实验 11：实现成绩管理

一、实验目的

1、掌握 Vue 的使用；

2、掌握 Vuex 的使用。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

（1）完成基于 Vuex 的学生成绩管理系统的开发

六、实验内容及步骤

步骤 1：创建 Vue 项目。

```bash
npm create vue@latest
```

步骤 2：安装依赖

```bash
npm install vuex vuex-persistedstate
```

步骤 3：创建 Vuex 存储模块

在 src 目录下创建 store 文件夹，然后在该文件夹中创建 index.js 文件，代码示例如下。

```javascript
import Vue from 'vue';
import Vuex from 'vuex';
import createPersistedState from 'vuex-persistedstate';

Vue.use(Vuex);

export default new Vuex.Store({
  plugins: [createPersistedState()],
  state: {
    students: []
  },
  mutations: {
    addStudent(state, student) {
      state.students.push(student);
    },
    deleteStudent(state, index) {
      state.students.splice(index, 1);
    }
  }
});
```

步骤 4：挂载 Vuex 存储

在 src 目录下的 main.js 文件中，挂载 Vuex 存储。

JavaScript

```
import Vue from 'vue';
import App from './App.vue';
import store from './store';

new Vue({
  store,
  render: h => h(App)
}).$mount('#app');
```

步骤 5：编写主应用组件

打开 src 目录下的 App.vue 文件，编写主应用组件。

XML/HTML

```
<template>
  <div id="app">
    <h1>学生成绩管理系统</h1>
    <input v-model="newStudent.name" placeholder="学生姓名" />
    <input v-model.number="newStudent.grade" placeholder="学生成绩" type
="number" />
    <button @click="addStudent">添加学生</button>
    <ul>
      <li v-for="(student, index) in students" :key="index">
        {{ student.name }}: {{ student.grade }}
        <button @click="deleteStudent(index)">删除</button>
      </li>
    </ul>
  </div>
</template>

<script setup>
import { ref } from 'vue';
import { useStore } from 'vuex';

const store = useStore();

// 响应式数据
const newStudent = ref({
  name: '',
  grade: 0
});

// 获取学生列表
const students = store.state.students;

// 添加学生方法
const addStudent = () => {
  if (newStudent.value.name && newStudent.value.grade) {
    store.commit('addStudent', { ...newStudent.value });
    newStudent.value = { name: '', grade: 0 }; // 重置表单
  }
};

// 删除学生方法
const deleteStudent = (index) => {
  store.commit('deleteStudent', index);
```

```
};
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

步骤 6：继续完善项目

请在此项目的基础上持续完善项目，具体要求如下：

- 在界面上添加筛选功能，允许用户根据成绩范围（如大于某个分数、小于某个分数、在某个分数区间）筛选学生成绩。
- 添加排序功能，允许用户按照成绩升序或降序排列学生列表。
- 允许用户编辑已有的学生成绩信息。
- 在添加和编辑学生信息时，对输入的数据进行验证，确保姓名和成绩的合法性。如果输入不合法，给出相应的错误提示。
- 当学生列表数据较多时，添加分页功能，每页显示一定数量的学生信息。

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

（1）学生演示汇报实验内容的完成情况，实验指导老师现场打分。

（2）指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。

（3）实验考核结束后，进行公布成绩。