

天津理工大学实验报告

学院（系）名称：计算机科学与工程学院

姓名	王帆	学号	20152180	专业	计算机科学与技术
班级	2015 级 1 班	实验项目	实验 3：扩展的 JavaScript 框架应用		
课程名称		Web 应用程序设计与开发		课程代码	0668026
实验时间		2018 年 5 月 25 日 1、2 节 2018 年 5 月 29 日 中午 5、6 节		实验地点	7-215
考核标准	实验准备（实验目的/工具熟悉情况）10 分	实验过程（实验方案可行性及步骤完整性）40 分	实验报告（实验内容丰富度与格式清晰度）30 分	实验结果（结论正确性以及分析合理性）20 分	成绩
考核内容	评价实验目的是否明确，实验工具是否清晰了解以及熟悉情况	<input type="radio"/> 可行，完整 <input type="radio"/> 可行，不完整 <input type="radio"/> 不可行，不完整	<input type="radio"/> 丰富，清晰 <input type="radio"/> 较丰富，较清晰 <input type="radio"/> 丰富，不清晰 <input type="radio"/> 不丰富，不清晰	<input type="radio"/> 结论正确，分析合理 <input type="radio"/> 结论正确，分析不充分 <input type="radio"/> 结论不正确，分析不合理	教师签字：

一、实验目的

- （1）掌握利用扩展的 JavaScript 框架构架 Web 应用程序。
- （2）掌握 EasyUI 中常用组件的使用方法。
- （3）掌握 jQuery 常用方法\$.post()的使用。
- （4）JSON 格式串和 Java 集合工具类的转换工具的使用。

二、实验环境

Windows 操作系统，Tomcat，MyEclipse，HBuilder，记事本。

三、实验要求

- 1、构建基于 Java 的 JDBC 访问数据库的类。具体要求：
 - （1）类的实现上应具有执行查询 SQL 语句的返回结果集的方法；
 - （2）能够实现执行 insert into、update 及 delete 方法；
 - （3）类应进行相关资源的释放。

(4) 构建时使用数据库连接池技术实现。

2、EasyUI 组件 ComboBox 的属性和使用，要求：

(1) 构建表，将表中的数据初始化到 ComboBox 中；

(2) 将 ComboBox 组件的值提交到 Servlet 中；

(3) 提交数据利用 jQuery 中的 \$.post() 方法。

3、EasyUI 组件数据表格组件：DataGrid 的属性和使用，具体要求：

(1) 初始化 DataGrid 组件：将表中的数据显示在 DataGrid 组件中；

(2) 获得选中 DataGrid 组件中某一行中的数据；

(3) DataGrid 控件中应带有分页功能；

(4) 当对数据库表中的数据进行修改时，能够自动刷新 DataGrid 中的数据。

四、实验过程记录（源程序、测试用例、测试结果及心得体会等）

1、构建基于 Java 的 JDBC 访问数据库的类。具体要求：

(1) 类的实现上应具有执行查询 SQL 语句的返回结果集的方法；

(2) 能够实现执行 insert into、update 及 delete 方法；

(3) 类应进行相关资源的释放。

(4) 构建时使用数据库连接池技术实现。

数据库连接池配置：

(1) 在项目：【WebRoot】-【META-INF】下：Context.xml 文件中加入如下内容：

```
<Context>
<Resource name="jdbc/DBPool" auth="Container"
    type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp2.BasicDataSourceFactory"
    username="用户名"
    password="密码"
    driverClassName="数据库驱动名"
    url="数据库连接串"
    maxTotal="100"
    maxIdle="1000"
    maxWaitMillis="5000" />
```

```
</Context>
```

(2) 在项目：【WebRoot】-【WEB-INF】下：web.xml 文件中加入如下内容：

```
<resource-ref>
    <description>DB Connection</description>
    <res-ref-name>jdbc/DBPool</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
</resource-ref>
```

数据库工具类 (DBUtil.class)

```
/**
 * @ 函数名称: getConn
 * @ 功能描述: 获取数据库连接(通过连接池)
 * @ 传入参数: 无
 * @ 返回类型: Connection
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-04-30
 * @ 版本编号: 1.00
 */
public static Connection getConn(){
    try{
        Context ctx = new InitialContext();
        DataSource ds=(DataSource) ctx.lookup("java:comp/env/jdbc/DBPool");
        conn=ds.getConnection();
    }catch(Exception e){
        e.printStackTrace();
    }
    return conn;
}

/**
 * @ 函数名称: executeBatch
 * @ 功能描述: 根据查询SQL语句进行增删改操作。
 * @ 传入参数: 用于查询的SQL语句sql
 * @ 返回类型: boolean
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-04-30
 * @ 版本编号: 1.00
 */
public static boolean executeBatch(String sql) {
    boolean flag = true; // 返回值默认为true
    try {
        conn = getConn(); // 调用getConn()方法, 初始化数据库连接
        conn.setAutoCommit(false);
        st = conn.createStatement();
        st.addBatch(sql);
        st.executeBatch();
        conn.commit(); // 执行事务
        conn.setAutoCommit(true);
    } catch (Exception ex) {
        try {
            conn.rollback(); // 事务回滚
        } catch (SQLException e) {

```

```

        e.printStackTrace();
    }
    flag = false; // 执行失败，返回false
    ex.printStackTrace();
} finally {
    finallyHandle(conn, st, rs); // 关闭数据库连接
}
return flag;
}

/**
 * @ 函数名称: executeBatch
 * @ 功能描述: 根据查询SQL语句进行增删改操作。
 * @ 传入参数: 用于查询的SQL语句list (ArrayList<HashMap<String,Object>>)
 * @ 返回类型: boolean
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-04-30
 * @ 版本编号: 1.00
 */
public static boolean executeBatch(ArrayList<String> list) {
    boolean flag = true; // 返回值默认为true
    try {
        conn = getConn(); // 调用getConn()方法，初始化数据库连接
        conn.setAutoCommit(false);
        st = conn.createStatement();
        for (int i = 0; i < list.size(); i++) {
            st.addBatch(list.get(i));
        }
        st.executeBatch();
        conn.commit(); // 执行事务
        conn.setAutoCommit(true);

    } catch (Exception ex) {
        try {
            conn.rollback(); // 事务回滚
        } catch (SQLException e) {
            e.printStackTrace();
        }
        flag = false; // 执行失败，返回false
        ex.printStackTrace();
    } finally {
        finallyHandle(conn, st, rs); // 关闭数据库连接
    }
    return flag;
}
}

```

```

/**
 * @ 函数名称: getDataSet
 * @ 功能描述: 根据查询SQL语句进行查询操作。
 * @ 传入参数: 用于查询的SQL语句sql
 * @ 返回类型: (ArrayList<HashMap<String, String>>)
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-04-30
 * @ 版本编号: 1.00
 */
public static ArrayList<HashMap<String, String>> getDataSet(String sql) {
    HashMap<String, String> hash = null;
    ArrayList<HashMap<String, String>> list = new ArrayList<>();
    ResultSetMetaData rsma = null;
    int columncount = 0;

    try {
        conn = DBUtil.getConn();
        st = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        rs = st.executeQuery(sql);
        rsma = rs.getMetaData();
        while (rs.next()) {
            hash = new HashMap<>();
            columncount = rsma.getColumnCount();
            for (int i = 1; i <= columncount; i++) {
                hash.put(rsma.getColumnName(i), rs.getString(i));
            }
            list.add(hash);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        finallyHandle(conn, st, rs);
    }
    return list;
}

/**
 * @ 函数名称: getDataSetInfoByCon
 * @ 功能描述: 根据查询SQL语句、页码及页数返回部分多条记录。
 * @ 传入参数: 用于查询的SQL语句、页码、页数
 * @ 返回类型: (ArrayList<HashMap<String, Object>>)
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-05-06
 * @ 版本编号: 1.00
 */

```

```

public static ArrayList<HashMap<String, String>> getDataSetInfoByCon(String sql, int
rowCount, int page) {
    Connection conn = null;
    ArrayList<HashMap<String, String>> result = null;
    Statement st = null;
    ResultSet rs = null;
    ResultSetMetaData rsmd = null;
    try {
        conn = getConn();
        st = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        if (rowCount > 0)
            st.setMaxRows(page * rowCount);
        rs = st.executeQuery(sql);
        if (page >= 0 && rowCount > 0)
            rs.absolute((page - 1) * rowCount);
        rsmd = rs.getMetaData();
        result = new ArrayList<HashMap<String, String>>();
        while (rs.next()) {
            int columnCount = rsmd.getColumnCount();
            HashMap<String, String> record = new HashMap<String, String>();
            for (int i = 1; i <= columnCount; i++) {
                record.put(rsmd.getColumnName(i), rs.getString(i));
            }
            result.add(record);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        finallyHandle(conn, st, rs);
    }
    return result;
}

/**
 * @ 函数名称: finallyHandle
 * @ 功能描述: 对数据库操作结束进行资源释放工作。
 * @ 传入参数: 当前连接conn、状态st、结果集rs
 * @ 返回类型: void
 * @ 文件作者: DukeWF
 * @ 创建时间: 2018-04-30
 * @ 版本编号: 1.00
 */
private static void finallyHandle(Connection conn, Statement st, ResultSet rs) {
    try {
        if (rs != null) {

```

```

        rs.close();
        rs = null;
    }
    if (st != null) {
        st.close();
        st = null;
    }
    if (conn != null) {
        conn.close();
        conn = null;
    }
} catch (Exception ex) {
    ex.printStackTrace();
}
}
}

```

演示：

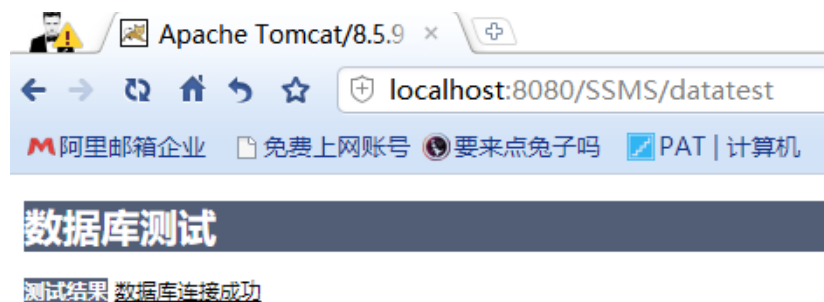


图 1 数据库测试结果

2、EasyUI 组件 ComboBox 的属性和使用，要求：

- (1) 构建表，将表中的数据初始化到 ComboBox 中；
- (2) 将 ComboBox 组件的值提交到 Servlet 中；
- (3) 提交数据利用 jQuery 中的 \$.post() 方法。

前端代码：

```

<div>
    &nbsp;姓名: <input class="easyui-combobox" id="div_name"
        style="width:150px"></input> &nbsp;所在学院: <input
        class="easyui-combobox" id="div_institute" style="width:150px"></input>
    &nbsp;<a id="query" class="easyui-linkbutton" iconCls="icon-search"
        plain="true">查 询</a>
</div>

```

JS 代码：

```

$(document).ready(function() {
    $("#div_institute").combobox({

```

```

        url : "SystemStudentService?op=institute",
        valueField : "institute_id",
        textField : "institute_name",
        panelHeight : 'auto'
    });
}

```

后端代码:

```

private void getComboBox(HttpServletResponse response, String type) {
    String result = "";
    ArrayList<HashMap<String, String>> dt = null;
    String sql;
    int rowcount = 0;
    try {
        sql = "SELECT * FROM " + type;
        dt = DBUtil.getDataSet(sql);
        result = JSON.toJSONString(dt);
        System.out.println(result);
        response.setCharacterEncoding("utf-8");
        PrintWriter out = response.getWriter();

        out.print(result);
        out.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

演示:

姓名	学号	所在学院
王帆	1	计算机科学与工程学院
李妹洁	7	机械工程学院
陈国栋	8	管理学院
陈国栋	15	海运学院
陈国栋	16	艺术学院
陈国栋	18	软件学院
陈国栋	25	外国语学院
陈国栋	26	汉语言文化学院
陈国栋	27	社会发展学院
陈国栋	20152100	环境科学与安全学院
陈国栋		理学院
陈国栋		电气电子工程学院

图 2 ComboBox 控件使用演示

3、EasyUI 组件数据表格组件：DataGrid 的属性和使用，具体要求：

- (1) 初始化 DataGrid 组件：将表中的数据显示在 DataGrid 组件中；
- (2) 获得选中 DataGrid 组件中某一行中的数据；
- (3) DataGrid 控件中应带有分页功能；
- (4) 当对数据库表中的数据进行修改时，能够自动刷新 DataGrid 中的数据。

前端代码：

```
<div id="dlgStudentInfo" class="easyui-dialog" style="width:360px;height:450px;
padding:10px 10px" closed="true" buttons="#dlg-buttons">
    <div class="ftitle">学生信息</div>
    <form id="fm" method="post" novalidate>

        <table class="fitem">
            <tr>
                <td><label class="fitem">学 号</label></td>
                <td><input type="text" id="sno" name="sno"></td>
            </tr>
            <tr>
                <td><label class="fitem">姓名</label></td>
                <td><input type="text" id="sname" name="sname"></td>
            </tr>
            <tr>
                <td><label class="fitem">年 龄</label></td>
                <td><input type="text" id="sname" name="sname"></td>
            </tr>
            <tr>
                <td><label class="fitem">政治面貌</label></td>
                <td><input class="easyui-combobox" type="text"
                    name="politicalstate" id="politicalstate" /></td>
            </tr>
            <tr>
                <td><label class="fitem">出生日期</label></td>
                <td><input type="text" id="birthday" name="birthday" /></td>
            </tr>
            <tr>
                <td><label class="fitem">地址</label></td>
                <td><input type="text" id="address" name="address" /></td>
            </tr>
            <tr>
                <td><label>联系方式</label></td>
                <td><input type="text" id="phone" name="phone" /></td>
            </tr>
            <tr>
                <td><label>学院</label></td>
```

```

        <td><input class="easyui-combobox" type="text"
            id="dlg_institute" name="dlg_institute" /></td>
    </tr>
    <tr>
        <td><label class="fitem">备注</label></td>
        <td><input type="text" id="demo" name="demo" /></td>
    </tr>
</table>
</form>
</div>

```

JS 代码:

```

$(function() {
    $('#birthday').datebox({
        required : false
    });

    $("#info").datagrid({
        loadMsg : "数据加载中, 请等待...",
        iconCls : 'icon-issue',
        nowrap : false,
        striped : true,
        collapsible : true,
        rownumbers : true,
        pagination : true,
        singleSelect : true,
        autoRowHeight : true,
        fitColumns : false,
        pageSize : 10,
        pageList : [ 10, 20, 30, 40 ],
        cache : false,
        url : "SystemStudentService?op=init"
    });

    $("#add").click(function() {
        operation = "add";
        $('#dlgStudentInfo').dialog('open').dialog('center').dialog('setTitle', '添加学生信息');
        $('#fm').form('clear');
        $("#dlg_institute").combobox({
            url : "SystemStudentService?op=institute",
            valueField : "institute_id",
            textField : "institute_name",
            panelHeight : 'auto'
        });
    });
}

```

```

        $("#politicalstate").combobox({
            url : "SystemStudentService?op=politicalstate",
            valueField : "politicalstate_id",
            textField : "politicalstate_name",
            panelHeight : 'auto'
        });
    });

    $("#edit").click(function() {
        var row = $('#info').datagrid('getSelected');
        if (row) {
            sno = row.sno;
            $('#fm').form('clear');
            operation = "edit";
            $('#dlgStudentInfo').dialog('open').dialog('center').dialog('setTitle', '编辑
学生信息');
            $("#dlg_institute").combobox({
                url : "SystemStudentService?op=institute",
                valueField : "institute_id",
                textField : "institute_name",
                panelHeight : 'auto'
            });
            $("#politicalstate").combobox({
                url : "SystemStudentService?op=politicalstate",
                valueField : "politicalstate_id",
                textField : "politicalstate_name",
                panelHeight : 'auto'
            });
            $("#sno").val(row.sno);
            $("#sname").val(row.sname);
            $("#age").val(row.age);
            $("#politicalstate").combobox("setValue", row.politicalstate);
            $("#birthday").datebox("setValue", row.birthday);
            $("#address").val(row.address);
            $("#phone").val(row.phone);
            $("#dlg_institute").combobox("setValue", row.institute);
            $("#demo").val(row.demo);
        } else {
            $.messager.alert("提示", "请选择要编辑的学生信息!", 'info');
            return;
        }
    });

    //对话框中的保存按钮操作:完成添加和编辑操作
    $("#save").click(function() {

```

```

    if ($("#sno").val() == "") {
        $.messager.alert("提示", "学生编码不能为空!", 'info');
        return;
    }

    if (operation == "add") {
        $.post("SystemStudentService?op=add",
            $("#fm").serialize(),
            function(data) {
                var value = eval('(' + data + ')');
                if (value.ret == "0") {
                    $.messager.alert("提示", "添加成功!", 'info');
                    initstudentinfo();
                    return;
                } else {
                    $.messager.alert("提示", value.reason, 'info');
                    return;
                }
            });
    } else if (operation == "edit") {
        $.post("SystemStudentService?op=edit&oldsno=" + sno,
            $("#fm").serialize(),
            function(data) {
                var value = eval("(" + data + ")");
                if (value.ret == "0") {
                    $.messager.alert("提示", "修改成功!", 'info');
                    initstudentinfo();
                    return;
                } else {
                    $.messager.alert("提示", value.reason, 'info');
                    return;
                }
            });
    }
});

$("#delete").click(function() {
    var row = $('#info').datagrid('getSelected');
    if (row == "" || row == null) {
        $.messager.alert("提示", "请选择要删除的信息!", 'info');
        return;
    }
    $.post("SystemStudentService?op=delete&sno=" + row.sno,

    function(data) {

```

```

        var value = eval("(" + data + ")")
        if (value.ret == "1") {
            $.messenger.alert("提示", "删除成功!", 'info');
            initstudentinfo();
            return;
        } else {
            $.messenger.alert("提示", "删除失败! " + value.reason, 'info');
            return;
        }
    });
});

$("#close").click(function() {
    $('#dlgStudentInfo').dialog('close');
    return;
});

function initstudentinfo() {
    $("#studentinfo").datagrid({
        loadMsg : "数据加载中, 请等待...",
        url : 'SystemStudentService?op=init&con='
    });
}
})

```

后端代码:

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // TODO Auto-generated method stub
    String op = request.getParameter("op");
    System.out.println(op);

    switch (op) {
        case "politicalstate":
            getComboBox(response, "politicalstate");
            break;

        case "institute":
            getComboBox(response, "institute");
            break;

        case "student":
            getComboBox(response, "student_simple");
            break;
    }
}

```

```

        case "init":
            String row = request.getParameter("rows");
            String page = request.getParameter("page");
            getStudentInfo(response, "", page, row);
            break;
        default:
            break;
    }
}

private static String getStudentInfo(HttpServletResponse response, String con, String page,
String row) {
    String result = "";
    Map<String, Object> map = new HashMap<String, Object>();
    ArrayList<HashMap<String, String>> dt = null;
    String sql;

    int rowcount = 0;
    if (con == null)
        con = "";
    if (row == null)
        row = "0";
    if (page == null)
        page = "0";
    try {
        int r = Integer.parseInt(row);
        int p = Integer.parseInt(page);
        if (!con.equals("")) {
            sql = "select * from student where " + con;
        } else {
            sql = "select * from student";
        }
        dt = DBUtil.getDataSetInfoByCon(sql, r, p);
        rowcount = DBUtil.getRowCount(sql);
        map.put("total", rowcount);
        map.put("rows", dt);
        result = JSON.toJSONString(map);
        response.setCharacterEncoding("utf-8");
        PrintWriter out = response.getWriter();

        out.print(result);
        out.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return result;
}

```

}

演示：

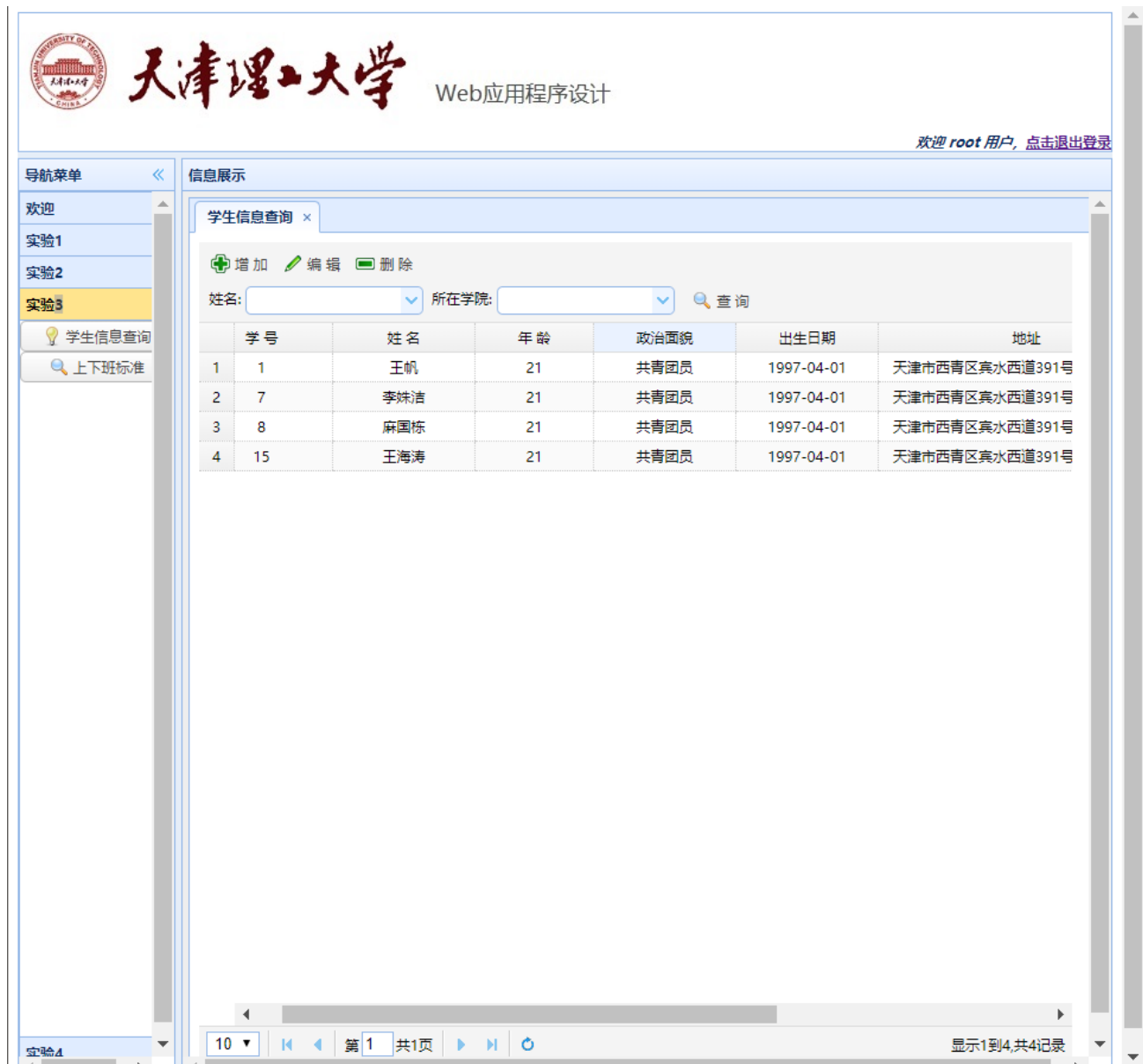


图 3 DataGrid 控件使用演示

心得体会：

通过本次实验，我初步了解了利用扩展的 JavaScript 框架构架 Web 应用程序的方法，并学习掌握了 EasyUI 中常用组件的使用方法，包括下拉列表（ComboBox）与表格（DataGrid）控件。此外我也掌握 jQuery 常用方法的使用，并对 JSON 格式串和 Java 集合工具类的转换工具的使用有了一定的了解。Web 开发包括前后端与数据库等方面，通过前三个实验的学习，我能够初步实现一个有较好交互界面的 Web 应用程序。虽然还有很多问题亟需解决，但在学习的过程中我的编程能力有了较好的提升。这对于日后的编程课程与其他专业课程的学习，乃至日后的工作与科研，都有着较为关键的帮助。