최종 보고서

과제	PitchMatch: 축구 필드에서 펼쳐지는 경기 같은 커뮤니티 서버 개발
수업	웹서버프로그래밍(7-9)
지도교수	정복래 교수님
팀	Team CSS(6 조)
연구참여자	팀장 김찬영(20190895) : 프로젝트 총괄 및 보고서 작성
	팀원 지성원(20190948) : 프로그래밍, 디버깅 및 오류 수정
	팀원 우상욱(20190919) :알고리즘 작성, 자료 조사
제출일자	2023.06.08

PitchMatch Team CSS

1. 설계과제 제목

PitchMatch: 축구 필드에서 펼쳐지는 경기 같은 커뮤니티 서버 개발

2. 설계과제 추진 목적

1) 웹 서버 프레임워크를 이용한 서버 구축을 통해 백엔드 프로그래밍 능력 향상

2) Node.JS 를 이용한 자바스크립트 서버 개발

3) MySQL 를 이용한 데이터베이스 구축 및 RDBMS 활용 능력 향상

4) API 를 제공받아 데이터 처리 능력 향상

5) 여러 축구 데이터를 이용한 파워랭킹 설계 및 알고리즘 설계 능력 향상

6) 축구팬 유저들에게 축구 관련 커뮤니티 기능 제공

3. 설계과제의 필요성

1) 2020 년 9 월 악성 댓글을 차단한다는 목적으로 NAVER 스포츠 기사 댓글 서비스를 폐지하였으며, 2023 년 3 월 기존에 작성된 스포츠 기사의 모든 댓글 DB 를 삭제하였다. 대형 포털에서 제공하던 축구팬들의 소통 공간의 부재로 인해 축구 커뮤니티의 필요성을 느꼈다.

2) 이번 2023 K 리그 1 득점은 전년 대비 33% 증가하였으며, 코로나 19 영향력이 컸던 지난해와 직접적인 비교는 어렵겠지만, 관중 또한 지난해에 비해 3.2 배 증가하였다. 갈수록 커져가는 국내 축구관련 시장의 성장세에 여러 컨텐츠를 제공할 수 있는 커뮤니티 및 서버를 제공할 수 있으면 좋겠다고 생각하였다.

3) 해외 축구 시장의 경우, AR 관련 신기술을 이용한 중계 시스템의 성장 및 축구에 관한 데이터를 여러 기준과 지표를 설정하여 분석하는 스포츠 과학 시장이 크게 증가하고 있다. 이에 편승하여 국내 축구 컨텐츠 서비스에도 분석 방식의 다양성을 위해 본 프로젝트의 목적을 설정하였다.

4. 설계과제의 목표

Node.JS 를 기반으로 사용자에게 축구 관련된 정보를 제공하는 것을 주기능으로 갖는 웹 서버를 개발한다. 프로젝트 제목은 'PitchMatch'로 설정했으며, 사용자들에게 축구 관련된 소식, 정보를 제공하며, 커뮤니티의 기능도 수행할 수 있는 웹서버를 개발하는 것을 목적으로 한다. 구현하고자 하는 기능의 목록은 다음과 같다.

1) Node.JS와 웹 서버 프레임워크를 이용한 커뮤니티 사이트를 구현한다. 커뮤니티의 기능으로는 로그인 기능, 통합 게시판, 게시물 작성, 댓글 작성, 사이트 내에서만 이용이 가능한 포인트 획득 등 기본적인 커뮤니티가 제공하는 기능들을 포함한다.

2) 축구 구단 별 게시판을 만들어 각 팀의 팬들끼리 소통할 수 있는 게시판을 구현한다.

3) 축구 정보를 제공해주는 서버(fotmob.com)로부터 API 를 제공받아 축구 구단 및 선수에 대한 데이터를 사용한다.

4) 각 축구 구단마다 점수(Rating)를 부여하는 알고리즘을 작성한다. 이 알고리즘을 통해 부여되는 점수는 단순히 구단의 리그 내 순위가 아닌, 실질적인 그 구단의 파워를 나타내는 것을 목표로 한다. 점수에 영향을 끼치는 파라미터에는 최근 경기의 성적 흐름, 현재 시즌의 성적 등으로 설정한다.

5) 실제 진행될 매치 데이터를 바탕으로 승부예측에 도움이 될 만한 기능을 제공한다. 각 팀의 Rating 을 표시해주고, 양 팀의 상대전적을 5 경기 데이터를 제공한다.

6) 각 구단의 이적시장 정보를 제공하는 기능을 구현한다. 제공되는 정보에는 선수명, 어느 팀에서 어느 팀으로 이적하였는지, 이적료(이적 형태) 등이 있다.

7) 실시간 경기 진행 시 그 경기에 대한 채팅방을 개설하여 사용자들이 경기에 대하여 소통할 수 있도록 기능을 제공한다.

현실적 제한 요소	내용
경제	(1) 서버 트래픽 과다 시를 고려한 안정적인 서버 장비 확보 (2) 낭비되는 리소스 없이 최적의 리소스만을 사용하여 웹 서버를 구현해 경제성 고려
편리	(1) 사용자가 서버에 접속했을 시 최대한 사용자의 가독성을 고려하여 개발 (2) UI 디자인 및 사이트 내 요소들의 가독성 및 직관성
윤리	(1) 사이트 내 승부예측 기능의 경우 사행성 요소로 번질 가능성이 있음 (2) 관련 기준에 따른 윤리성 고려
사회	(1) 축구 관련 소식이 아닌 정보는 제공하지 않음 (2) 다른 사용자들에게 피해나 불쾌감을 줄 수 있는 데이터를 방지

5. 설계과정

5.1 설계 기초이론

5.1.1 JavaScript : 백엔드 개발에 JavaScript 를 사용하였음. 자바스크립트는 웹페이지를 동적으로 만들고 상호작용하는 데 사용되는 언어로, HTML/CSS 와 함께 웹 페이지 서비스를 제공하게 해 준다.

5.1.2 Node.JS : Node.js 는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임 환경이다. Node.js 는 서버 측 애플리케이션을 개발하기 위해 사용되며, 웹 브라우저가 아닌 컴퓨터의 로컬 환경에서 JavaScript 코드를 실행할 수 있도록 한다. Node.js 는 이벤트 기반, 비차단 I/O 모델을 채택하여 높은 성능을 제공하며, 이는 여러 작업을 동시에 처리할 수 있는 능력을 갖추고 있으며, I/O 작업이나 네트워크 요청과 같은 비동기 작업에 특히 효과적이다. 이러한 특성으로 인해 Node.js 는 대규모 실시간 애플리케이션, 웹 서버, API 서버, 마이크로서비스 등 다양한 종류의 서버 측 애플리케이션 개발에 많이 사용된다.

5.1.3 Express 프레임워크 : Express 는 Node.js 를 위한 빠르고 유연한 웹 애플리케이션 프레임워크이다. Express 는 Node.js 의 핵심 기능을 보완하고, 웹 및 모바일 애플리케이션을 개발하기 위한 강력한 도구와 기능을 제공한다. Express 는 간결하고 직관적인 API 를 가지고 있어 개발자가 웹 애플리케이션을 쉽게 작성할 수 있도록 도와주는데, 라우팅, 미들웨어, 요청 및 응답 처리 등의 핵심 기능을 제공하며, 이를 통해 웹 애플리케이션의 라우팅 및 로직을 구성할 수 있다. Express 는 다양한 미들웨어를 지원하여 개발자가 웹 애플리케이션의 동작을 수정하고 확장할 수 있게 해 준다. 미들웨어는 요청과 응답 사이에서 동작하며, 로그 기록, 세션 관리, 사용자 인증, 데이터 변형 등 다양한 작업을 수행할 수 있다. 또한, 개발자가 필요에 따라 Express 애플리케이션에 다른 Node.js 패키지나 모듈을 쉽게 통합할 수 있다.

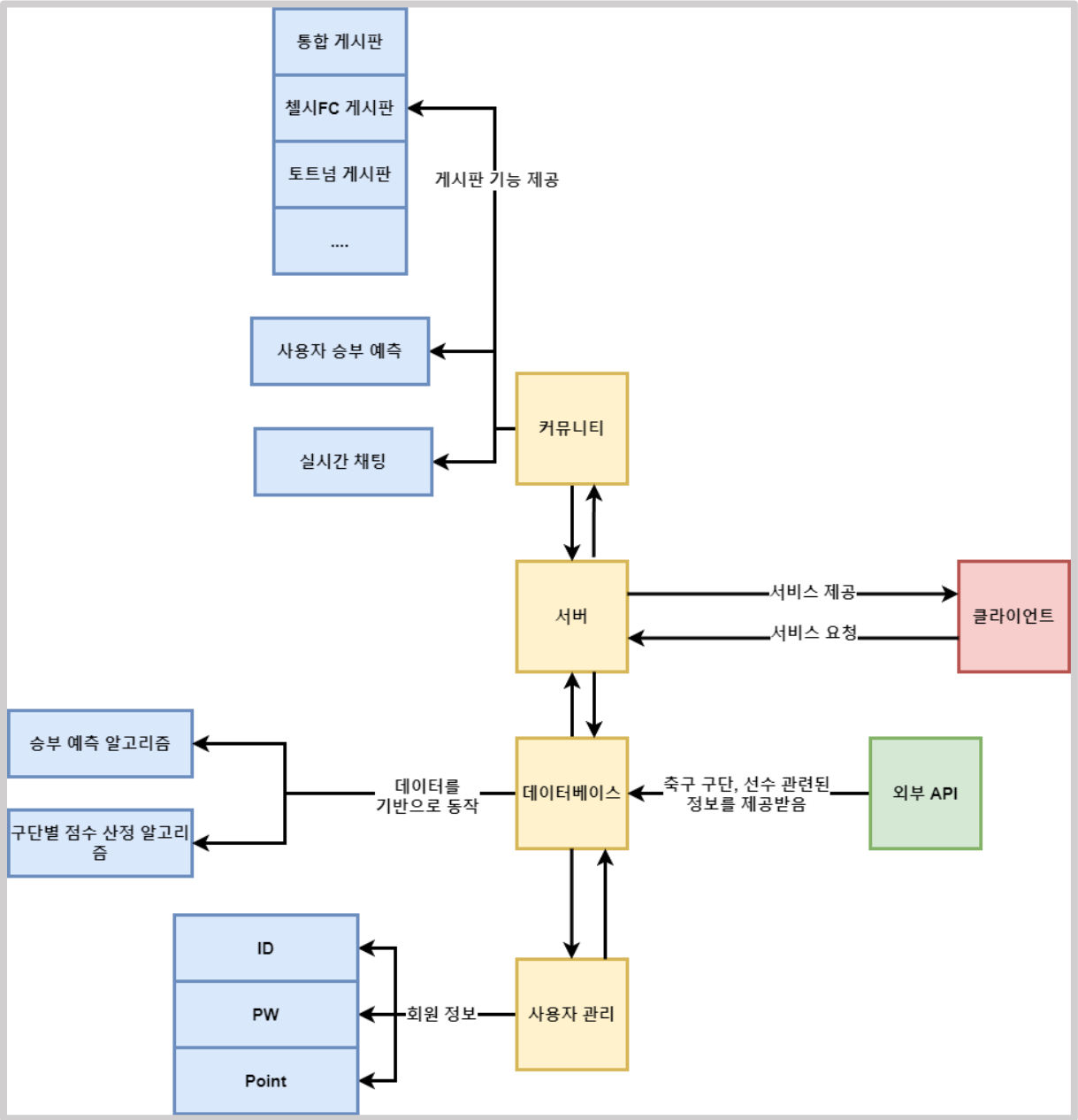
5.1.4 MySQL : MySQL 은 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)이다. MySQL 은 사용이 간편하며, 확장성이 좋고 안정적인 성능을 제공하여 다양한 종류의 애플리케이션에 널리 사용된다. MySQL 은 데이터를 테이블 형식으로 구성하며, SQL(Structured Query Language)을 사용하여 데이터의 생성, 조회, 수정 및 삭제 작업(CRUD)을 수행할 수 있다.

5.1.5 Sequelize : 시퀄라이즈(Sequelize)는 Node.js 기반의 ORM(Object-Relational Mapping) 라이브러리다. ORM 은 객체와 데이터베이스 간의 매핑을 담당하여 개발자가 SQL 쿼리를 직접 작성하지 않고도 데이터베이스를 조작하도록 한다. 시퀄라이즈는 JavaScript 객체와 데이터베이스 테이블 간의 매핑을 제공하고, 데이터베이스 쿼리와 조작을 위한 다양한 기능을 제공한다. 시퀄라이즈는 MySQL 을 비롯한 다양한 데이터베이스 시스템과 함께 작동할 수 있으며, 데이터베이스 스키마를 자동으로 생성하고 유지할 수 있는 마이그레이션 기능도 제공한다. 또한, 시퀄라이즈는 쿼리 작성을 위한 편리한 API, 데이터 유효성

검사, 관계 정의, 트랜잭션 관리 등의 기능을 제공하여 개발자가 데이터베이스와 상호 작용하는 작업을 간편하게 처리할 수 있도록 도와준다.

5.1.6 Pug : Pug 는 Node.js 를 위한 템플릿 엔진(Template Engine)으로, 템플릿 엔진은 동적인 웹 페이지를 생성하기 위해 사용되며, 데이터와 템플릿을 결합하여 HTML 을 동적으로 생성하는 기능을 제공한다.

5.2.1 소프트웨어 기능블록도



5.2.2 기능적 요구사항 명세서

ID	요구사항	내용	설명	우선순위
No_01	**웹 서버 구축**	Node.js 와 웹 서버 프레임워크(Express)를 이용하여 웹 서버 구축	트래픽을 예측하여 안정적인 웹 서버를 구축한다.	1
No_02	**커뮤니티 기능**	로그인, 통합 게시판, 게시물 작성, 댓글 작성, 포인트 획득, 구단 별 게시판 구현	기본적으로 커뮤니티가 갖춰야 할 기능들을 구현하며, 사이트 내에서만 사용 가능한 포인트로 여러 서비스를 제공한다. 게시판에는 통합 게시판 및 구단 별 게시판 등 여러 게시판을 사용자에게 제공한다.	2
No_03	**API 이용**	축구 데이터 관련 서버 및 사이트(fotmob)로부터 API 를 제공받음	축구 관련 데이터를 제공받아 구단 별 점수 계산 및 승부예측 알고리즘에 이용하며, 사용자에게 원하는 축구 데이터를 제공할 수 있도록 설계한다.	5
No_04	**구단 별 점수(Rating) 계산 알고리즘**	각 구단의 절대적인 점수를 부여하는 알고리즘의 설계	최근 경기 흐름, 현재 시즌의 성적 등 여러 파라미터를 고려하여 알고리즘을 설계한다.	3
No_05	**상대적 승률 계산 알고리즘**	사용자가 두 구단의 승부예측을 요청할 경우 자체적인 상대적 승률 알고리즘으로 예측 데이터 제공	최근 상대전적, 각 팀의 최근 흐름 등 여러 파라미터를 고려하여 알고리즘 설계 후 사용자에게 예측 결과를 제공한다.	4
No_06	**사용자 승부예측 서비스**	사용자들이 실제 경기에 대해서 사이트 내 포인트로 경기 결과를 예측하는 서비스 제공	사용자들이 실제 경기에 대해 승부예측을 진행하여 축구 경기에 대한 몰입도를 증가시키고, 예측 결과에 따라 포인트를 분배한다.	6

6. 제작(Implementation)

6.1 제작과정

웹 서버의 기본 틀로 express 프레임워크를 선택하였으며, 전공서의 6~9 장 내용을 참고하여 커뮤니티 서버의 기본 틀을 프로그래밍 하였다.

6.1.1 npm 을 통한 Express 서버 구축 : package.json

```
{
  "name": "pitchmatch", //패키지명
  "version": "0.0.1",   //SemVer 개발중인 버전
  "description": "",
  "main": "app.js",     //메인 app.js
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node app.js"
  },
  "author": "team_css", //제작자
  "license": "ISC",     //프로젝트 라이센스
  "dependencies": {
    "axios": "^1.4.0",  //http 크롤링 패키지
    "bcrypt": "^5.1.0", //해시 함수 기반 암호화 알고리즘. mysql 에 비밀번호 저장시 사용
    "cheerio": "^1.0.0-rc.12",  //html 파싱 라이브러리
    "connect-flash": "^0.1.1",  //일회성 메세지(flash) 표시 패키지
    "cookie-parser": "^1.4.6",  //쿠키를 파싱하고 관리하는 미들웨어
    "dotenv": "^16.0.3",        //.env 환경변수를 관리하는 라이브러리
    "express": "^4.18.2",       //express 웹서버 프레임워크
    "express-session": "^1.17.3", //express 서버 세션 관리용 미들웨어
    "morgan": "^1.10.0",        //http 요청에 관한 로깅을 처리하는 미들웨어
    "multer": "^1.4.5-lts.1",   //이미지 파일 업로드 모듈
    "mysql2": "^3.3.3",         //mysql 데이터베이스에 연결하는 라이브러리
    "passport": "^0.6.0",       //로그인 구현을 위한 사용자 인증 미들웨어
    "passport-kakao": "^1.0.1", //카카오 로그인
    "passport-local": "^1.0.0", //로컬 로그인
    "pug": "^3.0.2",            //템플릿 엔진은 pug 사용. 동적인 HTML 생성
    "sequelize": "^6.31.1",     //ORM 라이브러리, 데이터베이스 조작
    "serve-favicon": "^2.5.0"   //서버 favicon 제공
  },
  "devDependencies": {
    "nodemon": "^2.0.22"        //개발용 패키지, 코드 수정 시 서버에 바로 적용
  }
}
```

6.1.2 app.js

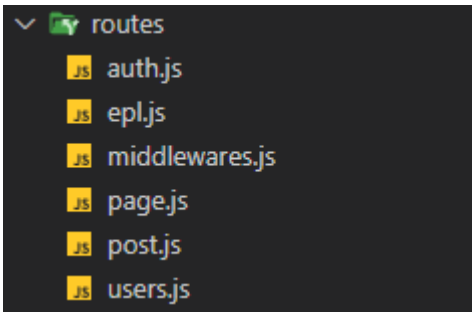
전체적인 틀은 교재 9 장에서 제시하는 방법을 사용하여 프로그래밍하였으며, 사용자의 요청에 대한 처리를 하기 위해 라우터 파일을 만들고, 다음과 같이 app.js 에 선언함.

```
//라우터 선언 및 경로 연결
const pageRouter = require('./routes/page');
const authRouter = require('./routes/auth');
const postRouter = require('./routes/post');
const userRouter = require('./routes/users');
const eplRouter = require('./routes/epl');

const passportConfig = require('./passport');
const { sequelize } = require('./models');    //시퀄라이즈 연결을 위해
```

다음과 같은 요청이 들어왔을 시 적절하게 서비스를 제공할 수 있도록 라우팅 미들웨어를 프로그래밍함

```
app.use('/', pageRouter);       //기본 경로
app.use('/auth', authRouter);   //로그인 처리
app.use('/post', postRouter);   //게시글 작성
app.use('/user', userRouter);   //사용자 정보
app.use('/epl', eplRouter);     //축구 데이터
//에러처리부분
app.use((req, res, next) => {
  const err = new Error('Not Found');
  err.status = 404;
  next(err);
});


//개발용
app.use((err, req, res) => {
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') ===='development' ? err : {};
  res.status(err.status || 500);
  res.render('error');
});

//서버 listening…
app.listen(app.get('port'), () => {
  console.log(app.get('port'),'번 포트에서 대기 중');
});
```

6.1.3 Router



라우터는 다음과 같이 프로그래밍하였다. 라우터는 클라이언트 요청에 대한 경로를 정의하고, 해당 경로에 대한 처리를 담당하는 역할을 수행한다. 라우터를 사용하여 요청을 적절한 Handler 함수로 라우팅해 해당 요청에 대한 응답을 처리할 수 있다.

6.1.2.1 page.js : 가장 기본적인 라우터. app.js 에서 '/'요청에 대한 응답에 대한 라우터로 정의하였다. 내부에서는 '/', '/join', '/profile' 3 가지 페이지가 프로그래밍 되어 있다. 로그인 여부에 따라 다르게 렌더링하도록 프로그래밍하였으며, 로그인을 하지 않았을 시 회원가입 페이지를 응답하게 하였다.

6.1.2.2 auth.js : 로그인 로그아웃 관련된 라우터. app.js 에서 '/auth' 요청 시 라우팅된다. '/join' 요청으로 회원가입, 'login'요청으로 로그인, '/logout' 요청으로 로그아웃에 대한 처리를 한다.

6.1.2.3 post.js : 게시글 관련 처리 라우터. app.js 에서 '/post' 요청 시 라우팅된다.'/' 요청은 게시글의 작성 및 경험치 획득, '/img' 요청은 multer 모듈을 통한 이미지 파일 업로드, '/hashtag' 요청은 해시태그 검색, '/category' 요청은 카테고리별 검색에 대한 처리를 한다.

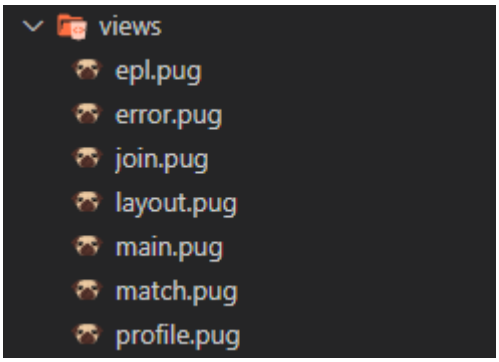
6.1.2.4 user.js : 다른 사용자를 팔로우할 수 있는 라우터이다. 먼저 팔로우할 사용자를 데이터베이스에서 조회한 후, 시퀄라이즈에서 추가한 addFollowing 메서드로 현재 로그인한 사용자와의 관계를 지정한다.

6.1.2.5 epl.js : 축구 데이터 관련 처리를 하는 라우터로, 교재의 코드를 참고하여 직접 프로그래밍 하였다. 간단히 구현한 Rating 계산 알고리즘을 통해 사용자가 '/epl/' 경로를 요청하였을 경우 실시간으로 업데이트를 한다. 또한, 매치정보에 대한 요청인 '/epl/match' 요청을 받았을 경우에 대한 라우팅도 구현하였다.

```
// EPL 순위 페이지 라우트
router.get('/', async (req, res) => {
  try {
    const eplTeams = await db.Epl.findAll();
    const transfers = await db.transfer.findAll(); // transfer 데이터 가져오기

    // curr_mmr 계산하여 데이터베이스 업데이트
    await Promise.all(
      eplTeams.map(async (team) => {
        const { tier, prev_win, prev_lose, curr_win, curr_lose } = team;
        const calculatedMmr = mmr(tier, prev_win, prev_lose, curr_win, curr_lose);
        await db.Epl.update({ curr_mmr: calculatedMmr }, { where: { num: team.num } });
      })
    );
    //epl.pug 렌더링
```

```
    res.render('epl', { eplTeams: eplTeams || [], transfer: transfers }); // transfer 데이터도 함께
전달
  } catch (error) {
    console.error(error);
    res.status(500).send('Internal Server Error');
  }
});

// 매치 정보 페이지 라우트
router.get('/match', async (req, res) => {
  try {
    const matches = await db.match.findAll();

    //매치데이터를 데이터베이스로부터 가져옴
    for (const match of matches) {
      const eplTeam = await db.Epl.findOne({
        where: { name: match.home },
      });


      //데이터베이스로부터 홈 팀의 레이팅 정보를 가져옴
      if (eplTeam) {
        match.home_mmr = eplTeam.curr_mmr;
        await match.save();
      }
      const eplAwayTeam = await db.Epl.findOne({
        where: { name: match.away },
      });
      //데이터베이스로부터 어웨이 팀의 레이팅 정보를 가져옴
      if (eplAwayTeam) {
        match.away_mmr = eplAwayTeam.curr_mmr;
        await match.save();
      }
    }
    //매치 데이터를 넘겨서 렌더링
    res.render('match', { matches: matches });

  } catch (error) {
    console.error(error);
    res.status(500).send('Server error');
  }
});

module.exports = router;
```

6.1.4 Pug



템플릿 엔진으로 pug 를 사용했으며, 라우팅 미들웨어가 사용자로부터 요청을 받아 처리할 때 응답으로 pug 를 렌더링한다. 기본적인 구조는 교재의 내용과 비슷하며, 본 프로젝트에서 추가한 내용을 구현하기 위하여 **layout.pug, main.pug 에 코드를 추가하였으며, epl.pug 와 match.pug 를 새로 프로그래밍하였다.**

6.1.4.1 layout.pug 수정사항

```
views > layout.pug
1  doctype html
2  html
3    head
4      <link rel="preconnect" href="https://fonts.googleapis.com">
5      <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
6      <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100;70
7      meta(charset='UTF-8')
```

1) 폰트를 바꾸기 위해 구글 폰트에서 가독성이 좋은 폰트를 가져옴

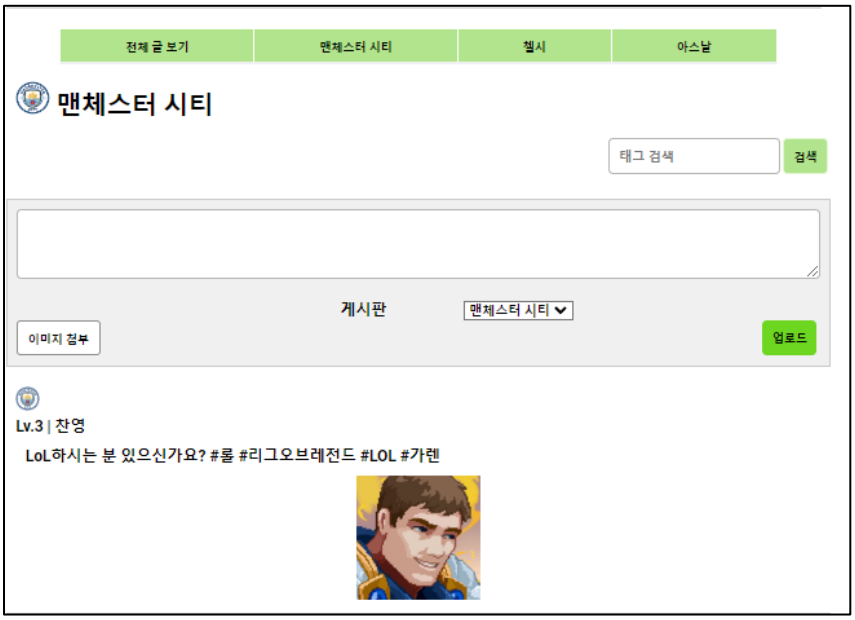
```
        body
          a#match.menu_btn(href='/epl/match') 경기 데이터
          a#epl.menu_btn(href='/epl') EPL 데이터
          a(href="/")
            img.main-img(src="/main.png")
        .mainframe
          .container
23          .user-level
24            | level: #{user.level}
25          .user-points
26            | 경험치: #{user.points} / #{user.level*user.level}
27          .half
28            div 팔로잉
29            .count.following-count= user.Followings && user.Followings.length || 0
30          .half
```

2)최상단에 서버의 배너를 출력하도록 15~16 줄에 코드를 추가. 배너를 클릭하면 서버의 기본 주소로 라우팅

3)레벨 및 경험치를 표시하기 위해 23~26 줄에 코드를 추가. 데이터베이스로부터 user.level 및 user.points 데이터를 불러와서 사용함.

6.1.4.2 main.pug 수정사항

(수정된 layout.pug 렌더링 결과)




```
block content
  .timeline
    .twits
      form#category-form(action='/post/category' method='get')
        a.category_btn(href='/') 전체 글 보기
        button.category_btn(type='submit' name='category' value='category1') 맨체스터 시티
        button.category_btn(type='submit' name='category' value='category2') 첼시
        button.category_btn(type='submit' name='category' value='category3') 아스날
```

1) 카테고리별로 게시글을 확인할 수 있도록 카테고리 탭 프로그래밍

```
.title
  h1
    - if (category === 'category1')
      img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/z44l-a0W1v5FmgPner
      |    맨체스터 시티
    - else if (category === 'category2')
      img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/fhBITrIlbQxhVB6Ijx
      |    첼시
    - else if (category === 'category3')
      img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/4us2nCgl6kgZc0t3hp
      |    아스날
    - else
      | PitchMatch 전체 글 보기
```

2) 카테고리를 골랐을 경우 어떤 카테고리인지 화면에 출력. 카테고리별로 이미지도 출력함

```
if user
  div
    form#twit-form(action='/post' method='post' enctype='multipart/form-data')
      .input-group
        textarea#twit(name='content' maxlength=140)
        .input-group
          label(for='category', style='color: black;') 게시판
          select#category(name='category')
            option(value='category1') 맨체스터 시티
            option(value='category2') 첼시
            option(value='category3') 아스날
        .img-preview
          img#img-preview(src='' style='display: none;' width='250' alt='미리보기')
        input#img-url(type='hidden' name='url')
      div
        label#img-label(for='img') 이미지 첨부
        input#img(type='file' accept='image/*')
      button#twit-btn.btn(type='submit') 업로드
```

3) 게시글 작성 시 어떤 카테고리에 작성할 것인지 선택하도록 프로그래밍.

```
for twit in twits
  .twit
    input.twit-user-id(type='hidden' value=twit.user.id)
    input.twit-id(type='hidden' value=twit.id)
    .category
      - if (twit.category === 'category1')
        img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/z44l-a0w
      - else if (twit.category === 'category2')
        img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/fhBITrIl
      - else if (twit.category === 'category3')
        img.team_img(src="https://ssl.gstatic.com/onebox/media/sports/logos/4us2nCgl
    .twit-level
      | Lv.#{twit.user.level} |
      .twit-author= twit.user.nick
    - const follow = user && user.Followings.map(f => f.id).includes(twit.user.id);
    if user && user.id !== twit.user.id && !follow
      button.twit-follow Follow
    .twit-content= twit.content
    if twit.img
      .twit-img
        img(src=twit.img alt='섬네일')
```

4) 데이터베이스에 존재하는 게시글을 출력할 시, 사용자의 레벨을 출력하고, 게시글이 어떤 카테고리에 작성되었는지에 따라 구단의 로고를 출력하도록 프로그래밍.

6.1.4.3 epl.pug

정기 데이터 | EPL 데이터 | PitchMatch Team CSS

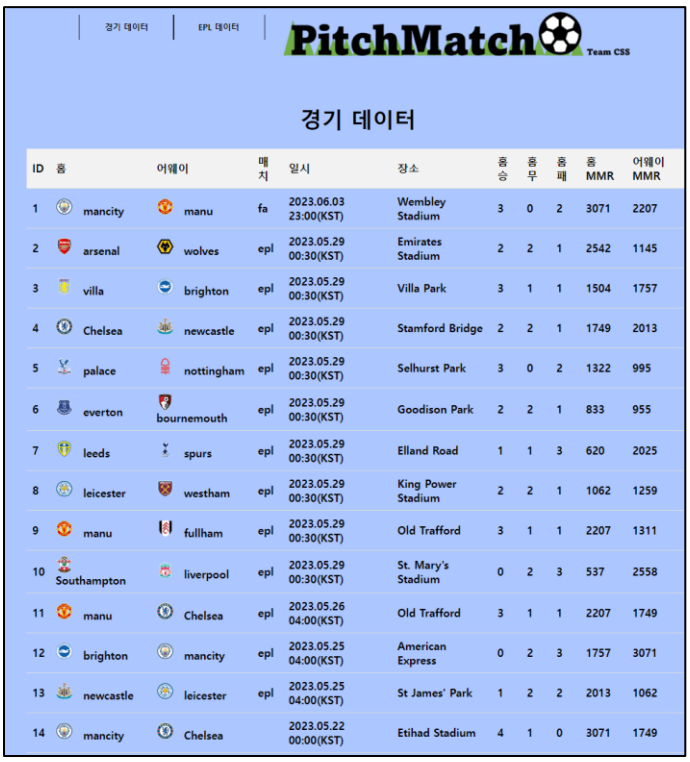
EPL 데이터

순위	팀	티어	prev 승	prev 무	perv 패	승	무	패	승점	MMR
1	mancity	1	29	6	3	28	5	5	89	3071
2	arsenal	2	22	3	13	26	6	6	84	2542
3	manu	2	16	10	12	23	6	9	75	2207
4	newcastle	3	13	10	15	19	14	5	71	2013
5	liverpool	1	28	8	2	19	10	9	67	2558
6	brighton	3	12	15	11	18	8	12	62	1757
7	villa	4	13	6	19	18	7	13	61	1504
8	spurs	1	22	5	11	18	6	14	60	2025
9	brentford	4	13	7	18	15	14	9	59	1563
10	fullham	5	0	0	0	15	7	16	52	1311
11	palace	3	11	15	12	11	12	15	45	1322
12	Chelsea	1	21	11	6	11	11	16	44	1749
13	wolves	3	15	6	17	11	8	19	41	1145
14	westham	2	16	8	14	11	7	20	40	1259
15	bournemouth	5	0	0	0	11	6	21	39	955
16	nottingham	5	0	0	0	9	11	18	38	995
17	everton	4	11	6	21	8	12	18	36	833
18	leicester	2	14	10	14	9	7	22	34	1062
19	leeds	5	9	11	18	7	10	21	31	620
20	Southampton	4	9	13	16	6	7	25	25	537

id	선수명	소속팀	이전 팀	이적료
1	J.Cancelo	munchen	mancity	loan
2	Maximo Perrone	mancity	Velez Sarsfield	undisclosed
3	Leandro Trossard	arsenal	Brighton	27m
4	Jakub Kiwior	arsenal	Spezia	20m
5	Jorginho	arsenal	Chelsea	12m
6	Albert Sambi Lokonga	Crystal Palace	arsenal	loan
7	Benoit Badiashile	Chelsea	Monaco	32m
8	Joao Felix	Chelsea	Atletico Madrid	loan
9	Enzo Fernandez	Chelsea	Benfica	107m
10	Mykhaylo Mudryk	Chelsea	Shakhtar Donetsk	89m
11	Cesare Casadei	Reading	Chelsea	loan
12	Malo Gusto	Lyon	Chelsea	loan
13	Jorginho	arsenal	Chelsea	12m
14	Cody Gakpo	liverpool	PSV	37m
15	Jarell Quansah	Bristol Rovers	liverpool	loan
16	Jake Cain	Swindon	liverpool	undisclosed
17	Marcel Sabitzer	manu	munchen	loan
18	Jack Butland	manu	palace	loan
19	Wout Weghorst	manu	Burnley	loan
20	Cristiano Ronaldo	Al-Nassr	manu	FA
21	Arnaut Danjuma	spurs	Villarreal	loan
22	Pedro Porro	spurs	Sporting	loan

1) epl.pug 에서는 데이터베이스의 pitchmatch.epl, pitchmatch.transfer 데이터로부터 각 팀의 정보, 이적 정보 등을 읽어와 table 로 출력한다. 조건문을 통해서 각 구단의 로고를 출력하도록 하였으며, 레이팅(mmr)에 대한 정보 또한 제공된다.

6.1.4.4 match.pug



경기 데이터 | EPL 데이터

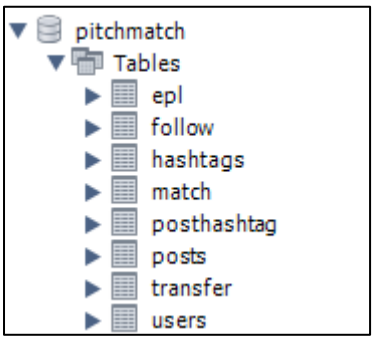
PitchMatch Team CSS

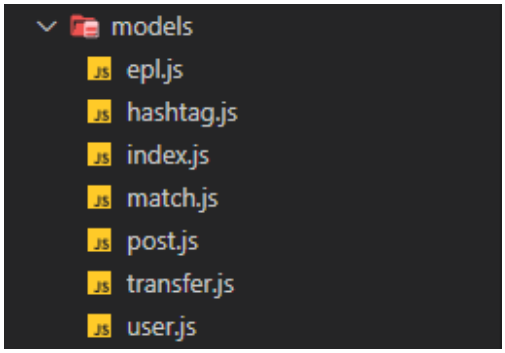
경기 데이터

ID	홈	어웨이	매치	일시	장소	홈 승	홈 무	홈 패	홈 MMR	어웨이 MMR
1	mancity	manu	fa	2023.06.03 23:00(KST)	Wembley Stadium	3	0	2	3071	2207
2	arsenal	wolves	epl	2023.05.29 00:30(KST)	Emirates Stadium	2	2	1	2542	1145
3	villa	brighton	epl	2023.05.29 00:30(KST)	Villa Park	3	1	1	1504	1757
4	Chelsea	newcastle	epl	2023.05.29 00:30(KST)	Stamford Bridge	2	2	1	1749	2013
5	palace	nottingham	epl	2023.05.29 00:30(KST)	Selhurst Park	3	0	2	1322	995
6	everton	bournemouth	epl	2023.05.29 00:30(KST)	Goodison Park	2	2	1	833	955
7	leeds	spurs	epl	2023.05.29 00:30(KST)	Elland Road	1	1	3	620	2025
8	leicester	westham	epl	2023.05.29 00:30(KST)	King Power Stadium	2	2	1	1062	1259
9	manu	fullham	epl	2023.05.29 00:30(KST)	Old Trafford	3	1	1	2207	1311
10	Southampton	liverpool	epl	2023.05.29 00:30(KST)	St. Mary's Stadium	0	2	3	537	2558
11	manu	Chelsea	epl	2023.05.26 04:00(KST)	Old Trafford	3	1	1	2207	1749
12	brighton	mancity	epl	2023.05.25 04:00(KST)	American Express	0	2	3	1757	3071
13	newcastle	leicester	epl	2023.05.25 04:00(KST)	St James' Park	1	2	2	2013	1062
14	mancity	Chelsea		2023.05.22 00:00(KST)	Etihad Stadium	4	1	0	3071	1749

1) match.pug 는 데이터베이스의 pitchmatch.match 로부터 경기에 대한 정보를 받아와서 table 로 출력한다. 또한, pitchmatch.epl 데이터베이스 내부의 curr_mmr 부분으로부터 홈팀과 어웨이팀의 레이팅 또한 가져와서 사용자에게 제공하도록 프로그래밍하였다.

6.1.5 데이터베이스(MySQL, Sequelize)





본 프로젝트의 데이터베이스는 RDBMS 인 MySQL 을 사용했고, 이를 사용하기 위해 Sequelize 를 사용하여 관리하였다. 좌측 그림은 Sequelize 와 MySQL 을 연결하여 테이블을 생성한 결과 화면이다. 아래는 좌측과 같은 테이블을 만들기 위해 생성한 models 파일 목록으로, **epl.js 는 epl 팀들의 정보가 저장되고, match.js 는 실제 존재하는 경기 정보가 저장되며, transfer.js 에는 이적시장에 관련된 정보가 저장된다.**

6.1.6 레이팅 계산 알고리즘

```
// 초기 레이팅 설정 및 티어에 따른 초기 레이팅의 보정
/*
const default_rating = 1500;
const tier1_rating = default_rating + 300;
const tier2_rating = default_rating + 150;
const tier3_rating = default_rating;
const tier4_rating = default_rating - 150;
const tier5_rating = default_rating - 300;
*/

// rating 계산 함수: 승률 보정이 존재하지 않는 버전. mmr.js
var k = 39.473;

function mmr(tier, prev_win, prev_lose, curr_win, curr_lose){
    let prev_mmr = 1200 + (5-tier)*150 + prev_win*k - prev_lose*k;
    let soft_mmr = (prev_mmr - 1500) * 0.5 + 1500;
    let mmr = soft_mmr + curr_win*k - curr_lose*k;
    return mmr;
}

module.exports = mmr;
//const mmr = require('./mmr.js');
```

각 구단의 레이팅을 계산하는 알고리즘을 간단하게 구현하였다. 초기 mmr 은 1500 으로 설정하였고, 자체적으로 판단하여 20 개 구단에 대한 tier 를 나눈 다음, 초기 mmr 에 보정치를 추가하였다. 1 티어는 초기 mmr 이 1800, 2 티어는 1650 등 티어가 낮아질수록 150 점씩 낮아지는 구조이다. EPL 은 1 시즌 각 팀이 38 경기를 하기에 전패를 했을 경우 레이팅을 0 으로 설정하게 하기 위하여 k 를 39.473 으로 설정하였다.

<u>이번 시즌(22/23 시즌) 각 팀의 mmr 계산 과정은 다음과 같다.</u>

각 팀의 초기 MMR -> 21/22 시즌 결과를 바탕으로 조정 -> 소프트 리셋 -> 현 시즌 결과를 바탕으로 조정

작년 시즌(21/22 시즌)을 기준으로 prev_win 과 prev_lose, 그리고 각 팀별로의 tier 를 통해 전 시즌 종료 시점의 mmr 을 계산한 다음, 그것을 softreset 한다. 이렇게 하는 이유는, 시즌이 거듭되면서 mmr 이 너무 높아지거나 낮아지는 현상을 막기 위해 하였다. 그렇게 계산한 결과가 22/23 시즌의 각 팀 초기 mmr 로 설정된다. 그것을 또한 이번 시즌 경기 결과를 바탕으로 계산하면 그것이 현 시점 각 팀의 강함을 표시할 수 있는 mmr 이라고 정의하였음.

6.1.7 그 외 기능들

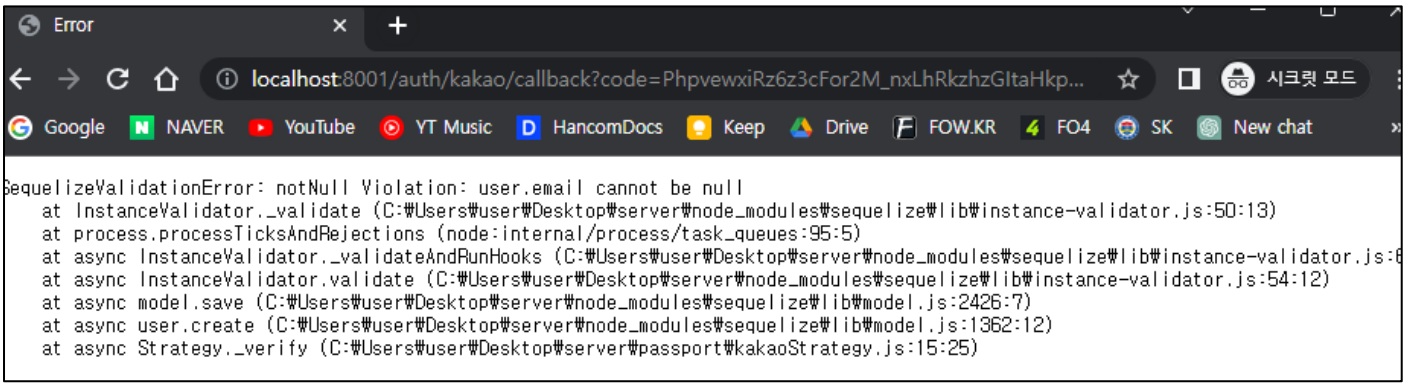
passport 모듈로 local 로그인 구현, server-favicon 모듈로 파비콘 제공

6.2 제작 시 문제점 및 개선사항

6.2.1 API 호출 불가 - 처음 목적은 축구 관련 API 를 호출하여 데이터를 가공하면서 정밀한 알고리즘을 만들 생각이었지만, API 를 호출하는 권한에 문제가 있어서 경기 데이터를 데이터베이스로 자동으로 불러오는 것에 문제가 발생하였다. 이에 따라 데이터베이스에 우선 정보를 수동으로 추가하였으며, 나중에 API 호출이 가능하다면 데이터를 어떻게 가공하여 사용자에게 제공할 것인지에 대한 방향을 설계하였다. 현 시점에서 API 를 통한 데이터 불러오기는 불가능하지만 확장성을 고려하여 프로그래밍을 진행하였다. 이 때문에 알고리즘 구현 부분을 간단하게 진행하였으며, 커뮤니티 서비스에 중점적으로 개발을 진행하였다.

6.2.2 passport-kakao 로그인 불가

(카카오톡 로그인 요청 시 출력되는 화면)

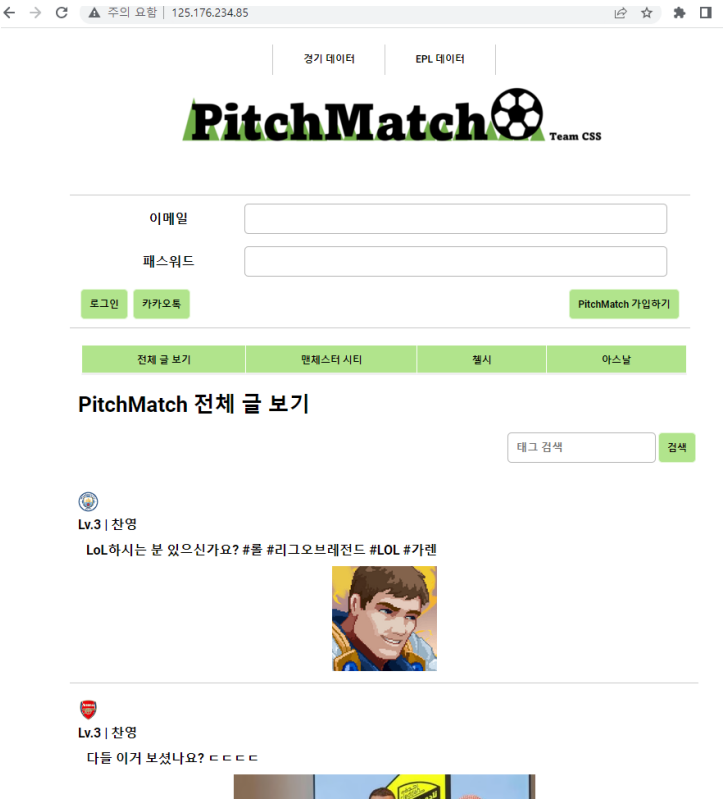


(로그인 결과 화면)

교재에 설명된 passport 모듈을 사용한 카카오 로그인을 구현하려고 하였으나, passport 모듈에 대한 이해 및 응용 능력 부족으로 에러가 발생하였음에도 해결하지 못하였음. **email 을 받아오는 과정에서 에러가 나는 것으로 추정되지만, 해결하지 못하였다.**

6.2.3 프론트엔드 CSS - 팀원 모두가 프론트 개발 능력이 부족하여 사이트의 가독성 및 디자인을 구현하는 것에 상당히 어려움을 겪었다. google 검색, chat gpt, google bard, ms bing 등을 통하여 최대한 가독성 좋게 구현할 수 있도록 팀원 모두가 노력하였고 만족할만한 결과물이 나왔다고 평가하였음.

7. 시험



서버 진입 화면

축구 관련 데이터시트 진입 버튼, 메인 복귀 배너, 로그인 폼, 게시판 탭, 게시판 명, 태그 검색, 게시글 출력됨.

PitchMatch 가입하기 클릭 시 회원가입 폼이 출력됨.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	email	nick	password	provider	snsId	points	level	createdAt	updatedAt
1	a@a	a	$2b$12$b.R...	local	NULL	4	3	2023-06-07 11:51:43	2023-06-07 14:!
2	i@i	i	$2b$12$kpI...	local	NULL	8	3	2023-06-07 11:56:38	2023-06-07 12:-
3	aa@aa	테스트용 a	$2b$12$7W...	local	NULL	1	2	2023-06-07 12:05:12	2023-06-07 12:(
4			$2b$12$bqz...	local	NULL	0	1	2023-06-07 14:52:17	2023-06-07 14:!
5	cksdud@cksdud	찬영	$2b$12$Dur...	local	NULL	4	3	2023-06-07 15:05:16	2023-06-07 15:(
6	w@w	지성원	$2b$12$GN...	local	NULL	1	2	2023-06-07 15:05:20	2023-06-07 15:(
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

가입완료 후 데이터베이스 users 테이블에 계정 정보가 생성된 모습

로그인 완료 후 프로필에 닉네임, 레벨, 경험치, 팔로잉, 팔로워 출력, 게시글 작성 폼 활성화됨.

내정보 버튼 클릭시 팔로잉 목록과 팔로워 목록이 출력됨.

전체 글 보기 탭을 클릭시 “PitchMatch 전체 글 보기” 출력, 게시글도 모든 글이 출력됨.

각 구단 게시판 탭 클릭시 그 구단의 로고와 구단 명 출력, 해당 구단 게시글만 출력됨.

게시글은 구단 로고, 유저의 레벨과 닉네임, 게시글의 내용이 출력됨.

다른 계정 작성자의 Follow 버튼을 누를 시 팔로잉이 1 증가하고, 팔로잉 목록에 추가됨.

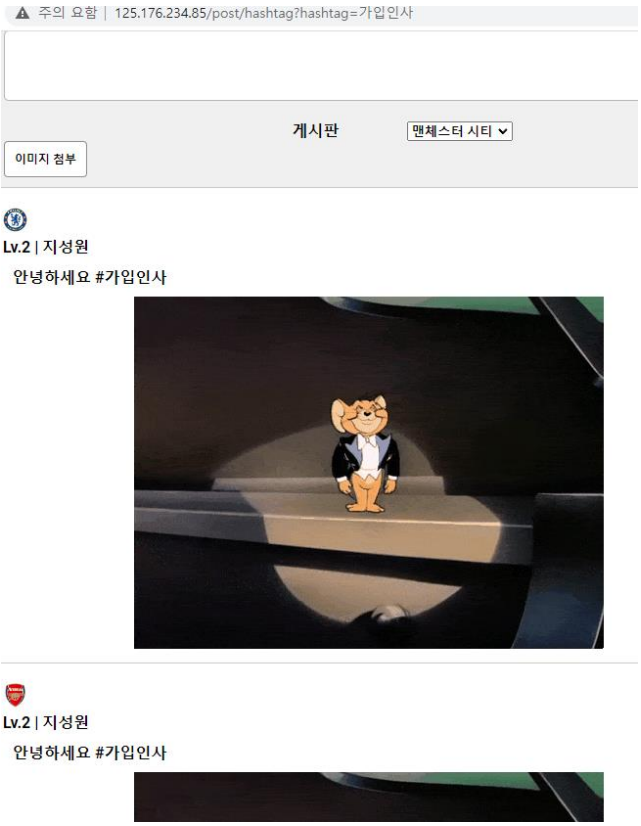
Follow 대상 계정은 팔로워가 1 증가하고, 팔로워 목록에 추가됨.

게시글 작성 시 경험치 +1 획득

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

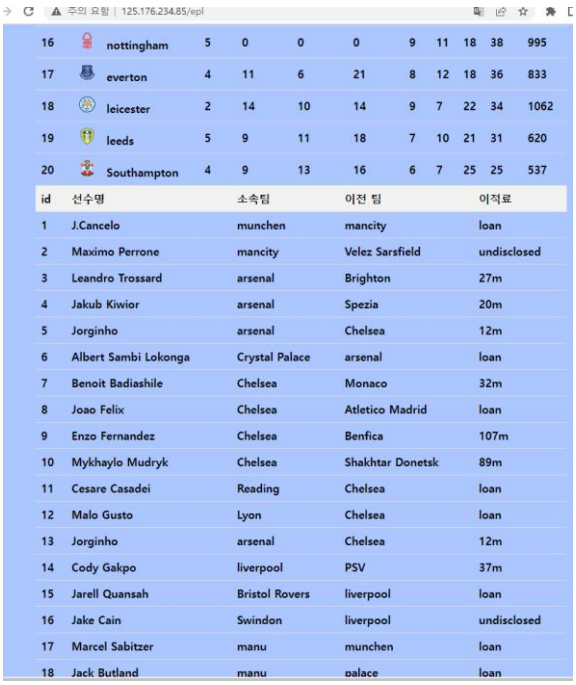
id	content	img	category	createdAt	updatedAt	dele
10	ㅎㅇㅎㅇ #1	/img/168605349...	category2	2023-06-07 1...	2023-06-07 14...	NULL
11	zz		category1	2023-06-07 1...	2023-06-07 14...	NULL
12	test		category1	2023-06-07 1...	2023-06-07 14...	NULL
13	테스트 #2		category1	2023-06-07 1...	2023-06-07 14...	NULL
14	안녕하세요! 처음 가입하고 인사 올립니다. 헬...		category2	2023-06-07 1...	2023-06-07 15...	NULL
15	눈에넣어도안아픈우리하베르츠....	/img/6a36f247d...	category2	2023-06-07 1...	2023-06-07 15...	NULL
16	안녕하세요 #가입인사	/img/img168615...	category3	2023-06-07 1...	2023-06-07 15...	NULL
17	다들 이거 보셨나요? ㄷㄷㄷㄷ	/img/ï¤í¬ë¦ª 20...	category3	2023-06-07 1...	2023-06-07 15...	NULL
18	LoL하시는 분 있으신가요? #롤 #리그오브레전...	/img/576616861...	category1	2023-06-07 1...	2023-06-07 15...	NULL
19	안녕하세요 #가입인사	/img/img168615...	category2	2023-06-07 1...	2023-06-07 16...	NULL
NULL		NULL	NULL	NULL	NULL	NULL

데이터베이스 posts 테이블에 게시글 데이터가 생성된 모습.



태그 검색으로 "가입인사" 검색, 해당 키워드가 있는 게시글만 출력됨.

순위	팀	티어	prev 승	prev 무	perv 패	승	무	패	승점	MMR
1	mancity	1	29	6	3	28	5	5	89	3071
2	arsenal	2	22	3	13	26	6	6	84	2542
3	manu	2	16	10	12	23	6	9	75	2207
4	newcastle	3	13	10	15	19	14	5	71	2013
5	liverpool	1	28	8	2	19	10	9	67	2558
6	brighton	3	12	15	11	18	8	12	62	1757
7	villa	4	13	6	19	18	7	13	61	1504
8	spurs	1	22	5	11	18	6	14	60	2025
9	brentford	4	13	7	18	15	14	9	59	1563
10	fullham	5	0	0	0	15	7	16	52	1311
11	palace	3	11	15	12	11	12	15	45	1322
12	Chelsea	1	21	11	6	11	11	16	44	1749
13	wolves	3	15	6	17	11	8	19	41	1145
14	westham	2	16	8	14	11	7	20	40	1259



16	nottingham	5	0	0	0	9	11	18	38	995
17	everton	4	11	6	21	8	12	18	36	833
18	leicester	2	14	10	14	9	7	22	34	1062
19	leeds	5	9	11	18	7	10	21	31	620
20	Southampton	4	9	13	16	6	7	25	25	537

id	선수명	소속팀	이전 팀	이적료
1	J.Cancelo	munchen	mancity	loan
2	Maximo Perrone	mancity	Velez Sarsfield	undisclosed
3	Leandro Trossard	arsenal	Brighton	27m
4	Jakub Kiwior	arsenal	Spezia	20m
5	Jorginho	arsenal	Chelsea	12m
6	Albert Sambi Lokonga	Crystal Palace	arsenal	loan
7	Benoit Badiashile	Chelsea	Monaco	32m
8	Joao Felix	Chelsea	Atletico Madrid	loan
9	Enzo Fernandez	Chelsea	Benfica	107m
10	Mykhaylo Mudryk	Chelsea	Shakhtar Donetsk	89m
11	Cesare Casadei	Reading	Chelsea	loan
12	Malo Gusto	Lyon	Chelsea	loan
13	Jorginho	arsenal	Chelsea	12m
14	Cody Gakpo	liverpool	PSV	37m
15	Jarell Quansah	Bristol Rovers	liverpool	loan
16	Jake Cain	Swindon	liverpool	undisclosed
17	Marcel Sabitzer	manu	munchen	loan
18	Jack Butland	manu	palace	loan

EPL 데이터 버튼 클릭 시 각 구단의 순위, 승점, 알고리즘으로 계산된 mmr 출력, 최근 이적시장에 대한 정보 출력됨.

ID	홈	어웨이	매치	일시	장소	홈 승	홈 무	홈 패	홈 MMR	어웨이 MMR
1	mancity	manu	fa	2023.06.03 23:00(KST)	Wembley Stadium	3	0	2	3071	2207
2	arsenal	wolves	epl	2023.05.29 00:30(KST)	Emirates Stadium	2	2	1	2542	1145
3	villa	brighton	epl	2023.05.29 00:30(KST)	Villa Park	3	1	1	1504	1757
4	Chelsea	newcastle	epl	2023.05.29 00:30(KST)	Stamford Bridge	2	2	1	1749	2013
5	palace	nottingham	epl	2023.05.29 00:30(KST)	Selhurst Park	3	0	2	1322	995
6	everton	bournemouth	epl	2023.05.29 00:30(KST)	Goodison Park	2	2	1	833	955
7	leeds	spurs	epl	2023.05.29 00:30(KST)	Elland Road	1	1	3	620	2025
8	leicester	westham	epl	2023.05.29 00:30(KST)	King Power Stadium	2	2	1	1062	1259
9	manu	fullham	epl	2023.05.29 00:30(KST)	Old Trafford	3	1	1	2207	1311
10	Southampton	liverpool	epl	2023.05.29 00:30(KST)	St. Mary's Stadium	0	2	3	537	2558

경기 데이터 버튼 클릭 시 각 구단들에 대한 경기일정, 두 구단에 대한 그간의 홈, 어웨이, 상대전적, 홈, 어웨이에 대한 mmr 이 출력

8. 평가

8.1 정량적/정성적 목표달성도 평가

항목	목표	달성률	비고
1	Node.JS 와 웹 서버 프레임워크를 이용한 커뮤니티 사이트를 구현한다. 커뮤니티의 기능으로는 로그인 기능, 통합 게시판, 게시물 작성, 댓글 작성, 사이트 내에서만 이용이 가능한 포인트 획득 등 기본적인 커뮤니티가 제공하는 기능들을 포함한다.	100%	달성 완료
2	축구 구단 별 게시판을 만들어 각 팀의 팬들끼리 소통할 수 있는 게시판을 구현한다.	100%	달성 완료
3	축구 정보를 제공해주는 서버(fotmob.com)로부터 API 를 제공받아 축구 구단 및 선수에 대한 데이터를 사용한다.	10%	외부 API 호출 권한 거절로 인한 실패. 수기로 mysql 에 epl 데이터를 추가하였다.
4	각 축구 구단마다 점수(Rating)를 부여하는 알고리즘을 작성한다. 이 알고리즘을 통해 부여되는 점수는 단순히 구단의 리그 내 순위가 아닌, 실질적인 그 구단의 파워를 나타내는 것을 목표로 한다. 점수에 영향을 끼치는 파라미터에는 최근 경기의 성적 흐름, 현재 시즌의 성적 등으로 설정한다.	50%	API 호출 실패로 간략하게 구현하였음. 파라미터에는 전 시즌, 현 시즌 승 패 정보만 포함됨
5	실제 진행될 매치 데이터를 바탕으로 승부예측에 도움이 될 만한 기능을 제공한다. 각 팀의 Rating 을 표시해주고, 양 팀의 상대전적을 5 경기 데이터를 제공한다.	100%	기능은 구현하였으나 API 로 데이터를 불러오는 것이 실패하여 수기로 경기 정보를 작성하여 어떤 식으로 표현하고 싶은지 프로그래밍.
6	각 구단의 이적시장 정보를 제공하는 기능을 구현한다. 제공되는 정보에는 선수명, 어느 팀에서 어느 팀으로 이적하였는지, 이적료(이적 형태) 등이 있다.	50%	기능은 구현하였으나 API 로 데이터를 불러오는 것이 실패하여 수기로 경기 정보를 작성하여 어떤 식으로 표현하고 싶은지 프로그래밍.

8.2 현실적 제한요소 달성도 평가

현실적 제한 요소	내용	달성 결과
경제	(1) 서버 트래픽 과다 시를 고려한 안정적인 서버 장비 확보 (2) 낭비되는 리소스 없이 최적의 리소스만을 사용하여 웹 서버를 구현해 경제성 고려	(1) 데이터베이스를 활용하여 달성 완료
편리	(1) 사용자가 서버에 접속했을 시 최대한 사용자의 가독성을 고려하여 개발 (2) UI 디자인 및 사이트 내 요소들의 가독성 및 직관성	(1) pug, css 를 통하여 달성 완료
윤리	(1) 사이트 내 승부예측 기능의 경우 사행성 요소로 번질 가능성이 있음 (2) 관련 기준에 따른 윤리성 고려	(1) 승부예측 기능은 사용자에게 달렸다고 평가
사회	(1) 축구 관련 소식이 아닌 정보는 제공하지 않음 (2) 다른 사용자들에게 피해나 불쾌감을 줄 수 있는 데이터를 방지	(1) 데이터베이스를 관리자가 관리하여 부적절한 게시글 삭제

8.3 기능적 요구사항 달성도 평가

ID	요구사항	내용	설명	달성률
No_01	**웹 서버 구축**	Node.js 와 웹 서버 프레임워크(Express)를 이용하여 웹 서버 구축	트래픽을 예측하여 안정적인 웹 서버를 구축한다.	**100%**
No_02	**커뮤니티 기능**	로그인, 통합 게시판, 게시물 작성, 댓글 작성, 포인트 획득, 구단 별 게시판 구현	기본적으로 커뮤니티가 갖춰야 할 기능들을 구현하며, 사이트 내에서만 사용 가능한 포인트로 여러 서비스를 제공한다. 게시판에는 통합 게시판 및 구단 별 게시판 등 여러 게시판을 사용자에게 제공한다.	**100%**
No_03	**API 이용**	축구 데이터 관련 서버 및 사이트(fotmob)로부터 API 를 제공받음	축구 관련 데이터를 제공받아 구단 별 점수 계산 및 승부예측 알고리즘에 이용하며, 사용자에게 원하는 축구 데이터를 제공할 수 있도록 설계한다.	**0%**
No_04	**구단 별 점수(Rating) 계산 알고리즘**	각 구단의 절대적인 점수를 부여하는 알고리즘의 설계	최근 경기 흐름, 현재 시즌의 성적 등 여러 파라미터를 고려하여 알고리즘을 설계한다.	**50%**

No_05	**상대적 승률 계산 알고리즘**	사용자가 두 구단의 승부예측을 요청할 경우 자체적인 상대적 승률 알고리즘으로 예측 데이터 제공	최근 상대전적, 각 팀의 최근 흐름 등 여러 파라미터를 고려하여 알고리즘 설계 후 사용자에게 예측 결과를 제공한다.	25%
No_06	**사용자 승부예측 서비스**	사용자들이 실제 경기에 대해서 사이트 내 포인트로 경기 결과를 예측하는 서비스 제공	사용자들이 실제 경기에 대해 승부예측을 진행하여 축구 경기에 대한 몰입도를 증가시키고, 예측 결과에 따라 포인트를 분배한다.	0%

9. 추진체계

본 과제의 추진을 위한 팀은 3 인으로 구성되며 다음과 같은 역할을 담당한다. 이론 연구 단계에서부터 과제 종료시까지 협력과 보완을 통해 팀의 목표를 달성할 수 있도록 한다.

팀장	**김찬영**	프로젝트 총괄, 백엔드 개발, 보고서 작성
팀원	**지성원**	프로그래밍, 백엔드 및 프론트 개발, 오류 수정, 디버깅
팀원	**우상욱**	프론트 프로그래밍, 자료 조사, 회의록 작성, ppt 제작

10. 설계 추진 일정 : *2023 년 4 월 13 일 ~ 2023 년 6 월 18 일*

수행 내용		일정							
		1	2	3	4	5	6	7	8
목표와 기준 설정	**-설계목표 설정 -목표 달성 방법에 대한 논의**								
합성	**-기능별 구현 방법 결정 -적용할 기술**								
분석	**-세부적인 기능 구현 방법 결정 -목표달성 가능성 확인**								
제작	**-프론트엔드, 백엔드 프로그래밍**								
시험/평가	**-시험 및 검증 -재설계**								

결과	-결과보고 및 시연	

11. 결론

이번 학기에 Node.js 와 Express 를 활용하여 축구 커뮤니티 서버를 개발한 경험은 정말 놀라운 여정이었다. 처음부터 아무것도 모르는 상태에서 시작했지만, 막막한 상황에서도 우리 팀은 서로를 믿고 협력하여 이 프로젝트를 성공적으로 완료했다고 생각한다. 이를 통해 많은 것을 배우고 느낄 수 있었다.

처음에는 Node.js 와 Express 의 개념조차 이해하기 어려웠습니다. 그러나 인내심을 가지고 여러 정보를 찾아보고 예제를 따라해보며 서서히 이해하고 활용하는 방법을 익혔다. 이러한 학습 과정은 기술적인 면에서만이 아니라, 문제 해결 및 협업 능력을 향상시키는 데에도 큰 도움이 되었다. 그것에 더해 프로젝트를 진행하면서 주기적인 회의와 소통을 통해 문제를 해결하고 아이디어를 공유했다. 서로의 역할과 책임을 분담하여 효율적으로 작업할 수 있었고, 이를 통해 더 나은 결과물을 만들어낼 수 있었다고 생각한다. 또한, 완전히 새로운 기술을 배우고 적용하면서 발생하는 어려움과 고민을 해결해 나갔다. 처음부터 우리가 아는 것이 없는 상황에서는 많은 도전과 시행착오가 있었지만, 결국에는 우리의 노력과 열정으로 이겨냈다고 평가한다.

비록, API 호출이나 passport kakao 로그인 모듈에서 목표 달성률이 낮은 부분이 존재하였지만, 이를 해결할 수 없다고 판단하고 다른 커뮤니티 기능들 및 데이터베이스 부분을 보완하자고 팀원 모두가 협의하여 결과적으로 좋은 프로젝트가 되었다고 생각한다.

이번 프로젝트를 통해 우리는 개발 과정에서의 문제 해결 능력과 협업 능력을 향상시킬 수 있었을 뿐만 아니라, Node.js 와 Express 를 활용하여 웹 서버를 구축하고 관리하며, MySQL 을 통해 데이터를 관리하는 기술을 배웠다. 이러한 경험과 지식은 저희에게 큰 자신감과 동기부여가 되었고, 미래의 개발자로서 더 큰 프로젝트에 도전할 자신감을 가질 수 있게 되었다.

이번 학기의 Node.js 와 Express 를 활용한 축구 커뮤니티 서버 개발 프로젝트는 많은 도전과 성취를 경험할 수 있는 소중한 기회였으며, 우리 팀은 이 경험이 우리의 개발자로서의 성장과 미래의 프로젝트에 큰 영감을 줄 것이라 생각한다.