            List<Subject> subject = new ArrayList<Subject>();
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(SELECT_ALL_SUBJECTS);

            ResultSet rs = stmnt.executeQuery();
            while (rs.next()) {
                int subject_id = rs.getInt("subject_id");
                int teacher_id = rs.getInt("teacher_id");
                String subject_name = rs.getString("subject_name");

                System.out.println(subject_id + teacher_id + subject_name + " - " + "This data from
selectAllStudents");
                subject.add(new Subject(subject_id, teacher_id, subject_name));
                System.out.println(subject);
            }
            conn.close();
            return subject;
        }

        // delete Subject
        public boolean deleteSubject(int subject_id) throws SQLException {
            boolean rowDeleted;
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(DELETE_SUBJECT);
            stmnt.setInt(1, subject_id);
```

```
		rowDeleted = stmnt.executeUpdate() > 0;
		conn.close();
		return rowDeleted;
	}
}
```

- ## StudentDao.java

```
package com.learners.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.learner.dbConnection.jdbcUtil;
import com.learners.model.Student;

public class StudentDao {
	Connection conn = null;

	private static final String INSERT_STUDENT = "INSERT INTO students "
			+ "(student_name,class_id,student_address,student_mobile) values" + "(?,?,?,?);";

	private static final String SELECT_STUDENT_BY_ID = "SELECT
student_id,student_name,class_id,student_address,student_mobile "
			+ "from students where student_id=?;";
	private static final String SELECT_ALL_STUDENTS = "SELECT * FROM students;";
	private static final String DELETE_STUDENT = "DELETE FROM students where student_id=?;";
	private static final String UPDATE_STUDENT = "UPDATE students SET
student_name=?,class_id=?,student_address=?,student_mobile=? WHERE student_id=?;";

	// Insert Student
	public void insertStudent(Student student) throws Exception {

		conn = jdbcUtil.getConnection();
		System.out.println("Insert query passed from insertStudent");
		PreparedStatement ps = conn.prepareStatement(INSERT_STUDENT);
		ps.setString(1, student.getStudent_name());
		ps.setInt(2, student.getClass_id());
		ps.setString(3, student.getStudent_address());
		ps.setString(4, student.getStudent_mobile());
		ps.executeUpdate();
		System.out.println("Data inserted");
		conn.close();
	}

	// Update Student
	public boolean updateStudent(Student student) throws Exception {
```

```
            boolean rowUpdated;
            conn = jdbcUtil.getConnection();
            PreparedStatement ps = conn.prepareStatement(UPDATE_STUDENT);
            ps.setString(1, student.getStudent_name());
            ps.setInt(2, student.getClass_id());
            ps.setString(3, student.getStudent_address());
            ps.setString(4, student.getStudent_mobile());
            ps.setInt(5, student.getStudent_id());
            rowUpdated = ps.executeUpdate() > 0;
            System.out.println("Data updated");
            conn.close();
            return rowUpdated;
        }

        // Select Student by id
        public Student selectStudent(int student_id) throws SQLException {
            Student student = null;
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(SELECT_STUDENT_BY_ID);
            stmnt.setInt(1, student_id);
            System.out.println(stmnt);

            ResultSet rs = stmnt.executeQuery();
            while (rs.next()) {
                String student_name = rs.getString("student_name");
                int class_id = rs.getInt("class_id");
                String student_address = rs.getString("student_address");
                String student_mobile = rs.getString("student_mobile");

                System.out.println(student_name + class_id + student_address + student_mobile);

                student = new Student(student_id, student_name, class_id, student_address,
student_mobile);
            }
            conn.close();
            return student;

        }

        // select all Students
        public List<Student> selectAllStudents() throws SQLException, Exception {
            List<Student> student = new ArrayList<Student>();
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(SELECT_ALL_STUDENTS);

            ResultSet rs = stmnt.executeQuery();
            while (rs.next()) {
                int student_id = rs.getInt("student_id");
                String student_name = rs.getString("student_name");
                int class_id = rs.getInt("class_id");
                String student_address = rs.getString("student_address");
                String student_mobile = rs.getString("student_mobile");
```

```
                System.out.println(student_id + student_name + class_id + student_address +
student_mobile + " - "
                                + "This data from selectAllStudents");
                student.add(new Student(student_id, student_name, class_id, student_address,
student_mobile));
                System.out.println(student);
            }
            conn.close();
            return student;
        }

        // delete Students
        public boolean deleteStudent(int student_id) throws SQLException {
            boolean rowDeleted;
            conn = jdbcUtil.getConnection();
            PreparedStatement stmnt = conn.prepareStatement(DELETE_STUDENT);
            stmnt.setInt(1, student_id);
            rowDeleted = stmnt.executeUpdate() > 0;
            conn.close();
            return rowDeleted;
        }
}
```

- LOGINDAO.JAVA

```
package com.learners.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import com.learner.dbConnection.jdbcUtil;
import com.learners.model.User;

public class LoginDao {

        Connection conn = null;

        public boolean validate(User user) {

            boolean status = false;
            conn = jdbcUtil.getConnection();
            String sqlquery = "select * from user_data where user_name=? and user_pwd=?";
            PreparedStatement ps;
            ResultSet rs;
            try {
                ps = conn.prepareStatement(sqlquery);
                ps.setString(1, user.getUsername());
                ps.setString(2, user.getPassword());
```

```
                rs = ps.executeQuery();
                status = rs.next();
                conn.close();

            } catch (SQLException e) {
                e.printStackTrace();
            }
            return status;
        }
}
```

- CLASSESDAO.JAVA

```
package com.learners.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import com.learner.dbConnection.jdbcUtil;
import com.learners.model.Classes;

public class ClassesDao {
        Connection conn = null;

        private static final String INSERT_CLASSES = "INSERT INTO classes (class_name) values(?);";
        private static final String SELECT_ALL_CLASSES = "SELECT * FROM classes;";

        //Insert Classes
        public void insertClass(Classes classes) throws Exception {
            conn = jdbcUtil.getConnection();
            System.out.println("Insert query passed from insertClasses");
            PreparedStatement ps = conn.prepareStatement(INSERT_CLASSES);
            ps.setString(1, classes.getClass_name());
            ps.executeUpdate();
            System.out.println("Data inserted");
            conn.close();
        }

        // select all Classes
            public List<Classes> selectAllClasses() throws SQLException, Exception {
                List<Classes> classes = new ArrayList<Classes>();
                conn = jdbcUtil.getConnection();
                PreparedStatement stmnt = conn.prepareStatement(SELECT_ALL_CLASSES);

                ResultSet rs = stmnt.executeQuery();
                while (rs.next()) {
                    int class_id = rs.getInt("class_id");
```

```
                String class_name = rs.getString("class_name");
                System.out.println(class_id + class_name +" - "       + "This data from
selectAllClasses");
                classes.add(new Classes(class_id, class_name));
                System.out.println(classes);
            }
            conn.close();
            return classes;
        }
}
```

- ## LOGINSERVLET.JAVA

```
package com.learners.servlet;

import java.io.IOException;

import com.learners.dao.LoginDao;
import com.learners.model.User;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public LoginServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        System.out.println("Username :- " + username);
        System.out.println("Password :- " + password);

        User user = new User();
        user.setUsername(username);
        user.setPassword(password);

        LoginDao logindao = new LoginDao();
        if (logindao.validate(user)==true) {
```

```
                response.sendRedirect("loginSuccess.jsp");
            } else {
                response.sendRedirect("loginAgain.jsp");
            }
        }

}
```

- ## TEACHERSERVLET.JAVA

```
package com.learners.servlet;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.sql.SQLException;
import java.util.List;

import com.learners.dao.ClassesDao;
import com.learners.dao.StudentDao;
import com.learners.dao.SubjectDao;
import com.learners.dao.TeacherDao;
import com.learners.model.Classes;
import com.learners.model.Student;
import com.learners.model.Subject;
import com.learners.model.Teacher;

/**
 * Servlet implementation class TeacherServlet
 */
@WebServlet("/")
public class TeacherServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;
        private TeacherDao teacherDAO;
        private StudentDao studentDAO;
        private ClassesDao classesDAO;
        private SubjectDao subjectDAO;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public TeacherServlet() {
            this.teacherDAO = new TeacherDao();
            this.studentDAO = new StudentDao();
            this.classesDAO = new ClassesDao();
            this.subjectDAO = new SubjectDao();
```

```
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
 *      response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
        this.doGet(request, response);
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
 *      response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException {
        String action = request.getServletPath();
        System.out.println(action);
        switch (action) {
        case "/newTeacher":
                showNewForm(request, response);
                break;
        case "/insertTeacher":
                insertTeacher(request, response);
                break;
        case "/deleteTeacher":
                deleteTeacher(request, response);
                break;
        case "/editTeacher":
                showEditForm(request, response);
                break;
        case "/updateTeacher":
                updateTeacher(request, response);
                break;
        case "/listTeacher":
                listTeacher(request, response);
                break;
        case "/newStudent":
                showNewStudentForm(request, response);
                break;
        case "/insertStudent":
                insertStudent(request, response);
                break;
        case "/deleteStudent":
                deleteStudent(request, response);
                break;
        case "/editStudent":
                showStudentEditForm(request, response);
                break;
        case "/updateStudent":
                updateStudent(request, response);
```

```
                break;
            case "/listStudent":
                listStudent(request, response);
                break;
            case "/newClasses":
                showNewClassForm(request, response);
                break;
            case "/insertClasses":
                insertClasses(request, response);
                break;
            case "/listClasses":
                listClasses(request, response);
                break;
            case "/newSubject":
                showNewSubjectForm(request, response);
                break;
            case "/insertSubject":
                insertSubject(request, response);
                break;
            case "/deleteSubject":
                deleteSubject(request, response);
                break;
            case "/editSubject":
                showEditSubjectForm(request, response);
                break;
            case "/updateSubject":
                updateSubject(request, response);
                break;
            case "/listSubject":
                listSubject(request, response);
                break;
            default:
                defaultLogin(request, response);
                break;
            }
        }

        private void listSubject(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            List<Subject> listSubject = null;
            try {
                listSubject=subjectDAO.selectAllSubjects();
            } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            request.setAttribute("listSubject", listSubject);
            System.out.println(listSubject + " " + "List Subject method in servlet\n");
            RequestDispatcher rd = request.getRequestDispatcher("subject-list.jsp");
            System.out.println("forward to subject-list.jsp");
            rd.forward(request, response);
```

```
                System.out.println("Forwarded to subject-list.jsp");
        }

        private void updateSubject(HttpServletRequest request, HttpServletResponse response) throws
IOException {
                int subject_id = Integer.parseInt(request.getParameter("subject_id"));
                int teacher_id = Integer.parseInt(request.getParameter("teacher_id"));
                String subject_name = request.getParameter("subject_name");

                System.out.println(subject_id + teacher_id + subject_name);
                Subject subject = new Subject(subject_id, teacher_id, subject_name);
                try {
                        subjectDAO.updateSubject(subject);
                } catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                response.sendRedirect("listSubject");
        }

        private void showEditSubjectForm(HttpServletRequest request, HttpServletResponse response)
                        throws ServletException, IOException {
                System.out.println("Called showEditSubjectForm");
                String data = request.getParameter("subject_id");
                System.out.println(data);
                int subject_id = Integer.parseInt(request.getParameter("subject_id"));
                System.out.println(subject_id + "Calling edit subject form from servlet");
                try {
                        Subject existingSubject = subjectDAO.selectSubject(subject_id);
                        RequestDispatcher rd = request.getRequestDispatcher("subject-form.jsp");
                        request.setAttribute("subject", existingSubject);

                        System.out.println(existingSubject);
                        rd.forward(request, response);
                } catch (SQLException e) {
                        e.printStackTrace();
                }

        }

        private void deleteSubject(HttpServletRequest request, HttpServletResponse response) throws
IOException {
                int subject_id = Integer.parseInt(request.getParameter("subject_id"));
                try {
                        subjectDAO.deleteSubject(subject_id);
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                System.out.println(subject_id);
                response.sendRedirect("listSubject");
```

```
	}

	private void insertSubject(HttpServletRequest request, HttpServletResponse response) throws
IOException {
		int teacher_id = Integer.parseInt(request.getParameter("teacher_id"));
		String subject_name = request.getParameter("subject_name");

		System.out.println(teacher_id + subject_name);
		Subject newSubject = new Subject(teacher_id, subject_name);
		try {
			subjectDAO.insertSubject(newSubject);
		} catch (Exception e) {
			e.printStackTrace();
		}
		response.sendRedirect("listSubject");
	}

	private void showNewSubjectForm(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {
		RequestDispatcher rd = request.getRequestDispatcher("subject-form.jsp");
		System.out.println("Forwarded to subject-form.jsp");
		rd.forward(request, response);

	}

	private void listClasses(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {
		List<Classes> listClasses = null;
		try {
			listClasses = classesDAO.selectAllClasses();
		} catch (Exception e) {
			e.printStackTrace();
		}

		request.setAttribute("listClasses", listClasses);
		System.out.println(listClasses + " " + "List Student method in servlet\n");
		RequestDispatcher rd = request.getRequestDispatcher("classes-list.jsp");
		System.out.println("forward to classes-list.jsp");
		rd.forward(request, response);
		System.out.println("Forwarded to classes-list.jsp");
	}

	private void insertClasses(HttpServletRequest request, HttpServletResponse response) throws
IOException {
		String class_name = request.getParameter("class_name");

		System.out.println(class_name);
		Classes newClasses = new Classes(class_name);
		try {
			classesDAO.insertClass(newClasses);
		} catch (Exception e) {
			e.printStackTrace();
```

```
            }
            response.sendRedirect("listClasses");

        }

        private void showNewClassForm(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            RequestDispatcher rd = request.getRequestDispatcher("classes-form.jsp");
            System.out.println("Forwarded to classes-form.jsp");
            rd.forward(request, response);

        }

        private void defaultLogin(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            RequestDispatcher rd = request.getRequestDispatcher("loginSuccess.jsp");
            System.out.println("Forwarded to Login-success.jsp");
            rd.forward(request, response);
        }

        private void showNewForm(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            RequestDispatcher rd = request.getRequestDispatcher("teacher-form.jsp");
            System.out.println("Forwarded to teacher-form.jsp");
            rd.forward(request, response);
        }

        private void insertTeacher(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            String teacher_name = request.getParameter("teacher_name");
            String teacher_email = request.getParameter("teacher_email");
            int teacher_class = Integer.parseInt(request.getParameter("teacher_class"));

            System.out.println(teacher_name + teacher_email + teacher_class);
            Teacher newTeacher = new Teacher(teacher_name, teacher_email, teacher_class);
            try {
                    teacherDAO.insertTeacher(newTeacher);
            } catch (Exception e) {
                    e.printStackTrace();
            }
            response.sendRedirect("listTeacher");
        }

        private void deleteTeacher(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            int teacher_id = Integer.parseInt(request.getParameter("teacher_id"));
            try {
                    teacherDAO.deleteTeacher(teacher_id);
            } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
```

```
		System.out.println(teacher_id);
		response.sendRedirect("listTeacher");
	}

	private void showEditForm(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {
		System.out.println("Called showEditForm");
		String data = request.getParameter("teacher_id");
		System.out.println(data);
		int teacher_id = Integer.parseInt(request.getParameter("teacher_id"));
		System.out.println(teacher_id + "Calling edit form from servlet");
		try {
			Teacher exestingTeacher = teacherDAO.selectTeacher(teacher_id);
			RequestDispatcher rd = request.getRequestDispatcher("teacher-form.jsp");
			request.setAttribute("teacher", exestingTeacher);

			System.out.println(exestingTeacher);
			rd.forward(request, response);
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	private void updateTeacher(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {
		int teacher_id = Integer.parseInt(request.getParameter("teacher_id"));
		String teacher_name = request.getParameter("teacher_name");
		String teacher_email = request.getParameter("teacher_email");
		int teacher_class = Integer.parseInt(request.getParameter("teacher_class"));

		System.out.println(teacher_id + teacher_name + teacher_email + teacher_class);

		Teacher teacher = new Teacher(teacher_id, teacher_name, teacher_email, teacher_class);
		try {
			teacherDAO.updateTeacher(teacher);
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		response.sendRedirect("listTeacher");

	}

	private void listTeacher(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {
		List<Teacher> listTeacher = null;
		try {
			listTeacher = teacherDAO.selectAllTeachers();
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
```

```
            request.setAttribute("listTeacher", listTeacher);
            System.out.println(listTeacher + " " + "List Teacher method in servlet\n");
            RequestDispatcher rd = request.getRequestDispatcher("teacher-list.jsp");
            System.out.println("forward to teacher-list.jsp");
            rd.forward(request, response);
            System.out.println("Forwarded to teacher-list.jsp");
        }

        private void showNewStudentForm(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            RequestDispatcher rd = request.getRequestDispatcher("student-form.jsp");
            System.out.println("Forwarded to student-form.jsp");
            rd.forward(request, response);
        }

        private void insertStudent(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            String student_name = request.getParameter("student_name");
            int class_id = Integer.parseInt(request.getParameter("class_id"));
            String student_address = request.getParameter("student_address");
            String student_mobile = request.getParameter("student_mobile");

            System.out.println(student_name + class_id + student_address + student_mobile);
            Student newStudent = new Student(student_name, class_id, student_address,
student_mobile);
            try {
                    studentDAO.insertStudent(newStudent);
            } catch (Exception e) {
                    e.printStackTrace();
            }
            response.sendRedirect("listStudent");
        }

        private void deleteStudent(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            int student_id = Integer.parseInt(request.getParameter("student_id"));
            try {
                    studentDAO.deleteStudent(student_id);
            } catch (SQLException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
            System.out.println(student_id);
            response.sendRedirect("listStudent");
        }

        private void showStudentEditForm(HttpServletRequest request, HttpServletResponse response)
                    throws ServletException, IOException {
            System.out.println("Called showStudentEditForm");
            String data = request.getParameter("student_id");
            System.out.println(data);
            int student_id = Integer.parseInt(request.getParameter("student_id"));
```

```
			System.out.println(student_id + "-" + "Calling student edit form from servlet");
			try {
				Student exestingStudent = studentDAO.selectStudent(student_id);
				RequestDispatcher rd = request.getRequestDispatcher("student-form.jsp");
				request.setAttribute("student", exestingStudent);
				System.out.println(exestingStudent);
				rd.forward(request, response);
			} catch (SQLException e) {
				e.printStackTrace();
			}
		}

		private void updateStudent(HttpServletRequest request, HttpServletResponse response)
				throws ServletException, IOException {

			System.out.println("Entered into updateStudent method");
			int student_id = Integer.parseInt(request.getParameter("student_id"));
			System.out.println(student_id);
			String student_name = request.getParameter("student_name");
			int class_id = Integer.parseInt(request.getParameter("class_id"));
			String student_address = request.getParameter("student_address");
			String student_mobile = request.getParameter("student_mobile");

			System.out.println(student_id + student_name + class_id + student_address +
student_mobile);
			Student student = new Student(student_id, student_name, class_id, student_address,
student_mobile);

			try {
				studentDAO.updateStudent(student);
			} catch (Exception e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
			response.sendRedirect("listStudent");

		}

		private void listStudent(HttpServletRequest request, HttpServletResponse response)
				throws ServletException, IOException {
			List<Student> listStudent = null;
			try {
				listStudent = studentDAO.selectAllStudents();
			} catch (Exception e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}

			request.setAttribute("listStudent", listStudent);
			System.out.println(listStudent + " " + "List Student method in servlet\n");
			RequestDispatcher rd = request.getRequestDispatcher("student-list.jsp");
			System.out.println("forward to student-list.jsp");
```

```
            rd.forward(request, response);
            System.out.println("Forwarded to student-list.jsp");
        }

}
```