



학생성적 관리 프로그램



교육과정 2개월차 (2월26일~2월28일)

CONTENTS

1

프로젝트 개요

2

주요 기술

3

요구 분석

4

실행 화면

5

프로젝트 후기

프로젝트 개요

1

본 프로젝트는 교육 2개월차에 만든 간단한 학생 성적 관리 프로그램입니다.



2

본 프로젝트는 JavaScript 수업을 듣고 만든 프로그램이며
사용자는 학생의 정보와 점수를 입력하고 등록할 수 있으며 삭제도 가능합니
다. 그리고 조건에 따라 정렬을 할 수 있고 이름으로 검색이 가능합니다.
학생들의 평균과 순위를 집계 할 수 있습니다.

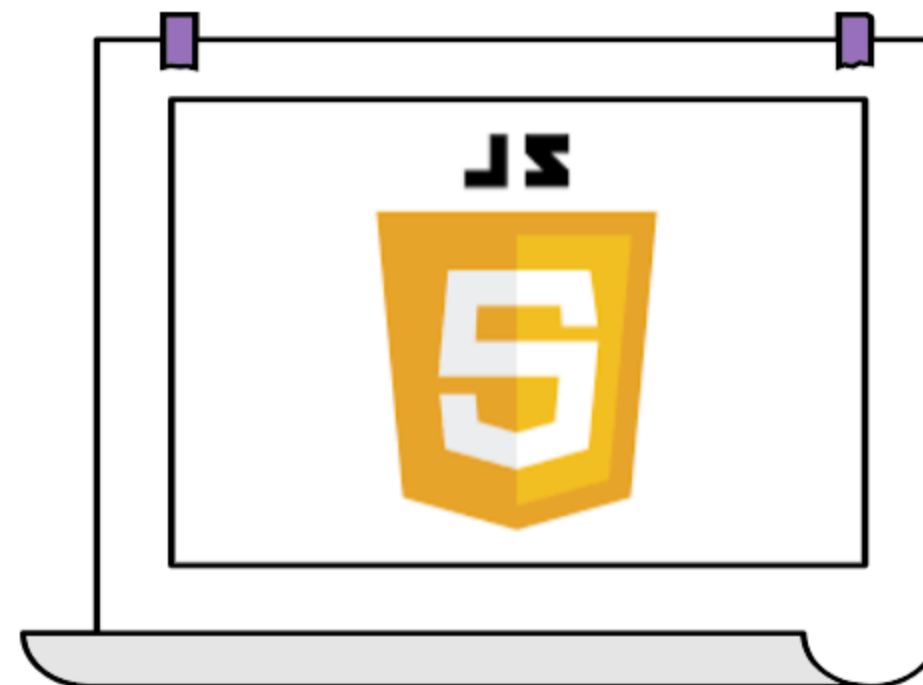


주요 기술



1 화면 구성

HTML, CSS, BOOTSTRAP을 사용했습니다



2 코드 구현

1. JavaScript DOM처리를 하였습니다.
2. 그외 JavaScript기본문법 사용 했습니다.
3. Validator 함수를 만들어서 유효성 검증을 했습니다.

요구 분석

1

새로운 학생 등록&삭제

사용자는 학생의 학번, 이름과 함께 국어,영어,수학 점수를 입력하고 등록버튼을 누르면 등록이 완료 됩니다.

학번과 점수는 반드시 숫자로 입력되게끔 유효성 검증이 되어야 합니다.



2

등록된 학생 검색기능

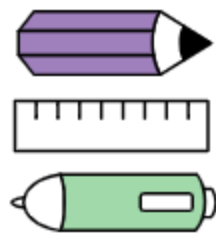
검색창에 학생의 이름을 정확히 입력하면 학생 목록에 해당 학생이 검색이 됩니다. 만약 등록된 학생이 없다면 등록된 학생이 없다는 창이 나옵니다.



3

등록된 학생의 순위 및 평균 기능

등록된 학생들의 점수를 기준으로 과목별 평균과 순위가 나오고 전체 학생의 총점과 평균이 나와야 합니다.



4

등록된 학생의 정렬기능

등록된 학생들은 학번,이름,총점으로 정렬이 가능합니다. SelectBox에서 원하는 정렬기준을 선택하면 정렬이 되어야 합니다.



실행 화면

실행 초기 화면

웹이 실행이 호출이 되면 화면 상단에
학번, 이름을 포함한 점수를 입력하는 공간이 있습니다.
학번과 점수는 숫자만 입력이 가능하게 만들었습니다. 만약
숫자가 기입되지 않았다면 alert창이 나옵니다.
화면 아래쪽엔 학생들의 목록이 나오고 사용자가 원하는 기준
으로 학생들을 정렬 할 수 있습니다.



이젠초등학교 성적 관리

학번	<input type="text"/>	이름	<input type="text"/>
국어점수	<input type="text"/>	영어점수	<input type="text"/>
수학점수	<input type="text"/>	<input type="button" value="등록"/>	<input type="button" value="삭제"/>

현재 정렬 조건	정렬 ▼	이름으로 검색하세요	<input type="button" value="검색"/>
----------	------	------------	-----------------------------------

학번	이름	국어	영어	수학	총점	평균	순위
10	신연재	100	100	100	300	100.0	1
11	이연재	95	100	90	285	95.0	2
2	홍길동	90	100	50	240	80.0	3
3	임꺽정	55	90	90	235	78.3	4
전체 평균					1060	88.3	

학번	<input type="text" value="이번"/>	이름	<input type="text"/>
국어점수	<input type="text"/>	영어점수	<input type="text"/>
수학점수	<input type="text"/>	<input type="button" value="확인"/>	

localhost:5502 내용:

학번을 숫자로 입력하여 주세요

실행 화면

실행 초기 화면

화면은 DOM을 이용하여 구성하였고 테이블을 body에 innerHTML을 이용하여 넣었습니다.

학생들의 순위를 정의하는건 assignRank함수를 사용하고 학생 테이블을 화면에 나오게하는건 renderingStudents 함수를 사용 하였습니다.

```
// 학생 전체 목록 출력
const students = studentService.findAll();
assignRank(students);
renderingStudents(students);
});
```

```
const renderingStudents = function (students) {
  const tbody = document.querySelector(".studentList");
  const tfoot = document.querySelector('table tfoot[name=table-tfoot]');

  let trs = "";
  let tfootContents = '';
  let total = 0;
  let average = 0.0;
  students.forEach(student => {
    trs += `
      <tr>
        <th>${student.ssn}</th>
        <td>${student.name}</td>
        <td>${student.korean}</td>
        <td>${student.english}</td>
        <td>${student.math}</td>
        <td>${student.getSum()}</td>
        <td>${student.getAverage()}</td>
        <td>${student.rank}</td>
      </tr>
    `;
    total = total + student.getSum();
    average = average + parseFloat(student.getAverage());
  });
  tbody.innerHTML = trs;
  students.forEach(student => {
    tfootContents = `
      <tr>
        <th scope="row"></th>
        <th colspan="4"><class="fa-solid fa-school">전체 평균</th>
        <td>${total}</td>
        <td>${(average / students.length).toFixed(1)}</td>
        <td></td>
      </tr>
    `;
  });
  tfoot.innerHTML = tfootContents;
}
```

기타 코드

순위정하기

1. 학생 객체 배열을 점수 합계 기준으로 내림차순으로 정렬합니다.
2. rank 변수를 1로 초기화합니다.
3. 학생 객체를 순회하면서 첫 번째 학생이라면 순위를 1로 설정합니다. 다음 학생부터는 이전 학생의 점수 합계와 다르면 현재 인덱스 + 1 값을 순위로 설정합니다. 학생 객체에 rank 속성을 추가하여 순위를 할당합니다.
4. 이 과정을 통해 점수가 높은 학생에게 낮은 순위를 부여하고, 동점자는 동일한 순위를 갖게 됩니다.

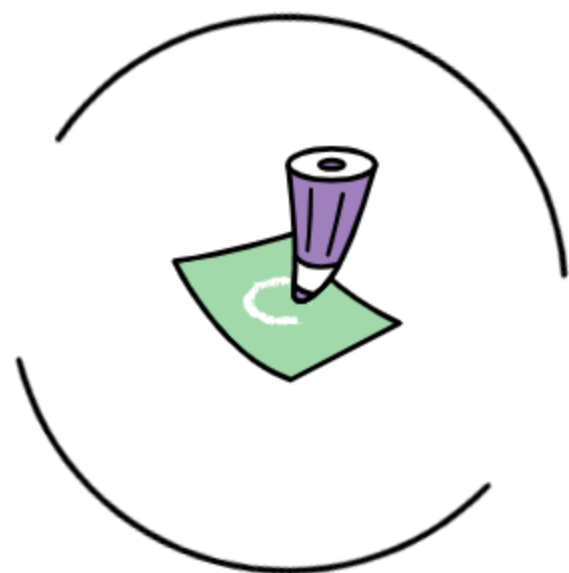
학생 목록 정렬

1. studentService.findAll()을 호출하여 모든 학생 객체를 가져옵니다.
2. "ssn": 학생들의 학번을 기준으로 정렬합니다. SSN을 정수로 변환한 후, 오름차순으로 정렬합니다.
"name": 학생들의 이름을 기준으로 정렬합니다. 이름의 첫 번째 문자의 유니코드 값을 비교하여 오름차순으로 정렬합니다.
"sum": 학생들의 점수 합계를 기준으로 정렬합니다. 점수 합계를 내림차순으로 정렬합니다.
3. 정렬된 학생 목록을 renderingStudents 함수를 호출하여 렌더링합니다.

```
const assignRank = function(students) {
  students.sort((a, b) => b.getSum() - a.getSum());
  let rank = 1;
  students.forEach((student, index) => {
    if (index > 0 && student.getSum() !== students[index - 1].getSum()) {
      rank = index + 1;
    }
    // 학생 객체에 순위(rank)를 추가
    student.rank = rank;
  });
};
```

```
const sortStudents = function(studentService, type) {
  const students = studentService.findAll();
  switch (type) {
    case "ssn":
      students.sort((student1, student2) => parseInt(student1.ssn) - parseInt(student2.ssn));
      break;
    case "name":
      students.sort((student1, student2) => student1.name.charCodeAt(0) - student2.name.charCodeAt(0));
      break;
    case "sum":
      students.sort((student1, student2) => student2.getSum() - student1.getSum());
      break;
  }
  renderingStudents(students);
}
```


프로젝트 후기



Java를 약 두달간 공부를 하고 처음으로 JavaScript를 배워서 진행한 프로젝트였습니다. 이름이 비슷해서 느낌도 비슷할 것 같다는 제 생각과는 달리 Java와 조금 다른 부분이 많아서 처음엔 조금 당황스러웠지만 배우면 배울수록 자유롭고 JavaScript만의 느낌이 아주 흥미롭고 재미있었습니다. Java의 클래스기반의 프로젝트를 하다가 함수기반의 프로젝트를 진행하는것에도 매우 흥미를 느꼈고 웹 화면에 제가 제작한 내용이 나올때 큰 희열을 느꼈습니다. 나중에 시간이 된다면 JavaScript를 조금 자세히 공부를 해서 고급기능들을 구현해 보고싶다는 생각을 하였습니다.