

Full Name	Karl Joseph Chetcuti
Email	Karlcht@hotmail.com
Batch	Full Stack Developer - December 2021 Cohort
Project Title	Developing a Backend Admin for Learner's Academy.
Git Hub Link:	https://github.com/karlcht/JavaFSDcourse3Project.git
Project Submission Date	23/08/2022

Core Concepts Used in this Sprint

- Creating a dynamic web project
- Use of Servlets and session management commands
- Use of JSP, HTML scripts
- Use of SQL Databases and familiarises with commands
- Use of JDBC to connect and communicate with database
- MVC pattern coding - DAO , Service Layer, Controller , model and view elements

Code

Please refer to the next pages.

Screenshots

Please refer to the next pages.

Flow Chart

Refer to next page.

Enhancing the application

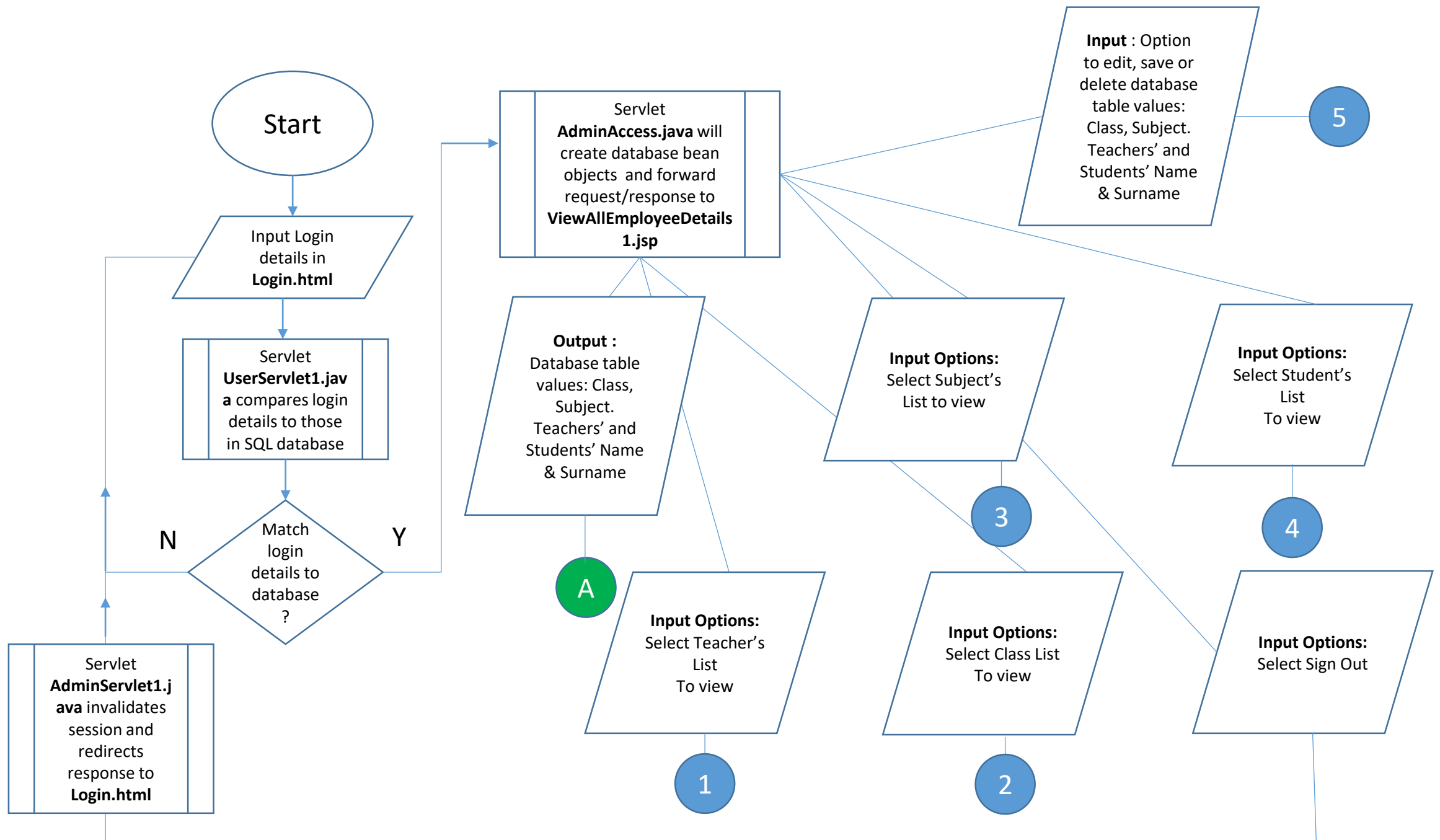
The application can be enhanced as follows:

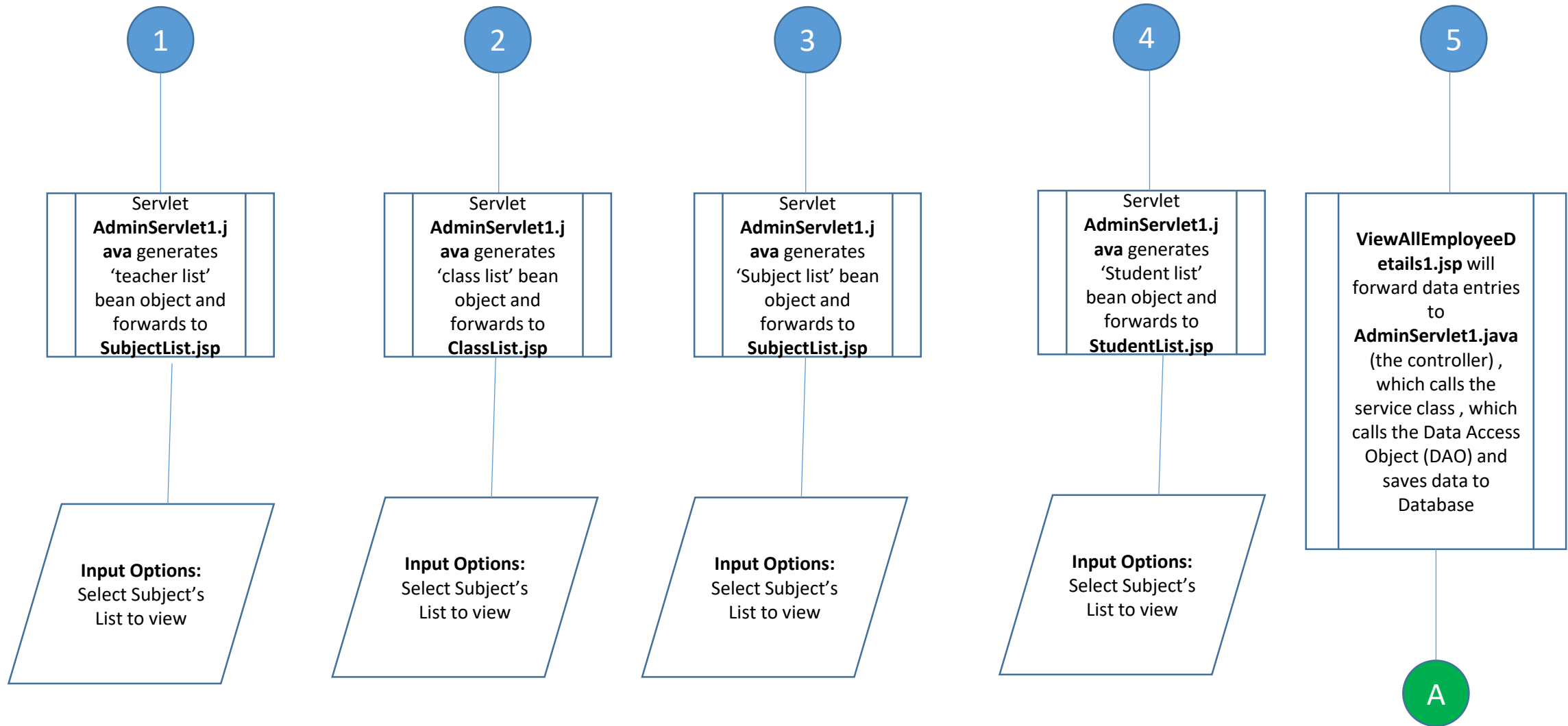
- Better user visuals
- Add more functions – such as sorting of data in alphabetical or numerical order
- Add function – to store session management data and cookies onto databases
- Add function for admin to view user's session actions.

Unique Selling Points

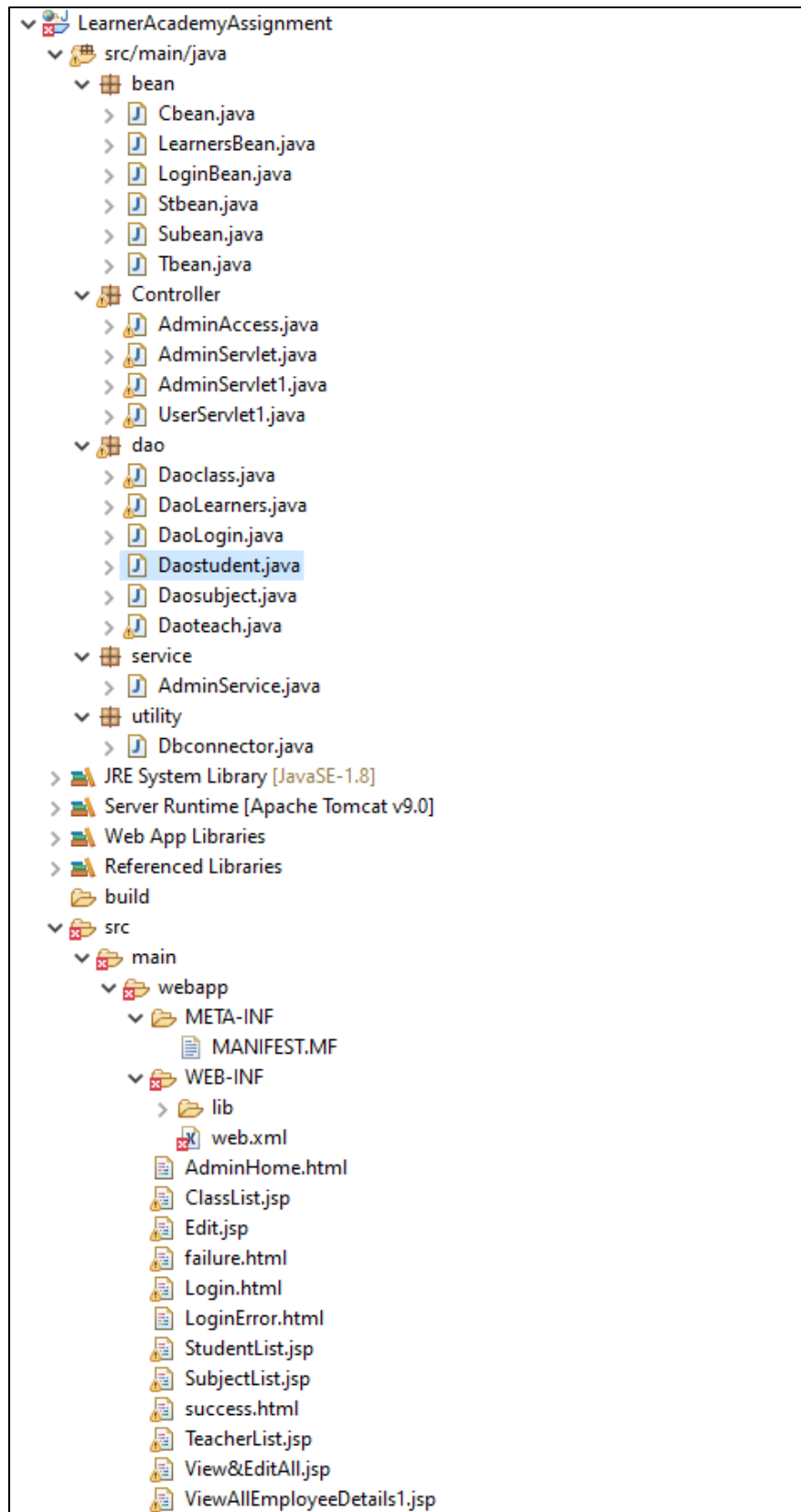
The application is:

- Easy to use
- Quick to implement using WAR files and uploading to Tomcat
- Straight-forward
- Can search large folders for existing files.





Model View Controller Architecture



[Cbean.java](#)

Source Code

```
package bean;

public class Cbean {

    private String className;

    public String getClassName() {
        return className;
    }

    public void setClassName(String className) {
        this.className = className;
    }

}
```

[LearnersBean.java](#)

Source Code

```
package bean;

public class LearnersBean {

    private String classroom;
    private String subject;
    private String TeacherName;
    private String TeacherSurname;
    private String StudentName ;
    private String StudentSurName ;
    public String getClassroom() {
        return classroom;
    }
    public void setClassroom(String classroom) {
        this.classroom = classroom;
    }
    public String getSubject() {
        return subject;
    }
    public void setSubject(String subject) {
        this.subject = subject;
    }
    public String getTeacherName() {
        return TeacherName;
    }
    public void setTeacherName(String teacherName) {
        TeacherName = teacherName;
    }
    public String getTeacherSurname() {
        return TeacherSurname;
    }
    public void setTeacherSurname(String teacherSurname) {
        TeacherSurname = teacherSurname;
    }

}
```

```
}  
public String getStudentName() {  
    return StudentName;  
}  
public void setStudentName(String studentName) {  
    StudentName = studentName;  
}  
public String getStudentSurName() {  
    return StudentSurName;  
}  
public void setStudentSurName(String studentSurName) {  
    StudentSurName = studentSurName;  
}  
}
```

LoginBean.java

Source Code

```
package bean;  
  
public class LoginBean {  
  
    private String username;  
    private int password;  
  
    public String getUsername() {  
        return username;  
    }  
    public void setUsername(String username) {  
        this.username = username;  
    }  
    public int getPassword() {  
        return password;  
    }  
    public void setPassword(int password) {  
        this.password = password;  
    }  
}
```

Stbean.java

Source Code

```
package bean;  
  
public class Stbean {  
  
    private String studentName;  
    private String studentSurname;  
  
    public String getStudentName() {  
        return studentName;  
    }  
    public void setStudentName(String studentName) {
```

```
        this.studentName = studentName;
    }
    public String getStudentSurname() {
        return studentSurname;
    }
    public void setStudentSurname(String studentSurname) {
        this.studentSurname = studentSurname;
    }
}

}
```

Subean.java

Source Code

```
package bean;

public class Subean {

    private String subjectName;
    private String subjectCode;

    public String getSubjectName() {
        return subjectName;
    }
    public void setSubjectName(String subjectName) {
        this.subjectName = subjectName;
    }
    public String getSubjectCode() {
        return subjectCode;
    }
    public void setSubjectCode(String subjectCode) {
        this.subjectCode = subjectCode;
    }
}

}
```

Tbean.java

Source Code

```
package bean;

public class Tbean {

    private String teacherName;
    private String teacherSurname;

    public String getTeacherName() {
        return teacherName;
    }
    public void setTeacherName(String teacherName) {
        this.teacherName = teacherName;
    }
    public String getTeacherSurname() {
```



```
        return teacherSurname;
    }
    public void setTeacherSurname(String teacherSurname) {
        this.teacherSurname = teacherSurname;
    }
}
```

AdminAccess.java (Servlet)

Source Code

```
package Controller;

import java.io.IOException;
import java.util.List;

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 bean.Cbean;
import bean.LearnersBean;
import bean.Tbean;
import service.AdminService;

/**
 * Servlet implementation class AdminAccess
 */
public class AdminAccess extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public AdminAccess() {

    }

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

        HttpSession session = request.getSession();
        AdminService service = new AdminService();
        String generateObj = (String) session.getAttribute("generateObj");

        if(generateObj.equalsIgnoreCase("true"))
        {

            List<Cbean> clistAdmin = service.viewClasses();
            List<LearnersBean> list = service.viewLearnerTable();
            List<Tbean> tlist = service.viewTProfile();
            request.setAttribute("listofLearnerObj", list);
            request.setAttribute("listofClassListObj", clistAdmin);
        }
    }
}
```

```
        RequestDispatcher rd =  
request.getRequestDispatcher("/ViewAllEmployeeDetails1.jsp");  
        rd.forward(request, response);  
  
    }  
  
    }  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {  
        // TODO Auto-generated method stub  
        doGet(request, response);  
    }  
  
}
```

AdminServlet1.java (Servlet)

Source Code

```
package Controller;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.StringTokenizer;
import java.util.TreeSet;

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 org.apache.catalina.connector.OutputBuffer;

import bean.Cbean;
import bean.LearnersBean;
import bean.Stbean;
import bean.Subbean;
import bean.Tbean;
import service.AdminService;

/**
 * Servlet implementation class AdminServlet1
 */
public class AdminServlet1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public AdminServlet1() {
```

```
}
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
    HttpSession session = request.getSession();  
    AdminService service = new AdminService();  
    String generateObj = (String) session.getAttribute("generateObj");  
  
  
    String buttonValue = request.getParameter("button");  
  
    if (buttonValue == null || buttonValue.isEmpty())  
    {  
        System.out.println("Button val is null");  
    }  
  
    else if(buttonValue.equalsIgnoreCase("Save"))  
    {  
  
        System.out.println(buttonValue);  
        String classes = request.getParameter("classlist");  
        String teachName = request.getParameter("editteachname");  
        String teachSurname = request.getParameter("editteachsurname");  
        String subject = request.getParameter("editsubject");  
        String studentName = request.getParameter("editstudentname");
```

```
String studentSurname = request.getParameter("editstudentsurname");

//System.out.println(classes+teachName+teachSurname+subject+studentName+studentSurname);

LearnersBean bean = new LearnersBean();

bean.setClassroom(classes);
bean.setTeacherName(teachName);
bean.setTeacherSurname(teachSurname);
bean.setStudentName(studentName);
bean.setStudentSurName(studentSurname);
bean.setSubject(subject);

System.out.println(bean.getClassroom() + bean.getStudentName()+bean.getSubject());

boolean list1 = service.insertinLearnersTable(bean);
System.out.println(list1);

if(list1 == true)
{
    RequestDispatcher rd2 = request.getRequestDispatcher("/success.html");
    rd2.forward(request, response);

    //response.setIntHeader("Refresh", 0);
    System.out.println(buttonValue);
}

else
{
    RequestDispatcher rd2 = request.getRequestDispatcher("/failure.html");
    rd2.forward(request, response);
    System.out.println(buttonValue);
}
```

```
    }

    else if ((buttonValue.contains("Delete")))
    {
        System.out.println(buttonValue);
        System.out.println("Delete Button Works");

        String studentName = buttonValue.split(",")[1];
        String studentSurname = buttonValue.split(",")[2];
        System.out.println(studentName+" "+studentSurname);

        int count = service.deleteLearnerByName(studentName,studentSurname);

        if (count>0)
        {
            RequestDispatcher rd = request.getRequestDispatcher("/success.html");
            rd.forward(request, response);
        }

        else
        {
            RequestDispatcher rd = request.getRequestDispatcher("/failure.html");
            rd.forward(request, response);
        }

    }

    else if ((buttonValue.contains("Edit")))
    {
        System.out.println(buttonValue);
        System.out.println("Edit Button Works");

        String studentName = buttonValue.split(",")[1];
```

Developing a Backend Admin for Learner's Academy.

```
String studentSurname = buttonValue.split(",")[2];
System.out.println(studentName+" "+studentSurname);
LearnersBean Search = service.SearchLearner(studentName, studentSurname);
request.setAttribute("Search", Search);
List<Cbean> clistAdmin = service.viewClasses();
request.setAttribute("listOfClassListObj", clistAdmin);

if (Search != null)
{
    int count = service.deleteLearnerByName(studentName, studentSurname);
}

RequestDispatcher rd = request.getRequestDispatcher("/Edit.jsp");
rd.forward(request, response);
}

else if ((buttonValue.contains("subjectlist")))
{
    System.out.println(buttonValue);
    Set<String> hs = new HashSet();
    List<LearnersBean> lbean = (List<LearnersBean>) service.viewLearnerTable();
    for(LearnersBean bean:lbean)
    {
        hs.add(bean.getSubject());
    }

    request.setAttribute("listofsubjects",hs );

    /*
    * for(LearnersBean b:hs) { System.out.println(b.getSubject()); }
    */

    RequestDispatcher rd = request.getRequestDispatcher("/SubjectList.jsp");
```

```
rd.forward(request, response);

    }

    else if ((buttonValue.contains("studentlist")))
    {

        System.out.println(buttonValue);

        Set<Stbean> hs = new HashSet<Stbean>();
        List<LearnersBean> lbean = (List<LearnersBean>) service.viewLearnerTable();
        for(LearnersBean bean:lbean)
        {

            Stbean st = new Stbean();
            st.setStudentName(bean.getStudentName());
            st.setStudentSurname(bean.getStudentSurName());
            hs.add(st);

        }

        for(Stbean bean:hs)
            System.out.println(bean.getStudentName() + bean.getStudentSurname());

        request.setAttribute("listofstudents",hs );

        RequestDispatcher rd = request.getRequestDispatcher("/StudentList.jsp");
        rd.forward(request, response);
```



```
    }

    else if ((buttonValue.contains("teacherlist")))
    {

        System.out.println(buttonValue);

        Set<Tbean> hs = new HashSet<Tbean>();
        List<LearnersBean> lbean = (List<LearnersBean>) service.viewLearnerTable();
        for(LearnersBean bean:lbean)
        {

            Tbean tb = new Tbean();
            tb.setTeacherName(bean.getTeacherName());
            tb.setTeacherSurname(bean.getTeacherSurname());
            hs.add(tb);

        }

        for(Tbean bean:hs)
            System.out.println(bean.getTeacherName() + bean.getTeacherSurname());

        request.setAttribute("listofteachers",hs );

        RequestDispatcher rd = request.getRequestDispatcher("/TeacherList.jsp");
        rd.forward(request, response);

    }

    else if ((buttonValue.contains("classlist")))
    {
```

```
System.out.println(buttonValue);

Set<String> hs = new HashSet<String>();
List<String> list = new <String>ArrayList();
TreeSet<String> myTreeSet = new TreeSet<String>();
List<LearnersBean> lbean = (List<LearnersBean>) service.viewLearnerTable();
for(LearnersBean bean:lbean)
{

    hs.add(bean.getClassroom());
    list.addAll(hs);
    myTreeSet.addAll(hs);

}

for(String classlist:hs)
    System.out.println(classlist.toString());

request.setAttribute("listofclass",myTreeSet );

RequestDispatcher rd = request.getRequestDispatcher("/ClassList.jsp");
rd.forward(request, response);

}

else if ((buttonValue.contains("logout")))
{

System.out.println(buttonValue);
```

```
        //request.getSession(false).invalidate();
        session.invalidate();
        session = null;

        if (session == null)
            System.out.println("Session invalidated");

        else
            System.out.println("Session NOT INVALIDATED");

        RequestDispatcher rd = request.getRequestDispatcher("/Login.html");
        System.out.println("Context path = "+request.getContextPath());
        response.sendRedirect(request.getContextPath() + "/Login.html");
        // rd.include(request, response);
        //response.sendRedirect("/Login.html");

    }

}

private void handleCaseFormessedUpForm() {
    // TODO Auto-generated method stub

}

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

```
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```

UserServlet1.java (Servlet)

Source Code

```
package Controller;

import java.io.IOException;
import java.util.List;

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 bean.Cbean;
import bean.LearnersBean;
import bean.LoginBean;
import service.AdminService;

public class UserServlet1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public UserServlet1() {
    }
}
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    //Logout  
    HttpSession session = request.getSession();  
    session.invalidate();  
}  
  
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    System.out.println("Post of UserServlet1");  
    String userName = request.getParameter("userName");  
    String generateObj = "false" ;  
    int password = Integer.parseInt(request.getParameter("password"));  
  
    LoginBean beanFromUser = new LoginBean();  
    beanFromUser.setUsername(userName);  
    beanFromUser.setPassword(password);  
    System.out.println(userName);  
    System.out.println(password);  
  
    AdminService service = new AdminService();  
    String user = service.login(beanFromUser);  
    HttpSession session = request.getSession();  
  
    if (user == null)  
    {  
        RequestDispatcher rd = request.getRequestDispatcher("/LoginError.html");  
        rd.forward(request, response);  
    }  
}
```

```
else if (user.equalsIgnoreCase("Admin"))
{
    session.setAttribute("userName", userName);
    generateObj = "true";
    session.setAttribute("generateObj", generateObj);

    String sessiontime = "Session ID: " +session.getId() + "Session created: "
    +session.getCreationTime();
    session.setAttribute("sessiontime",sessiontime);
    System.out.println(sessiontime);

    /*
    * RequestDispatcher rd = request.getRequestDispatcher("AdminServlet1.jsp");
    * rd.forward(request, response);
    */

    RequestDispatcher rd = request.getRequestDispatcher("AdminAccess");
    rd.forward(request, response);

}

}
```

[Daoclass.java](#)**Source Code**

```
package dao;

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

import bean.Cbean;
import bean.Stbean;
import utility.Dbconnector;

public class Daoclass {

    public List<Cbean> getAllClasses() {

        List<Cbean> listOfClasses = new ArrayList<Cbean>();

        try {
            Connection con = Dbconnector.getConnection();
            java.sql.Statement stmt = con.createStatement();
            String query = "Select * from classes";
            stmt.execute(query);
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next()) {
                Cbean bean = new Cbean();
                bean.setClassName(rs.getString(1));
                listOfClasses.add(bean);
            }

            con.close();
        }
    }
}
```

```
    }

    catch (Exception e) {

    }

    return listOfClasses;

}

public boolean insertClass(Cbean bean) {
    boolean insertStatus = false;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "insert into classes values (?)";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setString(1, bean.getClassName());
        int count = pstmt.executeUpdate();
        if (count > 0) {
            insertStatus = true;
        }

        System.out.println("Rows impacted: " + count);
        con.close();
    }

    catch (Exception e) {
        e.printStackTrace();
    }

    return insertStatus;
}

public int deleteClassByName(String name) {
    int deleteCount = 0;
```



```
        try {
            Connection con = Dbconnector.getConnection();
            String query = "delete classes where name =?";
            PreparedStatement pstmt = con.prepareStatement(query);
            pstmt.setString(1, name);
            deleteCount = pstmt.executeUpdate();
            System.out.println("Rows impacted: " + deleteCount);
            con.close();
        }

        catch (Exception e) {
            e.printStackTrace();
        }

        return deleteCount;
    }
}
```

[DaoLearners.java](#)**Source Code**

```
package dao;

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

import bean.Cbean;
import bean.LearnersBean;
import utility.Dbconnector;
```

```
public class DaoLearners {

    public List<LearnersBean> getAll() {

        List<LearnersBean> listOfTable = new ArrayList<LearnersBean>();

        // Always initialise the variable prior to using it !!!!!
        // Else you will run the risk of it having a value and executing a false
        // positive !!

        //listOfTable = null;

        try {
            Connection con = Dbconnector.getConnection();
            java.sql.Statement stmt = con.createStatement();
            String query = "Select * from Learners";
            stmt.execute(query);
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next()) {
                LearnersBean bean = new LearnersBean();
                bean.setClassroom(rs.getString(1));
                ;
                bean.setSubject(rs.getString(2));
                bean.setTeacherName(rs.getString(3));
                bean.setTeacherSurname(rs.getString(4));
                bean.setStudentName(rs.getString(5));
                bean.setStudentSurName(rs.getString(6));
                listOfTable.add(bean);
            }

            if(listOfTable.isEmpty())
            {
                listOfTable = null;
            }
        }
    }
}
```

```
        con.close();

    }

    catch (Exception e) {

    }

    return listOfTable;

}

public boolean insertAll(LearnersBean bean) {
    boolean insertStatus = false;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "insert into Learners values (?, ?, ?, ?, ?, ?)";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setString(1, bean.getClassroom());
        pstmt.setString(2, bean.getSubject());
        pstmt.setString(3, bean.getTeacherName());
        pstmt.setString(4, bean.getTeacherSurname());
        pstmt.setString(5, bean.getStudentName());
        pstmt.setString(6, bean.getStudentSurName());
        int count = pstmt.executeUpdate();
        if (count > 0) {
            insertStatus = true;
        }

        System.out.println("Rows impacted: " + count);
        con.close();
    }

    catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
        return insertStatus;
    }

    public int deleteClassByName(String name, String surname) {
        int deleteCount = 0;

        try {
            Connection con = Dbconnector.getConnection();
            String query = "delete Learners where StudentName =? And STUDENTSURNAME =?";
            PreparedStatement pstmt = con.prepareStatement(query);
            pstmt.setString(1, name);
            pstmt.setString(2, surname);
            deleteCount = pstmt.executeUpdate();
            System.out.println("Rows impacted: " + deleteCount);
            con.close();
        }

        catch (Exception e) {
            e.printStackTrace();
        }

        return deleteCount;
    }

    public LearnersBean searchLearnersByName(String name, String surname) {

        LearnersBean bean = new LearnersBean();

        try {
            Connection con = Dbconnector.getConnection();
            java.sql.Statement stmt = con.createStatement();
            String query = "SELECT * FROM Learners WHERE STUDENTNAME='"+name+"' AND STUDENTSURNAME='"+surname+"'";
            stmt.execute(query);
```

```
        ResultSet rs = stmt.executeQuery(query);
        if (rs.next())
        {
            bean.setClassroom(rs.getString(1));
            bean.setSubject(rs.getString(2));
            bean.setTeacherName(rs.getString(3));
            bean.setTeacherSurname(rs.getString(4));
            bean.setStudentName(rs.getString(5));
            bean.setStudentSurName(rs.getString(6));

        }

        else
        {
            bean = null;
        }
        System.out.println(bean.getClassroom() + bean.getStudentName() + bean.getStudentSurName());
    }

    catch (Exception e) {
        e.printStackTrace();
    }

    return bean;
}
}
```

[DaoLogin.java](#)**Source Code**

```
package dao;

import java.sql.Connection;
import java.sql.ResultSet;

import bean.LoginBean;
import utility.Dbconnector;

public class DaoLogin {

    public LoginBean getUserByUserName(String userName) {
        LoginBean bean = new LoginBean();

        try {
            Connection con = Dbconnector.getConnection();
            java.sql.Statement stmt = con.createStatement();
            String query = "Select * from Learners_login where username =" + "'" + userName + "'";
            stmt.execute(query);
            ResultSet rs = stmt.executeQuery(query);
            if(rs.next()) {

                bean.setUsername(rs.getString(1));
                bean.setPassword(rs.getInt(2));

            }
            else {
                bean = null;
            }

        }

        catch (Exception e) {

        }

    }

}
```

```
        return bean;
    }
}
```

Daostudent.java

Source Code

```
package dao;

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

import bean.Stbean;
import utility.Dbconnector;

public class Daostudent {

    /*
     * public static void main(String [] args) {
     *
     *     Daostudent std = new Daostudent(); List<Stbean> listOfStudents =
     *     std.getAllStudents(); for (Stbean bean:listOfStudents) {
     *     System.out.println(bean.getStudentName() + " & name: " +
     *     bean.getStudentSurname()); }
     * }
     */
}
```

```
public List<Stbean> getAllStudents()
{
    List<Stbean> listOfStudents = new ArrayList<Stbean>();

    try
    {
        Connection con = Dbconnector.getConnection();
        java.sql.Statement stmt = con.createStatement();
        String query = "Select * from students";
        stmt.execute(query);
        ResultSet rs = stmt.executeQuery(query);
        while(rs.next())
        {
            Stbean bean = new Stbean();
            bean.setStudentName(rs.getString(1));
            bean.setStudentSurname(rs.getString(2));
            listOfStudents.add(bean);

        }

        con.close();
    }

    catch(Exception e)
    {

    }

    return listOfStudents;
}
```



```
public boolean insertStudent(Stbean bean)
{
    boolean insertStatus = false;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "insert into students values (?,?)";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setString(1, bean.getStudentName());
        pstmt.setString(2, bean.getStudentSurname());
        int count = pstmt.executeUpdate();
        if(count>0)
        {
            insertStatus = true;
        }

        System.out.println("Rows impacted: " + count);
        con.close();
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }

    return insertStatus;
}

public int deleteStudentByName(int name)
{
    int deleteCount = 0;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "delete students where name =?";
```

Developing a Backend Admin for Learner's Academy.

```
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setInt(1, name);
        deleteCount = pstmt.executeUpdate();
        System.out.println("Rows impacted: " + deleteCount);
        con.close();
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }

    return deleteCount;
}

}
```

[Daosubject.java](#)**Source Code**

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

import bean.Subbean;
import utility.Dbconnector;

public class Daosubject {
```

```
public List<Subean> getAllSubjects()
{
    List<Subean> listOfSubjects = new ArrayList<Subean>();

    try
    {
        Connection con = Dbconnector.getConnection();
        java.sql.Statement stmt = con.createStatement();
        String query = "Select * from subjects";
        stmt.execute(query);
        ResultSet rs = stmt.executeQuery(query);
        while(rs.next())
        {
            Subean bean = new Subean();
            bean.setSubjectName(rs.getString(1));
            bean.setSubjectCode(rs.getString(2));
            listOfSubjects.add(bean);
        }

        con.close();
    }

    catch(Exception e)
    {
    }

    return listOfSubjects;
}

public boolean insertSubject(Subean bean)
```

```
{
    boolean insertStatus = false;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "insert into subjects values (?,?)";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setString(1, bean.getSubjectName());
        pstmt.setString(2, bean.getSubjectCode());
        int count = pstmt.executeUpdate();
        if(count>0)
        {
            insertStatus = true;
        }

        System.out.println("Rows impacted: " + count);
        con.close();
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }

    return insertStatus;
}

public int deleteSubjectByName(int name)
{
    int deleteCount = 0;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "delete subjects where name =?";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setInt(1, name);
```

```
        deleteCount = pstmt.executeUpdate();
        System.out.println("Rows impacted: " + deleteCount);
        con.close();
    }

    catch(Exception e)
    {
        e.printStackTrace();
    }

    return deleteCount;
}
}
```

Daoteach.java

Source Code

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

import bean.Stbean;
import bean.Tbean;
import utility.Dbconnector;

public class Daoteach {

    public List<Tbean> getAllTeachers()
    {

        List<Tbean> listOfTeachers= new ArrayList<Tbean>();
```

```
try
{
    Connection con = Dbconnector.getConnection();
    java.sql.Statement stmt = con.createStatement();
    String query = "Select * from subjects";
    stmt.execute(query);
    ResultSet rs = stmt.executeQuery(query);
    while(rs.next())
    {
        Tbean bean = new Tbean();
        bean.setTeacherName(rs.getString(1));
        bean.setTeacherSurname(rs.getString(2));
        listOfTeachers.add(bean);
    }

    con.close();
}

catch(Exception e)
{
}

return listOfTeachers;
}

public boolean insertTeach(Tbean bean)
{
    boolean insertStatus = false;

    try {
```

Developing a Backend Admin for Learner's Academy.

```
Connection con = Dbconnector.getConnection();
String query = "insert into teachers values (?,?)";
PreparedStatement pstmt = con.prepareStatement(query);
pstmt.setString(1, bean.getTeacherName());
pstmt.setString(2, bean.getTeacherSurname());
int count = pstmt.executeUpdate();
if(count>0)
{
    insertStatus = true;
}

System.out.println("Rows impacted: " + count);
con.close();
}

catch(Exception e)
{
    e.printStackTrace();
}

return insertStatus;
}

public int deleteTeachersByName(int name)
{
    int deleteCount = 0;

    try {
        Connection con = Dbconnector.getConnection();
        String query = "delete teachers where name =?";
        PreparedStatement pstmt = con.prepareStatement(query);
        pstmt.setInt(1, name);
        deleteCount = pstmt.executeUpdate();
        System.out.println("Rows impacted: " + deleteCount);
        con.close();
    }
```

```
        catch(Exception e)
        {
            e.printStackTrace();
        }
        return deleteCount;
    }
}
```

[AdminService.java](#)**Source Code**

```
package service;

import java.util.List;

import bean.Cbean;
import bean.LearnersBean;
import bean.LoginBean;
import bean.Stbean;
import bean.Subean;
import bean.Tbean;
import dao.DaoLearners;
import dao.DaoLogin;
import dao.Daoclass;
import dao.Daostudent;
import dao.Daosubject;
import dao.Daoteach;
```



```
public class AdminService {

    DaoLogin login = new DaoLogin();
    Daostudent student = new Daostudent();
    Daosubject subject = new Daosubject();
    Daoteach teach = new Daoteach();
    Daoclass classes = new Daoclass();
    DaoLearners learner= new DaoLearners();


    public boolean AddStProfile(Stbean bean)
    {
        return student.insertStudent(bean);
    }

    public int deleteStProfile(int stId)
    {
        return student.deleteStudentByName(stId);
    }

    public List<Stbean> viewStProfile()
    {
        return student.getAllStudents();
    }

    public boolean AddTPProfile(Tbean bean)
    {
        return teach.insertTeach(bean);
    }
}
```

```
public int deleteTProfile(int TId)
{
    return teach.deleteTeachersByName(TId);
}

public List<Tbean> viewTProfile()
{
    return teach.getAllTeachers();
}

public boolean AddSuProfile(Subean bean)
{
    return subject.insertSubject(bean);
}

public int deleteSuProfile(int SuId)
{
    return subject.deleteSubjectByName(SuId);
}

public List<Subean> viewSuProfile()
{
    return subject.getAllSubjects();
}

public boolean AddClasses(Cbean bean)
{
    return classes.insertClass(bean);
}

public int deleteClasses(String classname)
{

```

```
        return classes.deleteClassByName(classname);
    }

    public List<Cbean> viewClasses()
    {
        return classes.getAllClasses();
    }

    public boolean insertinLearnersTable(LearnersBean bean)
    {
        return learner.insertAll(bean);
    }

    public int deleteLearnerByName(String learnername, String learnerSurname)
    {
        return learner.deleteClassByName(learnername, learnerSurname);
    }

    public List<LearnersBean> viewLearnerTable()
    {
        return learner.getAll();
    }

    public LearnersBean SearchLearner(String name, String surname)
    {
        return learner.searchLearnersByName(name, surname);
    }
}
```

```
DaoLogin userDao = new DaoLogin();
```

Developing a Backend Admin for Learner's Academy.

```
public String login(LoginBean beanFromUser)
{
    String user = null;

    LoginBean beanFromDb = userDao.getUserByUserName(beanFromUser.getUsername()) ;

    if(beanFromDb != null)
    {

        if(beanFromUser.getUsername().equals(beanFromDb.getUsername()) && beanFromUser.getPassword() ==
beanFromDb.getPassword())
        {
            user = beanFromDb.getUsername();
            System.out.println("Password Is CORRECT");
            System.out.println("Admin Service - Bean from DB - username = '" +beanFromDb.getUsername()+"' &
password = '" + beanFromDb.getPassword()+"'");
            System.out.println("Admin Service - Bean from USER - username = '" +beanFromUser.getUsername()+"'
& password = '" + beanFromUser.getPassword()+"'");
        }
        else
        {
            System.out.println("ADMIN SERVICE - INCORRECT USER OR PASWORD");
            System.out.println("Admin Service - Bean from DB - username = '" +beanFromDb.getUsername()+"' &
password = '" + beanFromDb.getPassword()+"'");
            System.out.println("Admin Service - Bean from USER - username = '" +beanFromUser.getUsername()+"'
& password = '" + beanFromUser.getPassword()+"'");
        }
    }

    return user;
}
```

[Dbconnector.java](#)**Source Code**

```
package utility;

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

public class Dbconnector {

    static Connection con = null ;

    public static Connection getConnection() throws SQLException
    {

        //Added con.isClosed to avoid a closed connection con == null is not enough at times
        if (con == null || con.isClosed())
        {

            try {

                Class.forName("oracle.jdbc.driver.OracleDriver");
                con = (Connection) DriverManager.getConnection(
                    "jdbc:oracle:thin:@localhost:1521:xe", "system", "karl");
                System.out.println("Connection Success");
            }

            catch (Exception e)
            {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }

        return con;
    }
}
```

```
}  
}
```

[web.xml](#)**Source Code**

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee"  
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"  
version="2.5">  
  <display-name>LearnerAcademyAssignment</display-name>  
  
  <welcome-file-list>  
    <welcome-file>Login.html</welcome-file>  
    <!-- <welcome-file>index.html</welcome-file>  
    <welcome-file>index.jsp</welcome-file>  
    <welcome-file>index.htm</welcome-file>  
    <welcome-file>default.html</welcome-file>  
    <welcome-file>default.jsp</welcome-file>  
    <welcome-file>default.htm</welcome-file> -->  
  </welcome-file-list>  
  
  <servlet>  
    <description></description>  
    <display-name>AdminServlet1</display-name>  
    <servlet-name>AdminServlet1</servlet-name>  
    <servlet-class>Controller.AdminServlet1</servlet-class>  
  </servlet>  
  
  <servlet-mapping>  
    <servlet-name>AdminServlet1</servlet-name>  
    <url-pattern>/AdminServlet1</url-pattern>  
  </servlet-mapping>  
  
  <servlet>  
    <description></description>  
    <display-name>UserServlet1</display-name>  
    <servlet-name>UserServlet1</servlet-name>
```

```
<servlet-class>Controller.UserServlet1</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>UserServlet1</servlet-name>
  <url-pattern>/UserServlet1</url-pattern>
</servlet-mapping>
<servlet>
  <description></description>
  <display-name>AdminAccess</display-name>
  <servlet-name>AdminAccess</servlet-name>
  <servlet-class>Controller.AdminAccess</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AdminAccess</servlet-name>
  <url-pattern>/AdminAccess</url-pattern>
</servlet-mapping>
</web-app>
```

[AdminHome.html](#)**Source Code**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Admin home</title>
</head>
<body>
<h3 align="center"> Admin Home Page</h3>
<div><a href=""> Logout</a></div>
<br>
<form action="AdminServlet1" method="get">

<a href="ViewALLEmployeeDetails.jsp" > <input > View TABLE </a><br>

</form>
</body>
```

</html>

ClassList.jsp

Source Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="java.util.ArrayList"
    import=" java.util.List"
    import = "service.AdminService"
    import = "bean.LearnersBean"
    import = "bean.Cbean"
    import = "java.util.HashSet"
    import = "java.util.List"
    import = "java.util.Set"
    import = "bean.Tbean"
    import = "java.util.TreeSet"

    %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="AdminServlet1" method="get">
<h3 align="center"> All Classes</h3>
<hr>

<a href="AdminAccess" style="color:orange;"><h3>Press Link to go back to Learners List</h3> </a><br>

<table align="center" border="1">
    <tr>
        <td><H2> Class List</H2></td>
```



```
</tr>

<%
TreeSet <String>tslist = (TreeSet <String>)request.getAttribute("listofclass");
%>

<%
if(tslist!=null){

    for(String treevar:tslist)
    {
%>
<tr>
    <td><%=treevar%></td>
</tr>

        <%

    }
}

%>
</tr>
</form>
</body>
</html>
```

[Edit.jsp](#)**Source Code**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"

    import="java.util.ArrayList"
    import=" java.util.List"
    import = "service.AdminService"
import = "bean.LearnersBean"
import = "bean.Cbean"

%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="AdminServlet1" method="get">
<h3 align="center"> EDIT Employee Details</h3>
<hr>

    <table align="center" border="1">
        <tr>
            <td> Class</td>
            <td> Subject</td>
            <td> Teacher Name</td>
            <td> Teacher Surname</td>
            <td> Student Name</td>
            <td> Student Surname</td>
            <td colspan="1">Action</td>
        <tr/>

        <tr>
```

```
<td><select name="classList">
<%

    LearnersBean Lbean = (LearnersBean)request.getAttribute("Search");
    List<Cbean> Clist = (List<Cbean>)request.getAttribute("listofClassListObj");

    for(Cbean bean:Clist)
    {
        %>
        <option value="<%= bean.getClassName() %>"><%= bean.getClassName() %><option/>
    } %>
</select> </td>
<td><input type="text" value="<%=Lbean.getSubject()%>" name="editsubject"></td>
<td><input type="text" value="<%=Lbean.getTeacherName()%>" name="editteachname"></td>
<td><input type="text" value="<%=Lbean.getTeacherSurname() %>" name="editteachsurname"></td>
<td><input type="text" value="<%=Lbean.getStudentName()%>" name="editstudentname"></td>
<td><input type="text" value="<%=Lbean.getStudentSurName() %>" name="editstudentsurname"></td>
<td><input type="submit" value = "Save" name="button"></td>
</tr>

</table>
</form>
</body>
</html>
```

[failure.html](#)**Source Code**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
</head>
<body>
<h1 align="center"> FAILURE </h1><br>
<a href="AdminAccess" style="color: red;"><h3>Back to Home Page</h3> </a><br>
</body>
</html>
```

[Login.html](#)**Source Code**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action = "UserServlet1" method = "post">
<h2 align="center"> Login </h2>
<hr>
<table align="center">
  <tr>
    <td>UserName:</td>
    <td><input type="text" name="userName"></td>
  </tr>
  <tr>
    <td>Password:</td>
    <td><input type="password" name="password"></td>
  </tr>
  <tr>
    <td align="center"><input type="reset" value = "Reset"></td>
    <td align="center"><input type="submit" value="Login"></td>
  </tr>
</table>
```

```
</table>
</form>

</body>
</html>
```

LoginError.html

Source Code

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<div align="center" style="color: red;">
Invalid credentials <a href = "Login.html"> Login</a> again.
</div>

</body>
</html>
```

StudentList.jsp

Source Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    import="java.util.ArrayList"
    import=" java.util.List"
    import = "service.AdminService"
    import = "bean.LearnersBean"
    import = "bean.Stbean"
```

```
import = "java.util.HashSet"
import = "java.util.List"
import = "java.util.Set"
import = "java.util.HashSet"
import = "java.util.List"
import = "java.util.Set"

%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<form action="AdminServlet1" method="get">
<h3 align="center"> Student List</h3>
<hr>

<a href="AdminAccess" style="color:orange;"><h3>Press Link to go back to Learners List</h3> </a><br>

<table align="center" border="1">
  <!-- <tr>
    <td><H2> Student List</H2></td>
  <tr/> -->
  <tr>
    <td><H3> Student Name</H3></td>
    <td> <H3>Student Surname</H3></td>
  <tr/>

<%
Set<Stbean> hslist = (Set<Stbean>)request.getAttribute("listofstudents");
%>
```

```
<%  
if(hslist!=null){  
  
    for(Stbean hashvar:hslist)  
    {  
    %>  
  
    <tr>  
        <td><%=hashvar.getStudentName()%></td>  
        <td><%=hashvar.getStudentSurname()%></td>  
        <!-- <td><%=hashvar.hashCode()%></td> --%>  
    <tr/>  
  
        <%  
  
    }  
}  
%>  
<tr/>  
</form/>  
</body>  
</html>
```

SubjectList.jsp

Source Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"

    import="java.util.ArrayList"
        import=" java.util.List"
        import = "service.AdminService"
    import = "bean.LearnersBean"
        import = "bean.Cbean"
        import = "java.util.HashSet"
import = "java.util.List"
import = "java.util.Set"

    %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<form action="AdminServlet1" method="get">
<h3 align="center"> All Subjects</h3>
<hr>

<a href="AdminAccess" style="color:orange;"><h3>Press Link to go back to Learners List</h3> </a><br>

    <table align="center" border="1">
        <tr>
            <td><H2> Subject List</H2></td>
        <tr/>

    <%
Set<String> hslist = (Set<String>)request.getAttribute("listofsubjects");
```



```
%>

<%
if(hslist!=null){

    for(String hashvar:hslist)
    {
%>
<tr>
    <td><%=hashvar%></td>
<tr/>

        <%

    }
}

%>
<tr/>

<form/>

</body>
</html>
```

[success.html](#)**Source Code**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
```

```
</head>
<body>
<h1 align="center"> SUCCESS </h1><br>
<a href="AdminAccess" style="color: green;"><h3>Back to Home Page</h3> </a><br>
</body>
</html>
```

TeacherList.jsp

Source Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

import="java.util.ArrayList"
import=" java.util.List"
import = "service.AdminService"
import = "bean.LearnersBean"
import = "bean.Cbean"
import = "java.util.HashSet"
import = "java.util.List"
import = "java.util.Set"
import = "bean.Tbean"

%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<form action="AdminServlet1" method="get">
<h3 align="center"> Teachers List</h3>
<hr>
```

```
<a href="AdminAccess" style="color:orange;"><h3>Press Link to go back to Learners List</h3> </a><br>

<table align="center" border="1">
  <!-- <tr>
    <td><H2> Student List</H2></td>
  <tr/> -->
  <tr>
    <td><H3> Teacher Name</H3></td>
    <td> <H3>Teacher Surname</H3></td>
  <tr/>

<%
Set<Tbean> hslist = (Set<Tbean>)request.getAttribute("listofteachers");
%>

<%
if(hslist!=null){

  for(Tbean hashvar:hslist)
  {
%>

<tr>
  <td><%=hashvar.getTeacherName()%></td>
  <td><%=hashvar.getTeacherSurname()%></td>
  <!-- <td><%=hashvar.hashCode()%></td> --%>
<tr/>

      <%

  }

}
```

```
%>
<tr/>

<form/>

</body>
</html>
```

View&EditAll.jsp

Source Code

```
<%@page import="bean.LoginBean"%>
<%@page import="bean.LearnersBean"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    import="java.util.ArrayList"
    import=" java.util.List"
    import = "service.AdminService"
    import = "dao.DaoLearners"
    import = "bean.*"

    %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="AdminServlet1" method="get">
<h3 align="center"> All Employee Details</h3>
```

```
<hr>

<%
DaoLearners learners = new DaoLearners();
List<LearnersBean> list = learners.getAll();

%>

<table align="center" border="1">
  <tr>
    <td> Class</td>
    <td> Subject</td>
    <td> Teachers Name</td>
    <td> Teachers Surname</td>
    <td> Students Name</td>
    <td> Students Surname</td>
    <td colspan="2">Action</td>
    <td><input type="button" value="Edit" name="buttedit"></td>
    <td><input type="button" value="Delete" name="buttdelete"></a></td>
  <tr/>

</body>
</html>
```

ViewAllEmployeeDetails1.jsp

Source Code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
```

```
import="java.util.ArrayList"
import=" java.util.List"
import = "service.AdminService"
import = "bean.LearnersBean"
import = "bean.Cbean"

%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="AdminServlet1" method="get">
<h3 align="center"> All Employee Details</h3>
<hr>

<%

List<LearnersBean> list = (List<LearnersBean>)request.getAttribute("listofLearnerObj");

%>

<table align="center" border="">
  <tr>
    <td><button name="button" value="subjectlist" type= "submit" >Subjects' List</button></td>
    <td><button name="button" value="studentlist" type= "submit" >Students' List</button></td>
    <td><button name="button" value="teacherlist" type= "submit" >Teachers' List</button></td>
    <td><button name="button" value="classlist" type= "submit" >Class List</button></td>
    <td><button name="button" value="Logout" type= "submit" >Sign Out</button></td>
```

```

</tr>

</table>

<br>

<table align="center" border="1">
  <tr>
    <td> Class</td>
    <td> Subject</td>
    <td> Teacher Name</td>
    <td> Teacher Surname</td>
    <td> Student Name</td>
    <td> Student Surname</td>
    <td colspan="2">Action</td>
  </tr>

  <tr>
    <td><select name="classlist">
      <%
        if ( list == null || list.isEmpty())
        {

          List<Cbean> Clist = (List<Cbean>)request.getAttribute("listofClassListObj");
          for(Cbean bean:Clist)
          {
            %>
            <option value="<%= bean.getClassName() %>"><%= bean.getClassName() %></option>
            <% } %>
          </select> </td>
          <td><input type="text" value="Edit" name="editsubject"></td>

```

Developing a Backend Admin for Learner's Academy.

```

<td><input type="text" value="Edit" name="editteachname"></td>
<td><input type="text" value="Edit" name="editteachsurname"></td>
<td><input type="text" value="Edit" name="editstudentname"></td>
<td><input type="text" value="Edit" name="editstudentsurname"></td>
<td><input type="submit" value = "save" name="button"></td>

```

```

<!-- <td><%= bean.getTeacherName() %></td>
<td><%= bean.getTeacherSurname() %></td>
<td><%= bean.getStudentName() %></td>
<td><%= bean.getStudentSurName() %></td>

```

```

<td><input type="button" value="Edit" name="buttedit"></td>
<td><input type="button" value="Delete" name="buttdelete"></td>
</tr>      --%>

```

```

<%

```

```

}

```

```

else if (list!=null){

```

```

List<Cbean> Clist = (List<Cbean>)request.getAttribute("listofClassListObj");

```

```

for(LearnersBean bean:list)
{

```

```

%>

```

```

<tr>

```

```

    <td><%=bean.getClassroom() %></td>

```



```

<td><%=bean.getSubject() %></td>

<td><%= bean.getTeacherName() %></td>
<td><%= bean.getTeacherSurname() %></td>
<td><%= bean.getStudentName() %></td>
<td><%= bean.getStudentSurName() %></td>

<td><button name="button" value="Edit,<%=bean.getStudentName()%>,<%=bean.getStudentSurName()%>" type= "submit"
>Edit</button></td>
<td><button name="button" value="Delete,<%=bean.getStudentName()%>,<%=bean.getStudentSurName()%>"
type="submit">Delete</button></td>

<!-- <td><input type="button" value="Delete" name="button" ></td> -->

<%-- <%
String button2value = request.getParameter("buttdelete");
if (button2value.equalsIgnoreCase("Delete"))
{
    service.deleteProfile(bean.getEmpId());

}

%> --%>

<%

}

%>
<tr>
<td><select name="classlist">
<%
for(Cbean bean1:Clist)

```

```
{

    %>
    <option value="<%= bean1.getClassName() %>"><%= bean1.getClassName() %></option>
    <% } %>
</select> </td>
<td><input type="text" value="Edit" name="editsubject"></td>
<td><input type="text" value="Edit" name="editteachname"></td>
<td><input type="text" value="Edit" name="editteachsurname"></td>
<td><input type="text" value="Edit" name="editstudentname"></td>
<td><input type="text" value="Edit" name="editstudentsurname"></td>
<td><input type="submit" value = "save" name="button"></td>
</tr>

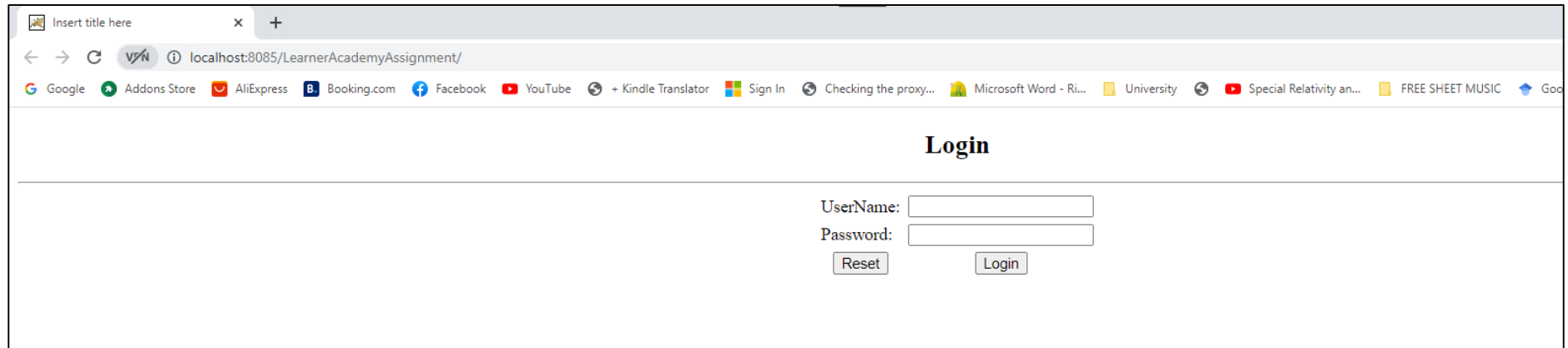
<%

}
%>

</table>

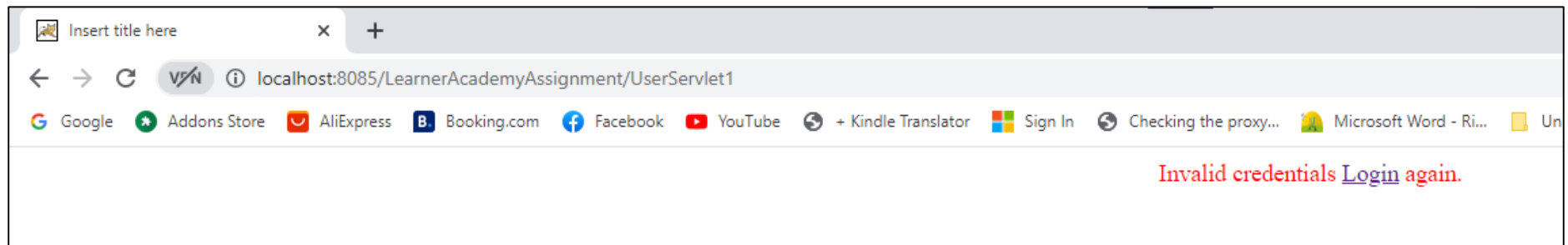
<br>
<br>

<h1> Session Details </h1>
<h2>< <%= session.getAttribute("sessiontime") %> ></h2>
</form>
</body>
</html>
```



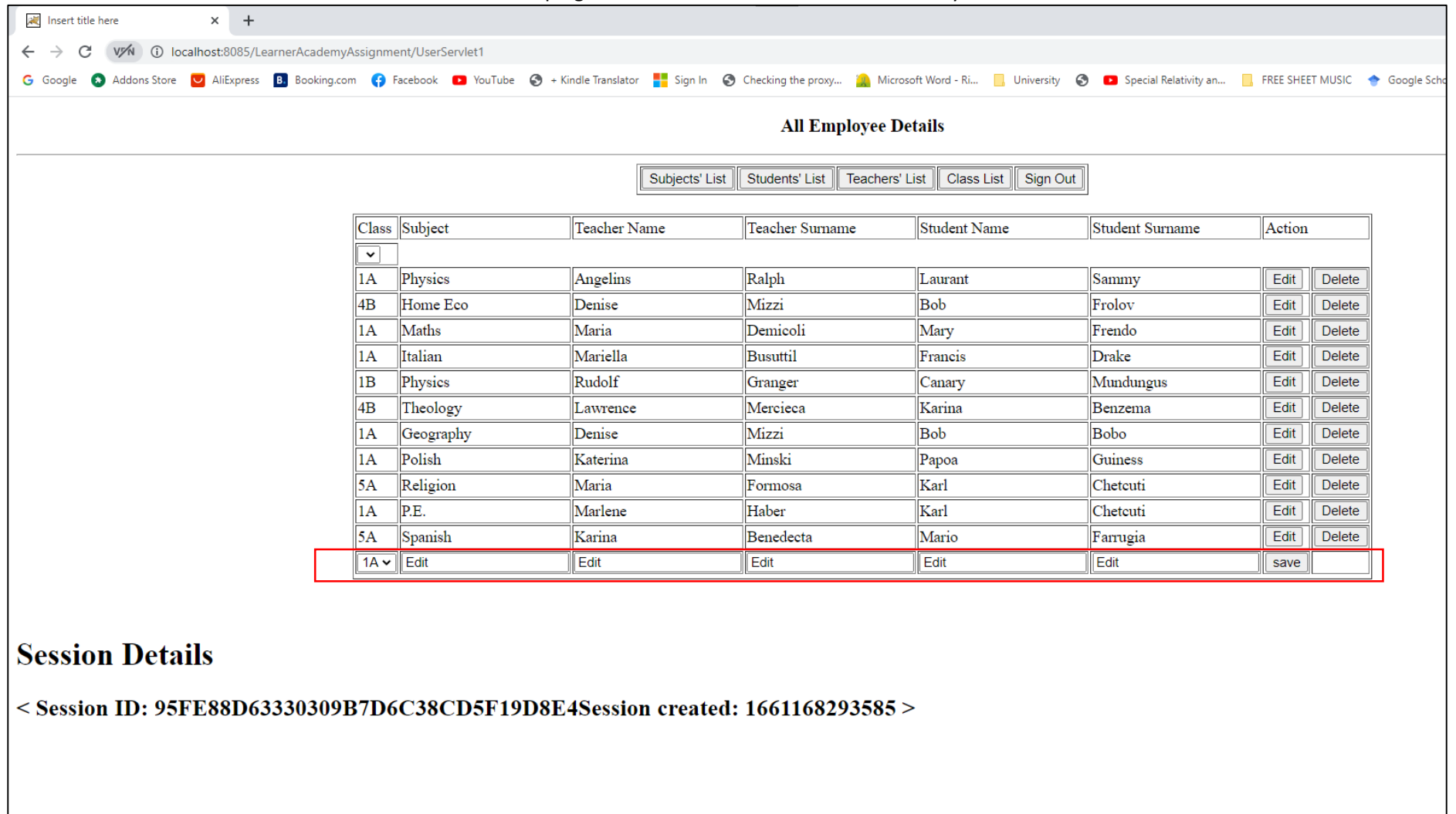
The screenshot shows a web browser window with the address bar displaying 'localhost:8085/LearnerAcademyAssignment/'. The page has a title 'Insert title here'. Below the title, the word 'Login' is centered in a bold font. Underneath, there are two input fields: 'UserName:' and 'Password:'. Below the 'Password:' field, there are two buttons: 'Reset' and 'Login'.

Figure 1 Login page for Admin



The screenshot shows the same web browser window, but the address bar now displays 'localhost:8085/LearnerAcademyAssignment/UserServlet1'. The page content is a red error message: 'Invalid credentials [Login](#) again.' The word 'Login' is a blue hyperlink.

Figure 2 Error message for failure to login with a `<href>` link to return to Login page.



The screenshot shows a web browser window with the URL `localhost:8085/LearnerAcademyAssignment/UserServlet1`. The page title is "All Employee Details". Below the title, there are navigation buttons: "Subjects' List", "Students' List", "Teachers' List", "Class List", and "Sign Out". The main content is a table with the following columns: Class, Subject, Teacher Name, Teacher Surname, Student Name, Student Surname, and Action. The table contains 13 rows of data. The last row is highlighted with a red border, indicating it is the current record being edited. The "Action" column for this row contains "save" and "Delete" buttons.

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action
▼						
1A	Physics	Angelins	Ralph	Laurant	Sammy	Edit Delete
4B	Home Eco	Denise	Mizzi	Bob	Frolov	Edit Delete
1A	Maths	Maria	Demicoli	Mary	Frengo	Edit Delete
1A	Italian	Mariella	Busuttill	Francis	Drake	Edit Delete
1B	Physics	Rudolf	Granger	Canary	Mundungus	Edit Delete
4B	Theology	Lawrence	Mercieca	Karina	Benzema	Edit Delete
1A	Geography	Denise	Mizzi	Bob	Bobo	Edit Delete
1A	Polish	Katerina	Minski	Papoa	Guiness	Edit Delete
5A	Religion	Maria	Formosa	Karl	Chetcuti	Edit Delete
1A	P.E.	Marlene	Haber	Karl	Chetcuti	Edit Delete
5A	Spanish	Karina	Benedecta	Mario	Farrugia	Edit Delete
1A ▼	Edit	Edit	Edit	Edit	Edit	save

Session Details

< Session ID: 95FE88D63330309B7D6C38CD5F19D8E4Session created: 1661168293585 >

Figure 3 Shows successful Login to application and red box to enter new details above.

Insert title here

localhost:8085/LearnerAcademyAssignment/UserServlet1

Google Addons Store AliExpress Booking.com Facebook YouTube Kindle Translator Sign In Checking the proxy... Microsoft Word - Ri... University Special Relativity an... FREE SHEET MUSIC Google Scholar

All Employee Details

Subjects' List

Students' List

Teachers' List

Class List

Sign Out

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action	
▼							
1A	Physics	Angelins	Ralph	Laurant	Sammy	Edit	Delete
4B	Home Eco	Denise	Mizzi	Bob	Frolov	Edit	Delete
1A	Maths	Maria	Demicoli	Mary	Frendo	Edit	Delete
1A	Italian	Mariella	Busuttil	Francis	Drake	Edit	Delete
1B	Physics	Rudolf	Granger	Canary	Mundungus	Edit	Delete
4B	Theology	Lawrence	Mercieca	Karina	Benzema	Edit	Delete
1A	Geography	Denise	Mizzi	Bob	Bobo	Edit	Delete
1A	Polish	Katerina	Minski	Papoa	Guiness	Edit	Delete
5A	Religion	Maria	Formosa	Karl	Chetcuti	Edit	Delete
1A	P.E.	Marlene	Haber	Karl	Chetcuti	Edit	Delete
5A	Spanish	Karina	Benedecta	Mario	Farrugia	Edit	Delete
1A ▼	Java FSD	Simpli	Lean	Karl	Chetcuti	save	

Session Details

< Session ID: 95FE88D63330309B7D6C38CD5F19D8E4Session created: 1661168293585 >

Figure 4 shows new details entered into the application. This will be saved.

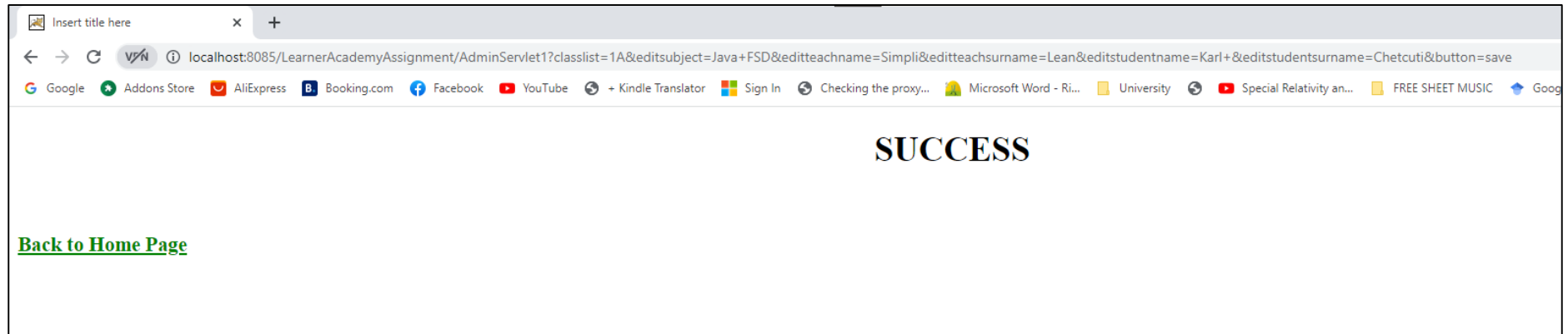


Figure 5 showed that details have been saved successfully.

Insert title here

localhost:8085/LearnerAcademyAssignment/AdminAccess

Google Addons Store AliExpress Booking.com Facebook YouTube Kindle Translator Sign In Checking the proxy... Microsoft Word - Ri... University Special Relativity an... FREE SHEET MUSIC Google Scholar Other bookmarks

All Employee Details

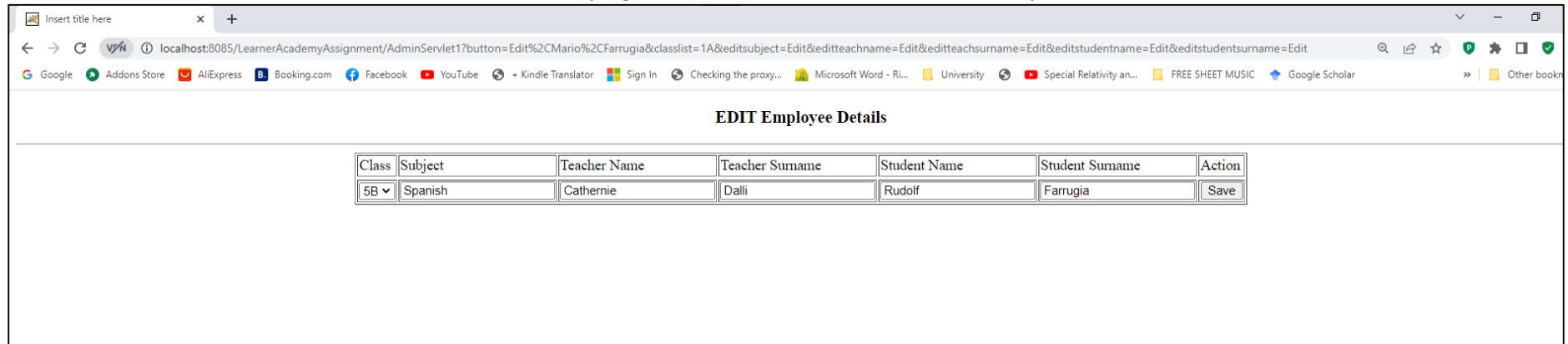
Subjects' List Students' List Teachers' List Class List Sign Out

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action
▼						
1A	Physics	Angelins	Ralph	Laurant	Sammy	Edit Delete
4B	Home Eco	Denise	Mizzi	Bob	Frolov	Edit Delete
1A	Maths	Maria	Demicoli	Mary	Frendo	Edit Delete
1A	Italian	Mariella	Busuttill	Francis	Drake	Edit Delete
1B	Physics	Rudolf	Granger	Canary	Mundungus	Edit Delete
4B	Theology	Lawrence	Mercieca	Karina	Benzema	Edit Delete
1A	Geography	Denise	Mizzi	Bob	Bobo	Edit Delete
1A	Polish	Katerina	Minski	Papoa	Guiness	Edit Delete
5A	Religion	Maria	Formosa	Karl	Chetcuti	Edit Delete
1A	P.E.	Marlene	Haber	Karl	Chetcuti	Edit Delete
1A	Java FSD	Simpli	Lean	Karl	Chetcuti	Edit Delete
5A	Spanish	Karina	Benedecta	Mario	Farrugia	Edit Delete
1A ▼	Edit	Edit	Edit	Edit	Edit	save

Session Details

< Session ID: 95FE88D63330309B7D6C38CD5F19D8E4Session created: 1661168293585 >

Figure 6 shows details saved in database in red, object in green will be edited.



EDIT Employee Details

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action
5B	Spanish	Cathernie	Dalli	Rudolf	Farrugia	Save

Figure 7 Shows editing of details which will be saved shortly.

Insert title here x +

localhost:8085/LearnerAcademyAssignment/AdminAccess

Google Addons Store AliExpress Booking.com Facebook YouTube + Kindle Translator Sign In Checking the proxy... Microsoft Word - Ri... University Special Relativity an... FREE SHEET MUSIC Google Scholar

All Employee Details

Subjects' List Students' List Teachers' List Class List Sign Out

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action
1A	Physics	Angelins	Ralph	Laurant	Sammy	Edit Delete
4B	Home Eco	Denise	Mizzi	Bob	Frolov	Edit Delete
1A	Maths	Maria	Demicoli	Mary	Frendo	Edit Delete
1A	Italian	Mariella	Busuttil	Francis	Drake	Edit Delete
1B	Physics	Rudolf	Granger	Canary	Mundungus	Edit Delete
4B	Theology	Lawrence	Mercieca	Karina	Benzema	Edit Delete
1A	Geography	Denise	Mizzi	Bob	Bobo	Edit Delete
1A	Polish	Katerina	Minski	Papoa	Guiness	Edit Delete
5A	Religion	Maria	Formosa	Karl	Chetcuti	Edit Delete
1A	P.E.	Marlene	Haber	Karl	Chetcuti	Edit Delete
1A	Java FSD	Simpli	Lean	Karl	Chetcuti	Edit Delete
5B	Spanish	Cathemie	Dalli	Rudolf	Farrugia	Edit Delete
1A	Edit	Edit	Edit	Edit	Edit	save

Session Details

< Session ID: 95FE88D63330309B7D6C38CD5F19D8E4Session created: 1661168293585 >

Figure 8 Shows that edited details in green have been saved successfully to database whereas detail in purple will be deleted.

Insert title here

localhost:8085/LearnerAcademyAssignment/AdminAccess

Google Addons Store AliExpress Booking.com Facebook YouTube + Kindle Translator Sign In Checking the proxy... Microsoft Word - Ri... University Special Relativity an... FREE SHEET MUSIC Google Scholar

All Employee Details

Subjects' List

Students' List

Teachers' List

Class List

Sign Out

Class	Subject	Teacher Name	Teacher Surname	Student Name	Student Surname	Action
1A	Physics	Angelins	Ralph	Laurant	Sammy	<div>EditDelete</div>
4B	Home Eco	Denise	Mizzi	Bob	Frolov	<div>EditDelete</div>
1A	Maths	Maria	Demicoli	Mary	Frendo	<div>EditDelete</div>
1A	Italian	Mariella	Busuttil	Francis	Drake	<div>EditDelete</div>
1B	Physics	Rudolf	Granger	Canary	Mundungus	<div>EditDelete</div>
4B	Theology	Lawrence	Mercieca	Karina	Benzema	<div>EditDelete</div>
1A	Geography	Denise	Mizzi	Bob	Bobo	<div>EditDelete</div>
1A	Polish	Katerina	Minski	Papoa	Guinness	<div>EditDelete</div>
1A	Java FSD	Simpli	Lean	Karl	Chetcuti	<div>EditDelete</div>
5B	Spanish	Cathernie	Dalli	Rudolf	Farrugia	<div>EditDelete</div>
1A	<div>Edit</div>	<div>Edit</div>	<div>Edit</div>	<div>Edit</div>	<div>Edit</div>	<div>save</div>

Session Details

< Session ID: 95FE88D63330309B7D6C38CD5F19D8E4Session created: 1661168293585 >

Figure 9 shows that subject religion has been deleted successfully. The buttons in a **blue** box above will now be pressed.

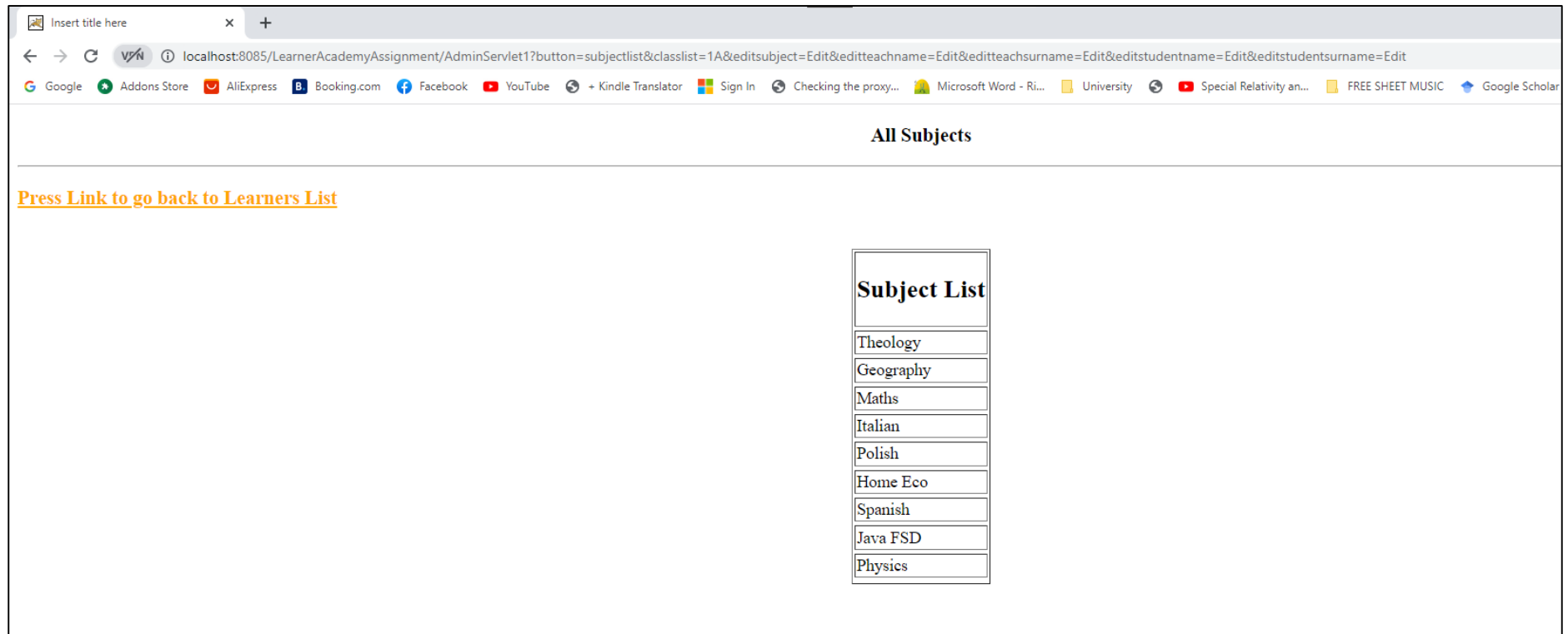


Figure 10 shows subject List where new entry **Java FSD** can be seen.

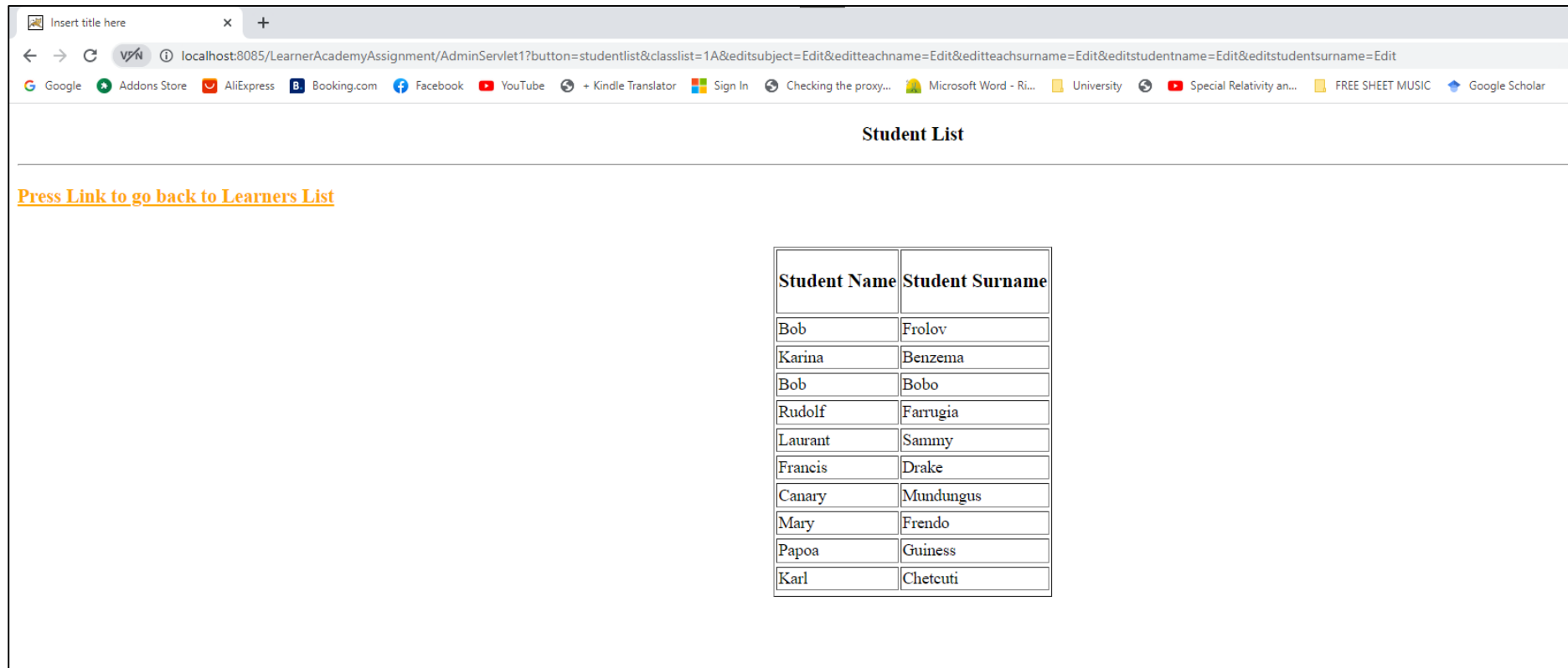


Figure 11 Shows the student List.

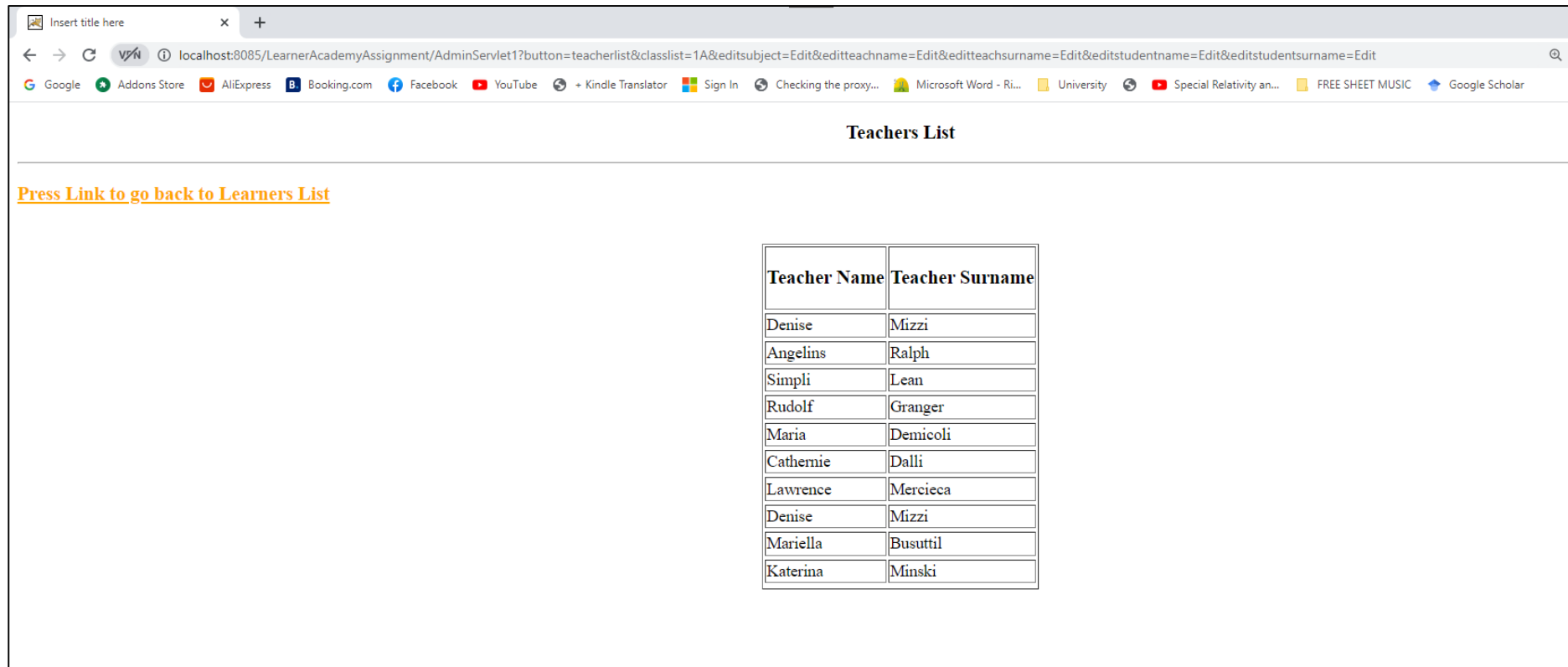


Figure 12 Shows Teacher's List.

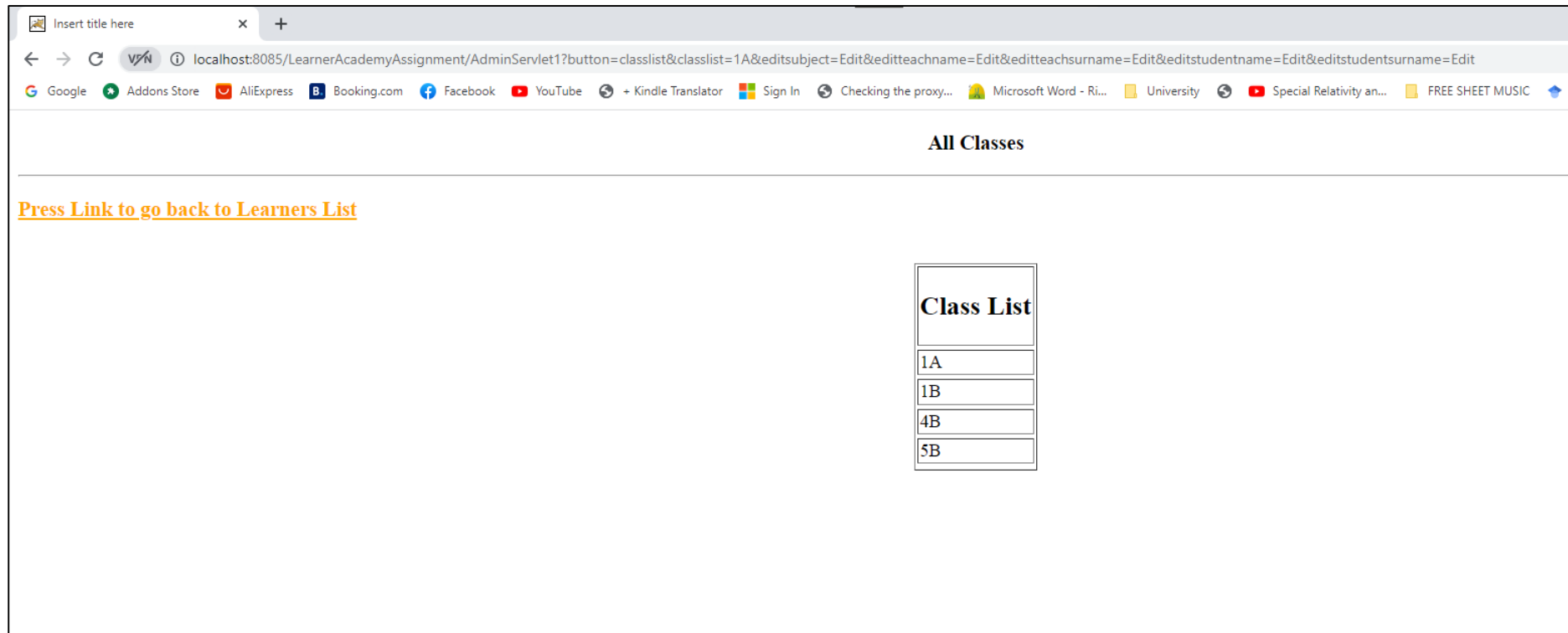
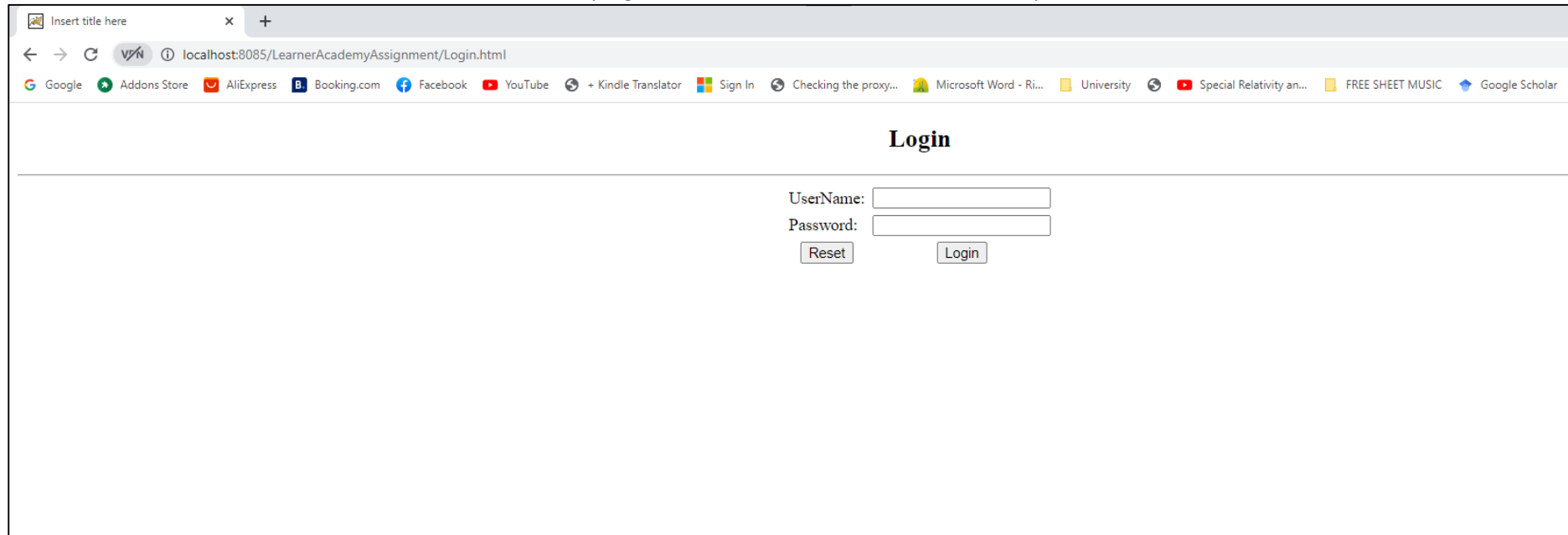


Figure 13 Shows Class List.



The screenshot shows a web browser window with the address bar displaying 'localhost:8085/LearnerAcademyAssignment/Login.html'. The page has a title 'Login' and contains a login form with the following elements:

- Username label: 'UserName:'
- Username input field:
- Password label: 'Password:'
- Password input field:
- Reset button:
- Login button:

Figure 14 Shows that Sign out button terminates the session

