The Learner’s Academy

**DEVELOPER NAME : ANKIT YADAV**

**USED TECHNOLOGIES & TOOLS :**

**FRONT END : HTML CSS**

**BACK END :**

**JAVA**

**MySQL**

**SERVLET**

**JSP**

**IDE : ECLIPSE.**

PROJECT DETAILS

This project aims to design and develop a backend administrative portal for the Learner’s Academy using Java EE technologies. I developed it as a project of phase 2 for the Become a back-end expert course. The goal of this project is to apply servlet, jsp and JDBC concepts.

Product Backlog:

1. Create database and tables.
2. Connect the database to the project.
3. Create models classes.
4. Create a database utility class to retrieve data.
5. Create login page.
6. Create JSP files for all pages of the project.
7. Create a servlet to get requests and send responses to the JSP files.
8. Add cookies.
9. Create a CSS file to format the contents.
10. Debug and Test the project.

Technologies and tools Used

• Servlet: to do the business logic and works a controller for the project.

• JSP: to handle the presentation view.

• SQL: to create and manage the database.

• JDBC: to make operations on the database for the project.

• CSS: to format the contents.

• MYSQL workbench: to administrate and manage the database manually.

• Eclipse: to write and run the code.

• Tomcat: to run and deploy servlet application.



**Flow chart**



Login Page

Core concepts used in the project.

• Object-Oriented: used to create and model objects for users and their credentials.

• Databases: used to store and retrieve data.

• Data Sources: used to define a set of properties required to identify and access the database.

• Collections: used some collections such array list to store collection of data.

• Exception Handling: used to catch problems that arises in the code especially in I/O blocks.

• Cookies: to store log-in data on the client browser.

Source code

**Dbutil file** :-

package com.dbutil;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class OracleConnection {

private static Connection connection;

public static Connection getConnection() throws ClassNotFoundException, SQLException {

Class.forName("oracle.jdbc.driver.OracleDriver");

connection=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","learnersacademy","2102");

return connection;

}

}

**Exception file** :-

package com.exception;

@SuppressWarnings("serial")

public class BusinessException extends Exception {

public BusinessException() {

super();

// TODO Auto-generated constructor stub

}

public BusinessException(final String message) {

super(message);

// TODO Auto-generated constructor stub

}

}

package com.learnersacademy.bo;

import java.util.List;

import com.exception.BusinessException;

import com.learnersacademy.model.ClassObj;

import com.learnersacademy.model.StudentObj;

import com.learnersacademy.model.SubjectObj;

import com.learnersacademy.model.TeacherObj;

public interface LearnersAcadBO {

public boolean userLogin(String username, String password) throws BusinessException;

public List<SubjectObj> listSubject() throws BusinessException;

public List<ClassObj> listClasses() throws BusinessException;

public List<TeacherObj> listTeacher() throws BusinessException;

public List<StudentObj> listStudents() throws BusinessException;

public boolean deleteSubject(Integer subjectID) throws BusinessException;

public boolean deleteClass(Integer classID) throws BusinessException;

public boolean deleteTeacher(Integer teacherID) throws BusinessException;

public boolean deleteStudent(Integer studentID) throws BusinessException;

public boolean addSubject(SubjectObj subject) throws BusinessException;

public boolean addClass(ClassObj cls) throws BusinessException;

public boolean addTeacher(TeacherObj teacher) throws BusinessException;

public boolean addStudent(StudentObj student) throws BusinessException;

public boolean updSubject(SubjectObj subject) throws BusinessException;

public boolean updTeacher(TeacherObj teacher) throws BusinessException;

public boolean updStudent(StudentObj student) throws BusinessException;

public List<Object> generateReport(ClassObj cls) throws BusinessException;

}

package com.learnersacademy.bo;

import java.util.List;

import com.exception.BusinessException;

import com.learnersacademy.dao.LearnersAcadDAOImpl;

import com.learnersacademy.model.ClassObj;

import com.learnersacademy.model.StudentObj;

import com.learnersacademy.model.SubjectObj;

import com.learnersacademy.model.TeacherObj;

public class LearnersAcadBOImpl implements LearnersAcadBO{

@Override

public boolean userLogin(String username, String password) throws BusinessException {

boolean bool=false;

//System.out.println("user "+username+username.isEmpty()+" pwd "+password+password.isEmpty());

if(!username.isEmpty() && !password.isEmpty()){

if(username.matches("^[A-Za-z0-9.\_]+$") && password.matches("^[A-Za-z0-9]+$")) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

daoObject.userLogin(username, password);

bool=true;

return bool;

}

else {throw new BusinessException("Invalid Credentials");}

}

else {throw new BusinessException("Username or Password Cannot be Null");}

}

@Override

public List<SubjectObj> listSubject() throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

List<SubjectObj> subjectList=daoObject.listSubject();

return subjectList;

}

@Override

public List<ClassObj> listClasses() throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

List<ClassObj> classList=daoObject.listClasses();

return classList;

}

@Override

public List<TeacherObj> listTeacher() throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

List<TeacherObj> teacherList=daoObject.listTeacher();

return teacherList;

}

@Override

public List<StudentObj> listStudents() throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

List<StudentObj> studentList=daoObject.listStudents();

return studentList;

}

@Override

public boolean deleteSubject(Integer subjectID) throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.deleteSubject(subjectID);

}

@Override

public boolean deleteClass(Integer classID) throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.deleteClass(classID);

}

@Override

public boolean deleteTeacher(Integer teacherID) throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.deleteTeacher(teacherID);

}

@Override

public boolean deleteStudent(Integer studentID) throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.deleteStudent(studentID);

}

@Override

public boolean addSubject(SubjectObj subject) throws BusinessException {

if(subject.getSubName().isEmpty()||subject.getSubLang().isEmpty()) {

throw new BusinessException("Subject Name or Language Cannot be Empty");

}

else {

if(subject.getSubName().matches("^[A-Za-z\\S]+$")) {

if(subject.getSubLang().matches("^[A-Za-z]+$")) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.addSubject(subject);

}

else {

throw new BusinessException("Subject Language should only have alphabets, No White Spaces Allowed");

}

}

else {

throw new BusinessException("Subject Name should only have alphabets and spaces");

}

}

}

@Override

public boolean addClass(ClassObj cls) throws BusinessException {

if(cls.getStandard()==null || cls.getDivision().isEmpty()) {

throw new BusinessException("Subject Name or Language Cannot be Empty");

}

else {

if(cls.getStandard()>0||cls.getStandard()<12) {

if(cls.getDivision().matches("[A-D]{1}")) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.addClass(cls);

}

else {

throw new BusinessException("Pls choose a division from A,B,C,D");

}

}

else {

throw new BusinessException("Standard Should be betweeen 1 and 12");

}

}

}

@Override

public boolean addTeacher(TeacherObj teacher) throws BusinessException {

if(teacher.getTeacherName().isEmpty()||teacher.getTeacherCategory().isEmpty()||teacher.getExperience()==null) {

throw new BusinessException("Teacher Name, Category or Experience Cannot be Empty");

}

else {

if(teacher.getTeacherName().matches("^[A-Za-z\\s]+$")) {

if(teacher.getTeacherCategory().matches("^[A-Za-z]+$")) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.addTeacher(teacher);

}

else {

throw new BusinessException("Teacher Category should only have alphabets, No White Spaces Allowed");

}

}

else {

throw new BusinessException("Teacher Name should only have alphabets and spaces");

}

}

}

@Override

public boolean addStudent(StudentObj student) throws BusinessException {

if(student.getStudentName().isEmpty()|| student.getStudentDOB()==null) {

throw new BusinessException("Student Name or DoB Cannot be Empty");

}

else {

if(student.getStudentName().matches("^[A-Za-z\\S]+$")) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.addStudent(student);

}

else {

throw new BusinessException("Teacher Name should only have alphabets and spaces");

}

}

}

@Override

public boolean updSubject(SubjectObj subject) throws BusinessException {

if(subject.getSubName().matches("^[A-Za-z\\s]+$")||subject.getSubName().isEmpty()) {

if(subject.getSubLang().matches("^[A-Za-z]+$")||subject.getSubLang().isEmpty()) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.updSubject(subject);

}

else {

throw new BusinessException("Subject Language should only have alphabets, No White Spaces Allowed");

}

}

else {

throw new BusinessException("Subject Name should only have alphabets and spaces");

}

}

@Override

public boolean updTeacher(TeacherObj teacher) throws BusinessException {

if(teacher.getTeacherName().isEmpty()&&teacher.getTeacherCategory().isEmpty()&&teacher.getExperience()==null) {

throw new BusinessException("Teacher Name, Category and Experience Cannot be Empty. No parameter to be updated");

}

else {

if(teacher.getTeacherName().matches("^[A-Za-z\\s]+$")||teacher.getTeacherName().isEmpty()) {

if(teacher.getTeacherCategory().matches("^[A-Za-z]+$")||teacher.getTeacherCategory().isEmpty()) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.updTeacher(teacher);

}

else {

throw new BusinessException("Teacher Category should only have alphabets, No White Spaces Allowed");

}

}

else {

throw new BusinessException("Teacher Name should only have alphabets and spaces");

}

}

}

@Override

public boolean updStudent(StudentObj student) throws BusinessException {

if(student.getStudentName().isEmpty() && student.getStudentDOB()==null && student.getClassID()==null) {

throw new BusinessException("Student Name, DoB and ClassID all Cannot be Empty. no updation required.");

}

else {

if(student.getStudentName().matches("^[A-Za-z\\s]+$") ||student.getStudentName().isEmpty()) {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.updStudent(student);

}

else {

throw new BusinessException("Student Name should only have alphabets and spaces");

}

}

}

@Override

public List<Object> generateReport(ClassObj cls) throws BusinessException {

LearnersAcadDAOImpl daoObject=new LearnersAcadDAOImpl();

return daoObject.generateReport(cls);

}

}

**DAO file** :-

package com.learnersacademy.dao;

import java.util.List;

import com.exception.BusinessException;

import com.learnersacademy.model.ClassObj;

import com.learnersacademy.model.StudentObj;

import com.learnersacademy.model.SubjectObj;

import com.learnersacademy.model.TeacherObj;

public interface LearnersAcadDAO {

public boolean userLogin(String username, String password) throws BusinessException;

public List<SubjectObj> listSubject() throws BusinessException;

public List<ClassObj> listClasses() throws BusinessException;

public List<TeacherObj> listTeacher() throws BusinessException;

public List<StudentObj> listStudents() throws BusinessException;

public boolean deleteSubject(Integer subjectID) throws BusinessException;

public boolean deleteClass(Integer classID) throws BusinessException;

public boolean deleteTeacher(Integer teacherID) throws BusinessException;

public boolean deleteStudent(Integer studentID) throws BusinessException;

public boolean addSubject(SubjectObj subject) throws BusinessException;

public boolean addClass(ClassObj cls) throws BusinessException;

public boolean addTeacher(TeacherObj teacher) throws BusinessException;

public boolean addStudent(StudentObj student) throws BusinessException;

public boolean updSubject(SubjectObj subject) throws BusinessException;

public boolean updTeacher(TeacherObj teacher) throws BusinessException;

public boolean updStudent(StudentObj student) throws BusinessException;

public List<Object> generateReport(ClassObj cls) throws BusinessException;

}

package com.learnersacademy.dao;

import java.sql.CallableStatement;

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.dbutil.OracleConnection;

import com.exception.BusinessException;

import com.learnersacademy.model.ClassObj;

import com.learnersacademy.model.StudentObj;

import com.learnersacademy.model.SubjectObj;

import com.learnersacademy.model.TeacherObj;

public class LearnersAcadDAOImpl implements LearnersAcadDAO{

@Override

public boolean userLogin(String username, String password) throws BusinessException {

Boolean bool=false;

try(Connection connection=OracleConnection.getConnection())

{

String sql="select 1 from learnersacademy.tab\_user where username=? and password=?";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

System.out.println(password);

ResultSet result=preparedStatement.executeQuery();

if(result.next()) {

bool=true;

return bool;

}

else {

throw new BusinessException("Username and Password dont match");

}

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public List<SubjectObj> listSubject() throws BusinessException {

List<SubjectObj> subList=new ArrayList<SubjectObj>();

try(Connection connection=OracleConnection.getConnection())

{

String sql="select sub\_name,sub\_id,sub\_lang,class\_id,teacher\_id from learnersacademy.tab\_subject";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

ResultSet result=preparedStatement.executeQuery();

while(result.next()) {

SubjectObj subject=new SubjectObj();

System.out.println(result.getInt("sub\_id"));

subject.setClassId(result.getInt("class\_id"));

subject.setSubjectID(result.getInt("sub\_id"));

subject.setSubLang(result.getString("sub\_lang"));

subject.setSubName(result.getString("sub\_name"));

subject.setTeacherId(result.getInt("teacher\_id"));

subList.add(subject);

}

return subList;

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public List<ClassObj> listClasses() throws BusinessException {

List<ClassObj> classList=new ArrayList<ClassObj>();

try(Connection connection=OracleConnection.getConnection())

{

String sql="select class\_id,division,std from learnersacademy.tab\_class";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

ResultSet result=preparedStatement.executeQuery();

while(result.next()) {

ClassObj cls=new ClassObj();

cls.setClassID(result.getInt("class\_id"));

cls.setDivision(result.getString("division"));

cls.setStandard(result.getInt("std"));

classList.add(cls);

}

return classList;

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public List<TeacherObj> listTeacher() throws BusinessException {

List<TeacherObj> teacherList=new ArrayList<TeacherObj>();

try(Connection connection=OracleConnection.getConnection())

{

String sql="select teacher\_id,teacher\_name,teacher\_category,experience from learnersacademy.tab\_teacher";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

ResultSet result=preparedStatement.executeQuery();

while(result.next()) {

TeacherObj teacher=new TeacherObj();

teacher.settID(result.getInt("teacher\_id"));

teacher.setTeacherCategory(result.getString("teacher\_category"));

teacher.setTeacherName(result.getString("teacher\_name"));

teacher.setExperience(result.getInt("experience"));

teacherList.add(teacher);

}

return teacherList;

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public List<StudentObj> listStudents() throws BusinessException {

List<StudentObj> studentList=new ArrayList<StudentObj>();

try(Connection connection=OracleConnection.getConnection())

{

String sql="select student\_id,student\_name,student\_dob,class\_id from learnersacademy.tab\_student";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

ResultSet result=preparedStatement.executeQuery();

while(result.next()) {

StudentObj student=new StudentObj();

student.setStudentId(result.getInt("student\_id"));

student.setClassID(result.getInt("class\_id"));

student.setStudentDOB(result.getDate("student\_dob"));

student.setStudentName(result.getString("student\_name"));

studentList.add(student);

}

return studentList;

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public boolean deleteSubject(Integer subjectID) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="delete from learnersacademy.tab\_subject where sub\_id=?";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setInt(1, subjectID);

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Deletion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

@Override

public boolean deleteClass(Integer classID) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql = "{call la\_update\_record\_pkg.del\_class(?,?,?)}";

CallableStatement callableStatement = connection.prepareCall(sql);

callableStatement.setInt(1, classID);

callableStatement.registerOutParameter(3, java.sql.Types.NUMERIC);

callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

callableStatement.execute();

Integer success\_code=callableStatement.getInt(3);

String err\_msg=callableStatement.getString(2);

if(success\_code!=1) {

throw new BusinessException("Deletion Failed. "+err\_msg);

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean deleteTeacher(Integer teacherID) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql = "{call la\_update\_record\_pkg.del\_teacher(?,?,?)}";

CallableStatement callableStatement = connection.prepareCall(sql);

callableStatement.setInt(1, teacherID);

callableStatement.registerOutParameter(3, java.sql.Types.NUMERIC);

callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

callableStatement.execute();

Integer success\_code=callableStatement.getInt(3);

String err\_msg=callableStatement.getString(2);

if(success\_code!=1) {

throw new BusinessException("Deletion Failed ."+err\_msg);

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean deleteStudent(Integer studentID) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="delete from learnersacademy.tab\_student where student\_id=?";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setInt(1, studentID);

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Deletion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean addSubject(SubjectObj subject) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="insert into learnersacademy.tab\_subject (sub\_name,sub\_lang) values (?,?)";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setString(1, subject.getSubName());

preparedStatement.setString(2, subject.getSubLang());

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Insertion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean addClass(ClassObj cls) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="Insert into learnersacademy.tab\_class (std,division) values(?,?)";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setInt(1, cls.getStandard());

preparedStatement.setString(2, cls.getDivision());

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Insertion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean addTeacher(TeacherObj teacher) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="Insert into learnersacademy.tab\_teacher(teacher\_name,teacher\_category,experience) values(?,?,?)";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setString(1, teacher.getTeacherName());

preparedStatement.setString(2, teacher.getTeacherCategory());

preparedStatement.setInt(3, teacher.getExperience());

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Insertion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean addStudent(StudentObj student) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="Insert into learnersacademy.tab\_student(student\_name,student\_dob) values(?,?)";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setString(1, student.getStudentName());

preparedStatement.setDate(2, student.getStudentDOB());

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Insertion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean updSubject(SubjectObj subject) throws BusinessException {

try (Connection connection = OracleConnection.getConnection()) {

String sql = "{call la\_update\_record\_pkg.upd\_subject(?,?,?,?,?,?,?)}";

CallableStatement callableStatement = connection.prepareCall(sql);

callableStatement.setInt(2, subject.getSubjectID());

callableStatement.setString(1,subject.getSubName());

callableStatement.setString(3,subject.getSubLang());

callableStatement.setInt(4,subject.getClassId());

callableStatement.setInt(5,subject.getTeacherId());

callableStatement.registerOutParameter(7, java.sql.Types.NUMERIC);

callableStatement.registerOutParameter(6, java.sql.Types.VARCHAR);

callableStatement.execute();

Integer success\_code=callableStatement.getInt(7);

String err\_msg=callableStatement.getString(6);

if(success\_code==1) {

return true;

}

else {

throw new BusinessException(err\_msg);

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal error occured kindly contact sysadmin with error message "+e);

}

}

@Override

public boolean updTeacher(TeacherObj teacher) throws BusinessException {

try(Connection connection=OracleConnection.getConnection())

{

String sql="update learnersacademy.tab\_teacher tt set tt.teacher\_name=nvl(?,tt.teacher\_name),"

+ "tt.teacher\_category=nvl(?,tt.teacher\_category),tt.experience=nvl(?,tt.experience) " +

"where tt.teacher\_id=?";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setString(1, teacher.getTeacherName());

preparedStatement.setString(2, teacher.getTeacherCategory());

preparedStatement.setInt(3, teacher.getExperience());

preparedStatement.setInt(4, teacher.gettID());

int c=preparedStatement.executeUpdate();

if(c!=1) {

throw new BusinessException("Insertion Failed");

}

else {

return true;

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} }

@Override

public boolean updStudent(StudentObj student) throws BusinessException {

try (Connection connection = OracleConnection.getConnection()) {

String sql = "{call la\_update\_record\_pkg.upd\_student(?,?,?,?,?,?)}";

CallableStatement callableStatement = connection.prepareCall(sql);

callableStatement.setInt(1, student.getStudentId());

callableStatement.setString(2,student.getStudentName());

callableStatement.setDate(3, student.getStudentDOB() );

callableStatement.setInt(4,student.getClassID());

callableStatement.registerOutParameter(6, java.sql.Types.NUMERIC);

callableStatement.registerOutParameter(5, java.sql.Types.VARCHAR);

callableStatement.execute();

Integer success\_code=callableStatement.getInt(6);

String err\_msg=callableStatement.getString(5);

if(success\_code==1) {

return true;

}

else {

throw new BusinessException(err\_msg);

}

} catch (ClassNotFoundException | SQLException e) {

throw new BusinessException("Internal error occured kindly contact sysadmin with error message "+e);

}

}

@Override

public List<Object> generateReport(ClassObj cls) throws BusinessException {

List<Object> reportList=new ArrayList<Object>();

try(Connection connection=OracleConnection.getConnection())

{

String sql="select sub\_name,sub\_id,sub\_lang,class\_id,teacher\_id from learnersacademy.tab\_subject where class\_id=?";

PreparedStatement preparedStatement=connection.prepareStatement(sql);

preparedStatement.setInt(1, cls.getClassID());

ResultSet resultSubject=preparedStatement.executeQuery();

while(resultSubject.next()) {

SubjectObj subject=new SubjectObj();

subject.setClassId(resultSubject.getInt("class\_id"));

subject.setSubjectID(resultSubject.getInt("sub\_id"));

subject.setSubLang(resultSubject.getString("sub\_lang"));

subject.setSubName(resultSubject.getString("sub\_name"));

subject.setTeacherId(resultSubject.getInt("teacher\_id"));

reportList.add(subject);

}

String sql2="select teacher\_id,teacher\_name,teacher\_category,experience from learnersacademy.tab\_teacher tt where "

+ "tt.teacher\_id in (select ts.teacher\_id from learnersacademy.tab\_subject ts where ts.class\_id=?)";

PreparedStatement preparedStatement2=connection.prepareStatement(sql2);

preparedStatement2.setInt(1, cls.getClassID());

ResultSet resultTeacher=preparedStatement2.executeQuery();

while(resultTeacher.next()) {

TeacherObj teacher=new TeacherObj();

teacher.settID(resultTeacher.getInt("teacher\_id"));

teacher.setTeacherCategory(resultTeacher.getString("teacher\_category"));

teacher.setTeacherName(resultTeacher.getString("teacher\_name"));

teacher.setExperience(resultTeacher.getInt("experience"));

reportList.add(teacher);

}

String sql3="select student\_id,student\_name,student\_dob,class\_id from learnersacademy.tab\_student where class\_id=?";

PreparedStatement preparedStatement3=connection.prepareStatement(sql3);

preparedStatement3.setInt(1, cls.getClassID());

ResultSet resultStudent=preparedStatement3.executeQuery();

while(resultStudent.next()) {

StudentObj student=new StudentObj();

student.setStudentId(resultStudent.getInt("student\_id"));

student.setClassID(resultStudent.getInt("class\_id"));

student.setStudentDOB(resultStudent.getDate("student\_dob"));

student.setStudentName(resultStudent.getString("student\_name"));

reportList.add(student);

}

return reportList;

} catch (ClassNotFoundException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

} catch (SQLException e) {

throw new BusinessException("Internal Error contact sysadmin with error message "+e);

}

}

}

Model

package com.learnersacademy.model;

public class ClassObj {

private Integer classID;

private Integer standard;

private String Division;

public ClassObj() {}

public Integer getClassID() {

return classID;

}

public void setClassID(Integer classID) {

this.classID = classID;

}

public Integer getStandard() {

return standard;

}

public void setStandard(Integer standard) {

this.standard = standard;

}

public String getDivision() {

return Division;

}

public void setDivision(String division) {

Division = division;

}

@Override

public String toString() {

return "ClassObj [classID=" + classID + ", standard=" + standard + ", Division=" + Division + "]";

}

}

**Student object** :-

package com.learnersacademy.model;

import java.sql.Date;

public class StudentObj {

private Integer studentId;

private String studentName;

private Date studentDOB;

private Integer classID;

public StudentObj() {}

public Integer getStudentId() {

return studentId;

}

public void setStudentId(Integer studentId) {

this.studentId = studentId;

}

public String getStudentName() {

return studentName;

}

public void setStudentName(String studentName) {

this.studentName = studentName;

}

public Date getStudentDOB() {

return studentDOB;

}

public void setStudentDOB(Date studentDOB) {

this.studentDOB = studentDOB;

}

public Integer getClassID() {

return classID;

}

public void setClassID(Integer classID) {

this.classID = classID;

}

*@*Override

public String toString() {

return "StudentObj [studentId=" + studentId + ", studentName=" + studentName + ", studentDOB=" + studentDOB

+ ", classID=" + classID + "]";

}

}

package com.learnersacademy.model;

public class SubjectObj {

private Integer subjectID;

private String subName;

private String subLang;

private Integer classId;

private Integer teacherId;

public SubjectObj() {}

public Integer getSubjectID() {

return subjectID;

}

public void setSubjectID(Integer subjectID) {

this.subjectID = subjectID;

}

public String getSubName() {

return subName;

}

public void setSubName(String subName) {

this.subName = subName;

}

public String getSubLang() {

return subLang;

}

public void setSubLang(String subLang) {

this.subLang = subLang;

}

public Integer getClassId() {

return classId;

}

public void setClassId(Integer classId) {

this.classId = classId;

}

public Integer getTeacherId() {

return teacherId;

}

public void setTeacherId(Integer teacherId) {

this.teacherId = teacherId;

}

@Override

public String toString() {

return "SubjectObj [subjectID=" + subjectID + ", subName=" + subName + ", subLang=" + subLang + ", classId="

+ classId + ", teacherId=" + teacherId + "]";

}

}

Teacher object:-

package com.learnersacademy.model;

public class TeacherObj {

private Integer tID;

private String teacherName;

private String teacherCategory;

private Integer experience;

public TeacherObj() {}

public Integer gettID() {

return tID;

}

public void settID(Integer tID) {

this.tID = tID;

}

public String getTeacherName() {

return teacherName;

}

public void setTeacherName(String teacherName) {

this.teacherName = teacherName;

}

public String getTeacherCategory() {

return teacherCategory;

}

public void setTeacherCategory(String teacherCategory) {

this.teacherCategory = teacherCategory;

}

public Integer getExperience() {

return experience;

}

public void setExperience(Integer experience) {

this.experience = experience;

}

@Override

public String toString() {

return "TeacherObj [tID=" + tID + ", teacherName=" + teacherName + ", teacherCategory=" + teacherCategory

+ ", experience=" + experience + "]";

}

}

Login controller :\_

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

/\*\*

\* Servlet implementation class LoginController

\*/

public class LoginController extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public LoginController() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

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

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

PrintWriter out=response.getWriter();

RequestDispatcher rd = null;

LearnersAcadBOImpl boObject=new LearnersAcadBOImpl();

try {

if(boObject.userLogin(username, password)) {

HttpSession session=request.getSession();

session.setAttribute("username", username);

session.setMaxInactiveInterval(60);

response.sendRedirect("Dashboard.jsp");

// rd=request.getRequestDispatcher("Dashboard.jsp");

// rd.forward(request, response);

}

} catch (BusinessException e) {

rd=request.getRequestDispatcher("login.html");

//out.print(username+" "+password);

rd.include(request, response);

out.print("<center><span style='color:red'>"+e.getMessage()+"</span></center>");

}

}}

**Logout controller** :-

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class LogoutController

\*/

public class LogoutController extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public LogoutController() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

PrintWriter out=response.getWriter();

HttpSession session=request.getSession(false);

RequestDispatcher rd=null;

if(session==null) {

rd=request.getRequestDispatcher("/login.html");

rd.include(request, response);

out.print("<center><span style='color:red'>Please login first.</span></center>");

}

else {

session.invalidate();

out.print("<center><h3>You have been logged out successfully.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Manage classes** :-

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.ClassObj;

/\*\*

\* Servlet implementation class ManageClasses

\*/

public class ManageClasses extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ManageClasses() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

if(session!=null) {

try {

List<ClassObj> classList=boObj.listClasses();

session.setAttribute("classList", classList);

response.sendRedirect("viewClass.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "Dashboard.jsp");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "Dashboard.jsp");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

if(session!=null) {

try {

Integer classId = Integer.parseInt(request.getParameter("classId\_delete"));

boObj.deleteClass(classId);

List<ClassObj> classList=boObj.listClasses();

session.setAttribute("classList", classList);

response.sendRedirect("viewClass.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "classes");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "classes");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Manage student :-**

**package com.learnersacademy.servlet;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.util.List;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**import com.exception.BusinessException;**

**import com.learnersacademy.bo.LearnersAcadBOImpl;**

**import com.learnersacademy.model.StudentObj;**

**/\*\***

**\* Servlet implementation class ManageStudent**

**\*/**

**public class ManageStudent extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**/\*\***

**\* @see HttpServlet#HttpServlet()**

**\*/**

**public ManageStudent() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**/\*\***

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

**\*/**

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

**response.setContentType("text/html");**

**HttpSession session=request.getSession(false);**

**PrintWriter out=response.getWriter();**

**LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();**

**if(session!=null) {**

**try {**

**List<StudentObj> studentList=boObj.listStudents();**

**session.setAttribute("studentList", studentList);**

**response.sendRedirect("viewStudent.jsp");**

**}catch (BusinessException e) {**

**session.setAttribute("exception", e.getMessage());**

**response.sendRedirect( "Dashboard.jsp");}**

**catch(Exception e1) {**

**session.setAttribute("exceptionSys", e1.getMessage());**

**response.sendRedirect( "Dashboard.jsp");**

**}**

**}**

**else {**

**out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");**

**response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");**

**}**

**}**

**/\*\***

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

**\*/**

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

**response.setContentType("text/html");**

**HttpSession session=request.getSession(false);**

**PrintWriter out=response.getWriter();**

**LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();**

**if(session!=null) {**

**try {**

**Integer studentId = Integer.parseInt(request.getParameter("studentId\_delete"));**

**boObj.deleteStudent(studentId);**

**List<StudentObj> studentList=boObj.listStudents();**

**session.setAttribute("studentList", studentList);**

**response.sendRedirect("viewStudent.jsp");**

**} catch (BusinessException e) {**

**session.setAttribute("exception", e.getMessage());**

**response.sendRedirect( "student");}**

**catch(Exception e1) {**

**session.setAttribute("exceptionSys", e1.getMessage());**

**response.sendRedirect( "student");**

**}**

**}**

**else {**

**out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");**

**response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");**

**} }**

**}**

**Manage subject :-**

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.SubjectObj;

/\*\*

\* Servlet implementation class ManageSubject

\*/

public class ManageSubject extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ManageSubject() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

if(session!=null) {

try {

List<SubjectObj> subList=boObj.listSubject();

session.setAttribute("subList", subList);

response.sendRedirect("viewSubject.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "Dashboard.jsp");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "Dashboard.jsp");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

if(session!=null) {

try {

Integer subjectId = Integer.parseInt(request.getParameter("subjectId\_delete"));

boObj.deleteSubject(subjectId);

List<SubjectObj> subList = boObj.listSubject();

session.setAttribute("subList", subList);

response.sendRedirect("viewSubject.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "subject");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "subject");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Manage Teacher :-**

**package com.learnersacademy.servlet;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.util.List;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**import com.exception.BusinessException;**

**import com.learnersacademy.bo.LearnersAcadBOImpl;**

**import com.learnersacademy.model.TeacherObj;**

**/\*\***

**\* Servlet implementation class ManageTeacher**

**\*/**

**public class ManageTeacher extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**/\*\***

**\* @see HttpServlet#HttpServlet()**

**\*/**

**public ManageTeacher() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**/\*\***

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

**\*/**

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

**response.setContentType("text/html");**

**HttpSession session=request.getSession(false);**

**PrintWriter out=response.getWriter();**

**LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();**

**if(session!=null) {**

**try {**

**List<TeacherObj> teacherList=boObj.listTeacher();**

**session.setAttribute("teacherList", teacherList);**

**response.sendRedirect("viewTeacher.jsp");**

**} catch (BusinessException e) {**

**session.setAttribute("exception", e.getMessage());**

**response.sendRedirect( "Dashboard.jsp");}**

**catch(Exception e1) {**

**session.setAttribute("exceptionSys", e1.getMessage());**

**response.sendRedirect( "Dashboard.jsp");**

**}**

**}**

**else {**

**out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");**

**response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");**

**}**

**}**

**/\*\***

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

**\*/**

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

**response.setContentType("text/html");**

**HttpSession session=request.getSession(false);**

**PrintWriter out=response.getWriter();**

**LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();**

**if(session!=null) {**

**try {**

**Integer teacherId = Integer.parseInt(request.getParameter("teacherId\_delete"));**

**boObj.deleteTeacher(teacherId);**

**List<TeacherObj> teacherList=boObj.listTeacher();**

**session.setAttribute("teacherList", teacherList);**

**response.sendRedirect("viewTeacher.jsp");**

**} catch (BusinessException e) {**

**session.setAttribute("exception", e.getMessage());**

**response.sendRedirect( "teacher");}**

**catch(Exception e1) {**

**session.setAttribute("exceptionSys", e1.getMessage());**

**response.sendRedirect( "teacher");**

**}**

**}**

**else {**

**out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");**

**response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");**

**}**

**}**

**}**

**Update Class:-**

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.ClassObj;

/\*\*

\* Servlet implementation class UpdateClass

\*/

public class UpdateClass extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public UpdateClass() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

ClassObj cls=new ClassObj();

if(session!=null) {

cls.setClassID(Integer.parseInt(request.getParameter("class\_id")));

cls.setStandard(Integer.parseInt(request.getParameter("standard")));

cls.setDivision(request.getParameter("division"));

try {

List<Object> reportList=boObj.generateReport(cls);

session.setAttribute("reportList", reportList);

session.setAttribute("class", cls);

response.sendRedirect("viewReport.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "classes");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "classes");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

ClassObj cls=new ClassObj();

if(session!=null) {

try {

if(!request.getParameter("standard").isEmpty())

{cls.setStandard(Integer.parseInt(request.getParameter("standard")));}

if(!request.getParameter("division").isEmpty()) {

cls.setDivision(request.getParameter("division"));

}

boObj.addClass(cls);

List<ClassObj> classList=boObj.listClasses();

session.setAttribute("classList", classList);

response.sendRedirect("viewClass.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "classes");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "classes");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Update Student :-**

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Date;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.StudentObj;

/\*\*

\* Servlet implementation class UpdateStudent

\*/

public class UpdateStudent extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public UpdateStudent() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

StudentObj student=new StudentObj();

if(session!=null) {

try {

student.setStudentName(request.getParameter("newName"));

if(!request.getParameter("newDOB").isEmpty()) {

Date date=new Date(new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("newDOB")).getTime());

student.setStudentDOB((date));}

student.setStudentId(Integer.parseInt(request.getParameter("id")));

if(!request.getParameter("newClassId").isEmpty()) {

student.setClassID(Integer.parseInt(request.getParameter("newClassId")));}

else {student.setClassID(0);}

boObj.updStudent(student);

List<StudentObj> studentList=boObj.listStudents();

session.setAttribute("studentList", studentList);

response.sendRedirect("viewStudent.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "student");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "student");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

StudentObj student=new StudentObj();

if(session!=null) {

try {

student.setStudentName(request.getParameter("studentName"));

if(!request.getParameter("studentDoB").isEmpty()) {

Date date=new Date(new SimpleDateFormat("yyyy-MM-dd").parse(request.getParameter("studentDoB")).getTime());

student.setStudentDOB((date));}

boObj.addStudent(student);

List<StudentObj> studentList=boObj.listStudents();

session.setAttribute("studentList", studentList);

response.sendRedirect("viewStudent.jsp");

}catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "student");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "student");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Update Subject :-**

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.SubjectObj;

/\*\*

\* Servlet implementation class UpdateSubject

\*/

public class UpdateSubject extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public UpdateSubject() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

SubjectObj subject=new SubjectObj();

if(session!=null) {

try {

subject.setSubjectID(Integer.parseInt(request.getParameter("id")));

subject.setSubName(request.getParameter("newName"));

subject.setSubLang(request.getParameter("newLang"));

if(!request.getParameter("newClassId").isEmpty()) {

subject.setClassId(Integer.parseInt(request.getParameter("newClassId")));}

else {subject.setClassId(0);}

if(!request.getParameter("newTeacherId").isEmpty()) {

subject.setTeacherId(Integer.parseInt(request.getParameter("newTeacherId")));}

else {subject.setTeacherId(0);}

boObj.updSubject(subject);

List<SubjectObj> subList = boObj.listSubject();

session.setAttribute("subList", subList);

response.sendRedirect("viewSubject.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "subject");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "subject");

}}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

SubjectObj subject=new SubjectObj();

if(session!=null) {

try {

subject.setSubName(request.getParameter("subName"));

subject.setSubLang(request.getParameter("subLang"));

boObj.addSubject(subject);

List<SubjectObj> subList = boObj.listSubject();

session.setAttribute("subList", subList);

response.sendRedirect("viewSubject.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "subject");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "subject");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Update Teacher** :-

package com.learnersacademy.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.exception.BusinessException;

import com.learnersacademy.bo.LearnersAcadBOImpl;

import com.learnersacademy.model.TeacherObj;

/\*\*

\* Servlet implementation class UpdateTeacher

\*/

public class UpdateTeacher extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public UpdateTeacher() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

TeacherObj teacher=new TeacherObj();

if(session!=null) {

try {

teacher.settID(Integer.parseInt(request.getParameter("id")));

teacher.setTeacherName(request.getParameter("newName"));

teacher.setTeacherCategory(request.getParameter("newCategory"));

if(request.getParameter("newExp").isEmpty()) {teacher.setExperience(0);}

else {

teacher.setExperience(Integer.parseInt(request.getParameter("newExp")));}

boObj.updTeacher(teacher);

List<TeacherObj> teacherList=boObj.listTeacher();

session.setAttribute("teacherList", teacherList);

response.sendRedirect("viewTeacher.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "teacher");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "teacher");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

/\*\*

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

\*/

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

response.setContentType("text/html");

HttpSession session=request.getSession(false);

PrintWriter out=response.getWriter();

LearnersAcadBOImpl boObj=new LearnersAcadBOImpl();

TeacherObj teacher=new TeacherObj();

if(session!=null) {

try {

teacher.setTeacherName(request.getParameter("teacherName"));

teacher.setTeacherCategory(request.getParameter("teacherCategory"));

if(request.getParameter("experience").isEmpty()) {

teacher.setExperience(0);

}else {

teacher.setExperience(Integer.parseInt(request.getParameter("experience")));}

boObj.addTeacher(teacher);

List<TeacherObj> teacherList=boObj.listTeacher();

session.setAttribute("teacherList", teacherList);

response.sendRedirect("viewTeacher.jsp");

} catch (BusinessException e) {

session.setAttribute("exception", e.getMessage());

response.sendRedirect( "teacher");}

catch(Exception e1) {

session.setAttribute("exceptionSys", e1.getMessage());

response.sendRedirect( "teacher");

}

}

else {

out.print("<center><h3>Your session has expired.. Navigating you to home page.....</h3></center>");

response.setHeader("refresh", "5;url='/SchoolManagmentSystem'");

}

}

}

**Print class:-**

package com.learnersacademy.tagclass;

import java.io.IOException;

import java.util.List;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import com.learnersacademy.model.ClassObj;

public class PrintClass extends SimpleTagSupport{

private List<ClassObj> classList;

public void setClassList(List<ClassObj> classList) {

this.classList = classList;

}

@Override

public void doTag() throws IOException {

JspWriter out = getJspContext().getOut();

String headings[]= {"ID","Standard","Division"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings) {

out.print("<th>" + heading + "</th>");

}

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

for(ClassObj cls:classList)

{

out.print("<tr>");

out.print("<form action='classes' method='post'>");

out.print("<td><input type='hidden' name='classId\_delete' value='" +cls.getClassID() + "'/>"+cls.getClassID()+"</td>");

out.print("<td>"+cls.getStandard()+"</td>");

out.print("<td>"+cls.getDivision()+"</td>");

out.print("<td><input type='submit' value='Delete'/></td>");

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

out.print("<form action='UpdateClass' method='get'>");

out.print("<td>"+"<input type='hidden' name='class\_id' value='"

+cls.getClassID()

+"'/>"+"<input type='hidden' name='division' value='" +cls.getDivision() +"'/>"

+"<input type='hidden' name='standard' value='" +cls.getStandard()

+"'/><input type='submit' value='Generate Report'/></td>");

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

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

}out.print("</table>");

}

}

**Print report:-**

package com.learnersacademy.tagclass;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import com.learnersacademy.model.StudentObj;

import com.learnersacademy.model.SubjectObj;

import com.learnersacademy.model.TeacherObj;

public class PrintReport extends SimpleTagSupport {

private List<Object> reportList;

public void setReportList(List<Object> reportList) {

this.reportList = reportList;

}

@Override

public void doTag() throws IOException {

SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

JspWriter out = getJspContext().getOut();

String headings[]= {"ID","Name","Date of Birth"};

out.print("<div style='text-align:center'>");

out.print("<b>Student Details</b>");

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

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings) {

out.print("<th>" + heading + "</th>");

}

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

for(Object studObj:reportList)

{

if(studObj.getClass()==StudentObj.class) {

StudentObj std=(StudentObj)studObj;

out.print("<tr>");

out.print("<form action='student' method='post'>");

out.print("<td>" +std.getStudentId() + "</td>");

out.print("<td>"+std.getStudentName()+"</td>");

out.print("<td>"+format.format(std.getStudentDOB())+"</td>");

out.print("<tr>");

}

}

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

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

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

out.print("<b>Subject Details</b>");

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

String headings2[]= {"ID","Name","Language","Teacher ID"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings2) {

out.print("<th>" + heading + "</th>");

}

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

for(Object subObj:reportList)

{

if(subObj.getClass()==SubjectObj.class) {

SubjectObj sub=(SubjectObj)subObj;

out.print("<tr>");

out.print("<form action='subject' method='post'>");

out.print("<td>"+sub.getSubjectID() +"</td>");

out.print("<td>"+sub.getSubName()+"</td>");

out.print("<td>"+sub.getSubLang()+"</td>");

out.print("<td>"+sub.getTeacherId()+"</td>");

out.print("<tr>");

}

}

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

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

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

out.print("<b>Teacher Details</b>");

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

String headings3[]= {"ID","Name","Category", "Experience"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings3) {

out.print("<th>" + heading + "</th>");

}

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

for(Object tchObj:reportList)

{

if(tchObj.getClass()==TeacherObj.class) {

TeacherObj tch=(TeacherObj)tchObj;

out.print("<tr>");

out.print("<form action='teacher' method='post'>");

out.print("<td>" +tch.gettID() +"</td>");

out.print("<td>"+tch.getTeacherName()+"</td>");

out.print("<td>"+tch.getTeacherCategory()+"</td>");

out.print("<td>"+tch.getExperience()+"</td>");

out.print("<tr>");

}

}

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

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

}

}

**Print Student:-**

package com.learnersacademy.tagclass;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.List;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import com.learnersacademy.model.StudentObj;

public class PrintStudent extends SimpleTagSupport{

private List<StudentObj> studentList;

public void setStudentList(List<StudentObj> studentList) {

this.studentList = studentList;

}

@Override

public void doTag() throws IOException {

SimpleDateFormat format = new SimpleDateFormat("dd-MM-yyyy");

JspWriter out = getJspContext().getOut();

String headings[]= {"ID","Name","Date of Birth", "Class ID"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>"); out.print("<tr>");

for (String heading : headings) {

out.print("<th>" + heading + "</th>");

}

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

for(StudentObj std:studentList)

{

out.print("<tr>");

out.print("<form action='student' method='post'>");

out.print("<td><input type='hidden' name='studentId\_delete' value='" +std.getStudentId() + "'/>"+std.getStudentId()+"</td>");

out.print("<td>"+std.getStudentName()+"</td>");

out.print("<td>"+format.format(std.getStudentDOB())+"</td>");

out.print("<td>"+std.getClassID()+"</td>");

out.print("<td><input type='submit' value='Delete'/></td>");

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

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

out.print("<tr bgcolor='grey'>");

out.print("<form action='UpdateStudent' method='get'>");

out.print("<td><input type='text' name='id' value='" +std.getStudentId() + "' readonly/></td>");

out.print("<td><input type='text' name='newName' placeholder='New Name'></td>");

out.print("<td><input type='date' name='newDOB' placeholder='dd-mm-yyyy'></td>");

out.print("<td><input type='text' name='newClassId' placeholder='New Class ID'></td>");

out.print("<td><input type='submit' value='Update'/></td>");

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

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

}

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

}

}

**Print Subject :-**

package com.learnersacademy.tagclass;

import java.io.IOException;

import java.util.List;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import com.learnersacademy.model.SubjectObj;

public class PrintSubject extends SimpleTagSupport {

private List<SubjectObj> sublist;

public void setSublist(List<SubjectObj> sublist) {

this.sublist = sublist;

}

@Override

public void doTag() throws IOException {

JspWriter out = getJspContext().getOut();

String headings[]= {"ID","Name","Language","Class ID","Teacher ID"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings) {

out.print("<th>" + heading + "</th>");

}

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

for(SubjectObj sub:sublist)

{

out.print("<tr>");

out.print("<form action='subject' method='post'>");

out.print("<td><input type='hidden' name='subjectId\_delete' value='" + sub.getSubjectID() + "'/>"+sub.getSubjectID()+"</td>");

out.print("<td>"+sub.getSubName()+"</td>");

out.print("<td>"+sub.getSubLang()+"</td>");

out.print("<td>"+sub.getClassId()+"</td>");

out.print("<td>"+sub.getTeacherId()+"</td>");

out.print("<td><input type='submit' value='Delete'/></td>");

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

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

out.print("<tr bgcolor='grey'>");

out.print("<form action='UpdateSubject' method='get'>");

out.print("<td><input type='text' name='id' value='" +sub.getSubjectID() + "' readonly/></td>");

out.print("<td><input type='text' name='newName' placeholder='New Sub Name'></td>");

out.print("<td><input type='text' name='newLang' placeholder='New Language'></td>");

out.print("<td><input type='number' name='newClassId' placeholder='New Class ID'></td>");

out.print("<td><input type='number' name='newTeacherId' placeholder='New Teacher ID'></td>");

out.print("<td><input type='submit' value='Update'/></td>");

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

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

}

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

}

}

**Print Teacher :-**

package com.learnersacademy.tagclass;

import java.io.IOException;

import java.util.List;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.tagext.SimpleTagSupport;

import com.learnersacademy.model.TeacherObj;

public class PrintTeacher extends SimpleTagSupport{

private List<TeacherObj> teacherList;

public void setTeacherList(List<TeacherObj> teacherList) {

this.teacherList = teacherList;

}

@Override

public void doTag() throws IOException{

JspWriter out = getJspContext().getOut();

String headings[]= {"ID","Name","Category", "Experience"};

out.print("<table border='1px' style='text-align:center; margin-left:auto;margin-right:auto;'>");

out.print("<tr>");

for (String heading : headings) {

out.print("<th>" + heading + "</th>");

}

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

for(TeacherObj tch:teacherList)

{

out.print("<tr>");

out.print("<form action='teacher' method='post'>");

out.print("<td><input type='hidden' name='teacherId\_delete' value='" +tch.gettID() + "'/>"+tch.gettID()+"</td>");

out.print("<td>"+tch.getTeacherName()+"</td>");

out.print("<td>"+tch.getTeacherCategory()+"</td>");

out.print("<td>"+tch.getExperience()+"</td>");

out.print("<td><input type='submit' value='Delete'/></td>");

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

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

out.print("<tr bgcolor='grey'>");

out.print("<form action='UpdateTeacher' method='get'>");

out.print("<td><input type='text' name='id' value='" +tch.gettID() + "' readonly/></td>");

out.print("<td><input type='text' name='newName' placeholder='New Teacher Name'></td>");

out.print("<td><input type='text' name='newCategory' placeholder='New Category'></td>");

out.print("<td><input type='text' name='newExp' value='"+tch.getExperience()+"'placeholder='New Experience'></td>");

out.print("<td><input type='submit' value='Update'/></td>");

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

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

}

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

}

}