

IPP현장실습 (9)월 현장실습 일지

학부(과)	컴퓨터 소프트웨어학과	학년	4	학번	2016726001
지도교수	고광용	실 습 학생명	원종진	지 도 담당자	권시현 ^{권인철}
실습기관명	볼트마이크로		NCS세분류	응용SW엔지니어링	
실습기간	2022년 7월 1일 ~ 2022년 12월 31일 까지 (27주간)				
기간	주요 업무				
1주차	실습내용	기존 스코어보드 React 포팅 작업			
	교육내용	React 포팅 관련 교육			
2주차	실습내용	React로 포팅한 스코어보드와 Realtime DB 연결 작업, 컨트롤 페이지 타이머 기능 추가,			
	교육내용	기존 realtime DB 구조와 스코어보드의 간의 연결 방법 공유			
3주차	실습내용	스코어보드 숨기기/보이기 기능 추가, 미식축구 컨트롤 페이지 추가			
	교육내용	기존 컨트롤 페이지 코드 리뷰 및 스키마			
4주차	실습내용	테니스 컨트롤 페이지 추가			
	교육내용	테니스 컨트롤 페이지 추가 관련 코드 리뷰 및 피드백			
5주차	실습내용	기존 컨트롤 페이지 버그 및 오류 수정 작업, 경기 페이지에서 컨트롤 페이지 이동 버튼 구현			

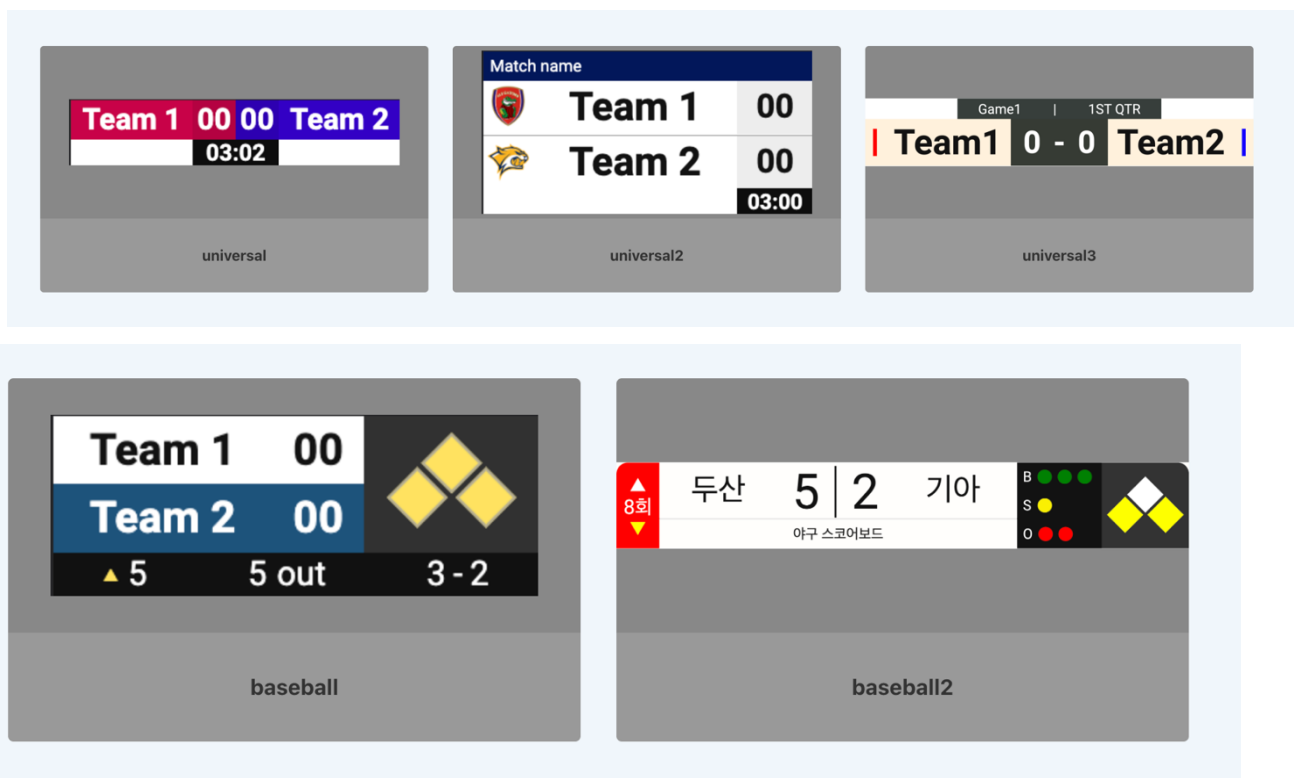
	교육내용	기존 경기 목록 페이지 코드 스키마 및 리뷰
--	------	--------------------------

현장실습 주요내용

기존 스코어보드 React 포팅 작업

기존에 Vanilla Javascript로 작성되었던 스코어보드들을 지난 팀 회의를 통해 React로 전면 변경하기로 결정됨에 따라 기존 스코어보드들을 react로 포팅하는 업무 배정

1. React 포팅 작업한 스코어보드 목록
 - Universal1, Universal2, Universal3
 - Baseball1, Baseball2
 - Tennis_wimbledon, Tennis_australian
 - billiard
 - football
 - setScoreboard
 - golf_lpga
2. React로 포팅된 기존 스코어보드 미리보기 화면





React로 포팅한 스코어보드와 Realtime DB 연결 작업

React로 포팅한 스코어보드에 realtime DB의 데이터를 적용하기 위한 연결 작업 배정

1. React 포팅 작업한 스코어보드 목록
 - Universal1, Universal2, Universal3
 - Baseball1, Baseball2
 - Tennis_wimbledon, Tennis_australian
 - billiard
 - football
 - setScoreboard
 - golf_lpga

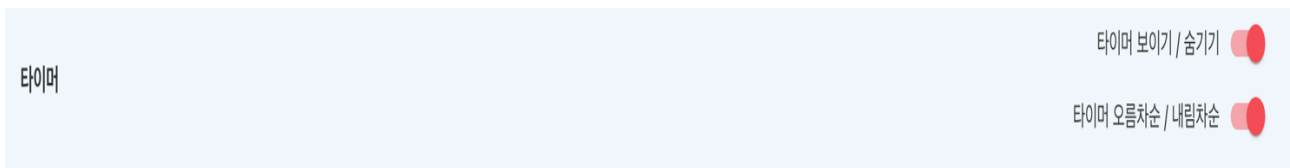
컨트롤 페이지 타이머 기능 추가

여러가지 타이머 기능을 추가 하기 위한 작업 배정

1. 기능 및 구현 해야할 To Do List
 - 타이머 시작 시간 설정 label 추가
 - 타이머 +5, +15, +45, Reset 버튼 추가
 - 타이머 시작 시간을 45:00으로 설정하면 축구 경기에서 후반전이 시작하듯 45:00부터 타이머가 흘러가는 기능



- 시간 count/up down 토글 버튼 추가 및 기능 구현
 - 농구 경기 같은 경우 10:00으로 시작하여 00:00으로 타이머가 내림차순으로 진행되므로 타이머의 오름차순/내림차순 기능 추가

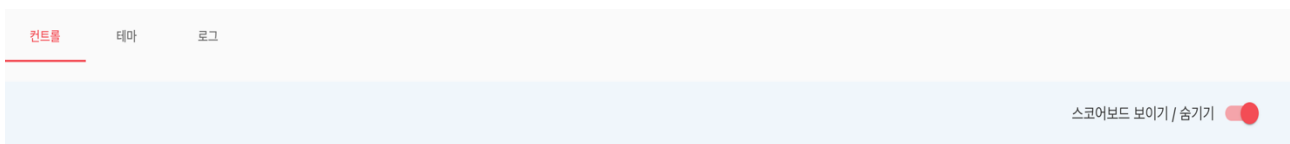


- 타이머 시작 시간 설정 DB 연결

컨트롤 스코어보드 보이기/숨기기 기능 추가

버튼에 따라 스코어보드를 보이게 하거나 숨길 수 있게 하는 기능 추가 업무 배정

1. 스코어보드 보이기/숨기기 토글 버튼 추가



2. 토글 버튼 조작에 따라 realtime DB 업데이트
3. Realtime DB 데이터에 따라 스코어보드 숨기기 또는 보이기 기능 구현

4. 결과 화면

- 보이기



- 숨기기



미식축구 컨트롤 페이지 추가

미식축구 컨트롤 페이지 추가를 위한 ui 작업 업무 배정

1. 추가해야할 컴포넌트 목록

- 팀 timeout 컨트롤 추가
- 경기 period 컨트롤 추가
- 공격 possession. 컨트롤 추가
- forthDown 컨트롤 추가

- 팀 컬러 선택 컨트롤 추가

2. 제작된 미식축구 컨트롤 페이지 화면

스코어보드 보이기 / 숨기기 ☒

스코어 설정

홈 팀

-

12

+

원정 팀

-

18

+

점수

타임아웃

-

3

+

타임아웃

-

3

+

피리어드

1ST

2ND

3RD

4TH

OT

공격권

HOME

AWAY

포스다운

1ST

2ND

3RD

4TH

남은 거리

-

16

+

컬러

적용

:

적용

테니스 컨트롤 페이지 추가

테니스 컨트롤 페이지 추가를 위한 ui 작업 업무 배정

1. 추가해야할 컴포넌트 목록 및 기능 구현 목록
 - 서브권 컨트롤 추가

- Set 컨트롤 추가
- Game 컨트롤 추가
- Point 컨트롤 추가
- Point to game 연동 로직 추가
- Point 듀스 시 setValue 예외처리
- Game to set 연동 로직 추가
- 데이터 reset 추가
- 점수 자동 연동 함수 최소화 작업

2. 제작된 테니스 컨트롤 페이지 화면

- 테니스 컨트롤 페이지는 2개의 테마가 같은 컨트롤 페이지를 사용할 수 있도록 구현
- 자동 점수 연동 로직으로 사용자의 편리성 최대화

3. 화면

The screenshot displays a '스코어 설정' (Score Setting) interface. At the top right, there is a toggle switch for '스코어보드 보이기 / 숨기기' (Show/Hide Scoreboard). The main area is divided into two columns: '홈 팀' (Home Team) and '원정 팀' (Away Team). Under each team header, there are three rows of controls: '포인트' (Point), '게임' (Game), and '서브권' (Serve). Each row contains a red minus button, a central display showing '0', and a red plus button. A colon separator is placed between the two teams' controls. At the bottom, there is a '초기화' (Reset) button with the text '테니스 데이터 초기화' (Reset Tennis Data).

기존 컨트롤 페이지 오류 수정 작업

기존 컨트롤 페이지의 오류 및 버그 검출 작업 및 수정 작업 업무 배정

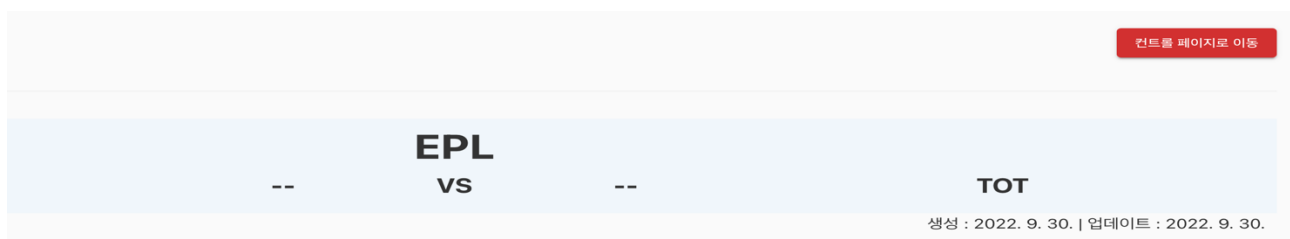
1. 검출된 오류 목록
 - 팀컬러 설정 후 페이지가 다시 로드되면 설정 되어있는 팀컬러가 컨트롤에 적용되지 않는 오류

- 스코어보드 하단 구분라인 추가
- 모바일에서 타이머가 보이지 않는 오류 수정
- 팀 컬러 색깔을 선택하지 않고 적용을 누르면 색깔이 사라지는 오류 수정
- 스코어보드 react 페이지 오류 및 버그 및 스타일 수정
- 테니스 듀스의 경구 상대가 AD일 때 자신의 점수를 + 하면 상대의 점수만 내려가야 하는데 자신의 점수 또한 플러스 되는 오류 수정

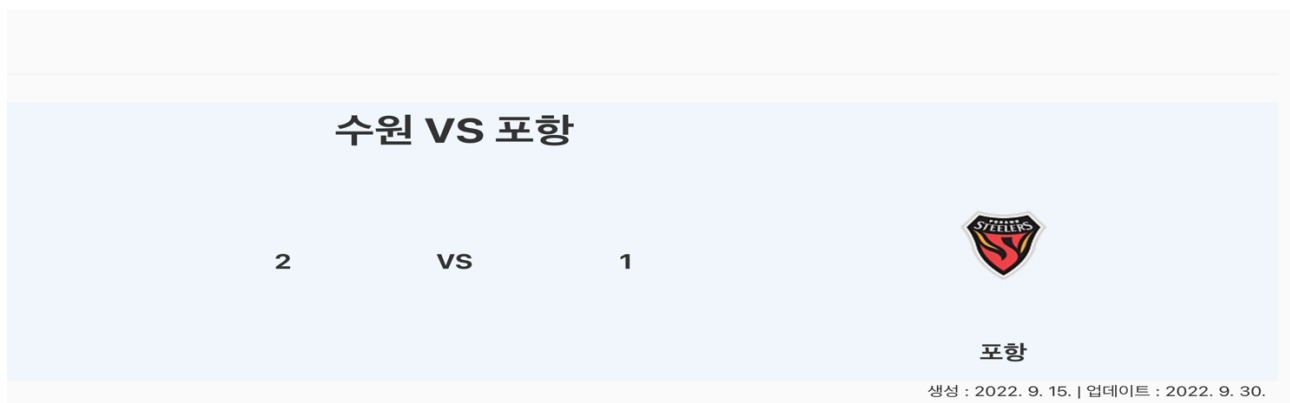
경기 페이지에서 컨트롤 페이지 이동 버튼 구현

추후 컨트롤 페이지로의 이동은 자신 생성한 경기 페이지에서만 이동이 가능하도록 변경될 것이기 때문에 그에 따른 페이지 이동 버튼 구현 담당 업무 배정

- 구현해야 할 기능 및 To Do List
 - 컨트롤 페이지 이동 버튼 추가
 - 선택된 경기가 사용자 자신의 경기일 때만 컨트롤 페이지 이동 버튼이 생성되도록 구현
 - 컨트롤 페이지 접속 시 경기 목록의 경기 데이터를 컨트롤 페이지로 전송
 - 컨트롤 페이지에서 받은 경기 데이터를 컨트롤 탭에 적용
 - 뒤로가기도 될 수 있도록 router 사용
- 구현 된 화면
 - 경기 페이지가 자신이 생성한 페이지가 아니라면 컨트롤 페이지 이동 버튼이 나타나지 않음
 - 자신이 생성한 경기 페이지



- 자신이 생성하지 않은 경기 페이지



9월 실습을 통하여 느낀점

여러가지 버그 검출 과정을 통하여 자신이 코드를 짤 때는 버그가 없다고 생각할 수 있지만 자신도 모르는 버그가 여러 곳에 숨겨져 있는만큼 처음 코드를 짤 때 버그가 검출되지 않도록 잘 짜야한다는 것을 느꼈다. 또한 react로 포팅하는 과정에서 나의 코드나 함수들이 성능에 있어서 성능을 저하시킬 수 있는 로직이 많음을 내 코드를 다시 봄으로써 느낄 수 있었고 그에 따라 코드를 처음 짤 때 기능의 구현도 중요하지만 성능이나 최적화 문제 또한 많은 중요도가 있고 그에 있어서 많은 고민이 필요하다는 것을 느꼈다.