



풋살경기 예약 플랫폼

# FLAB

→ 4season 박종훈 박혜영 백기림 이윤서

4team Project

# CONTENTS



MIRI  
INTELLIGENCE

1

PART  
01

- ① 프로젝트 개요
- ② 프로젝트 기능

- ① 플라스크 파이썬 기반으로 이용한  
풋살 구장 예약 중계 웹사이트 개발 프로젝트
- ② 풋살경기를 하고 싶은 사람들을 위한 서비스 구현

2

3

PART  
02

- ③ 기술 스택
- ④ 팀원 소개

- ③ HTML5, CSS3 , bootstrap5, jQuery, Flask ...
- ④ 백기림 박종훈 박혜영 이윤서

4

5

PART  
03

- ⑤ 개발 일정

- ⑤ 25.09.22~25.10.14

6

## PART 1 \* 프로젝트 개요



1

프로젝트 명

2

FLAB

3

4

개발 일정

5

25.09.22~25.10.14

6



DB와 연동하여 데이터관리

Front-end/back-end  
Flask development

누구나 자유롭게 접근 가능한 웹페이지

페이지 별 도메인 분리

# PART 1 \* 핵심 기능



1

로그인/회원가입

마이페이지

2



3

구장 정보 조회

충전하기

4

5

구장 예약취소

이용비 충전/환불

6

## PART 1 \* 핵심 기능



1

경기신청

매니저

2



3

구장리스트 확인

구장/시간추가

4

5

구장예약/및 결제

경기추가

6

## PART 2 \* 기술스택



1

2

3

4

5

6

Front-end

HTML5, CSS3, Bootstrap5,  
jQuery, Flatpickr.js

Back-end

Python (Flask)

DB

SQLAlchemy

Cloud

AWS Docker

ETC

Git, GitHub, Figma

## PART 2 \* 팀원분담



1

### Full-stack

2

백기림

박혜영

이윤서

박종훈

3

4

- 총괄
- DB설계

- 마이페이지 구현
- 머니페이지 구현

- HEADER/FOOTER  
BASE.html제작
- 매니저 페이지 구현

- STADIUM 페이지  
구현
- MATCHES 페이지  
구현

5

6

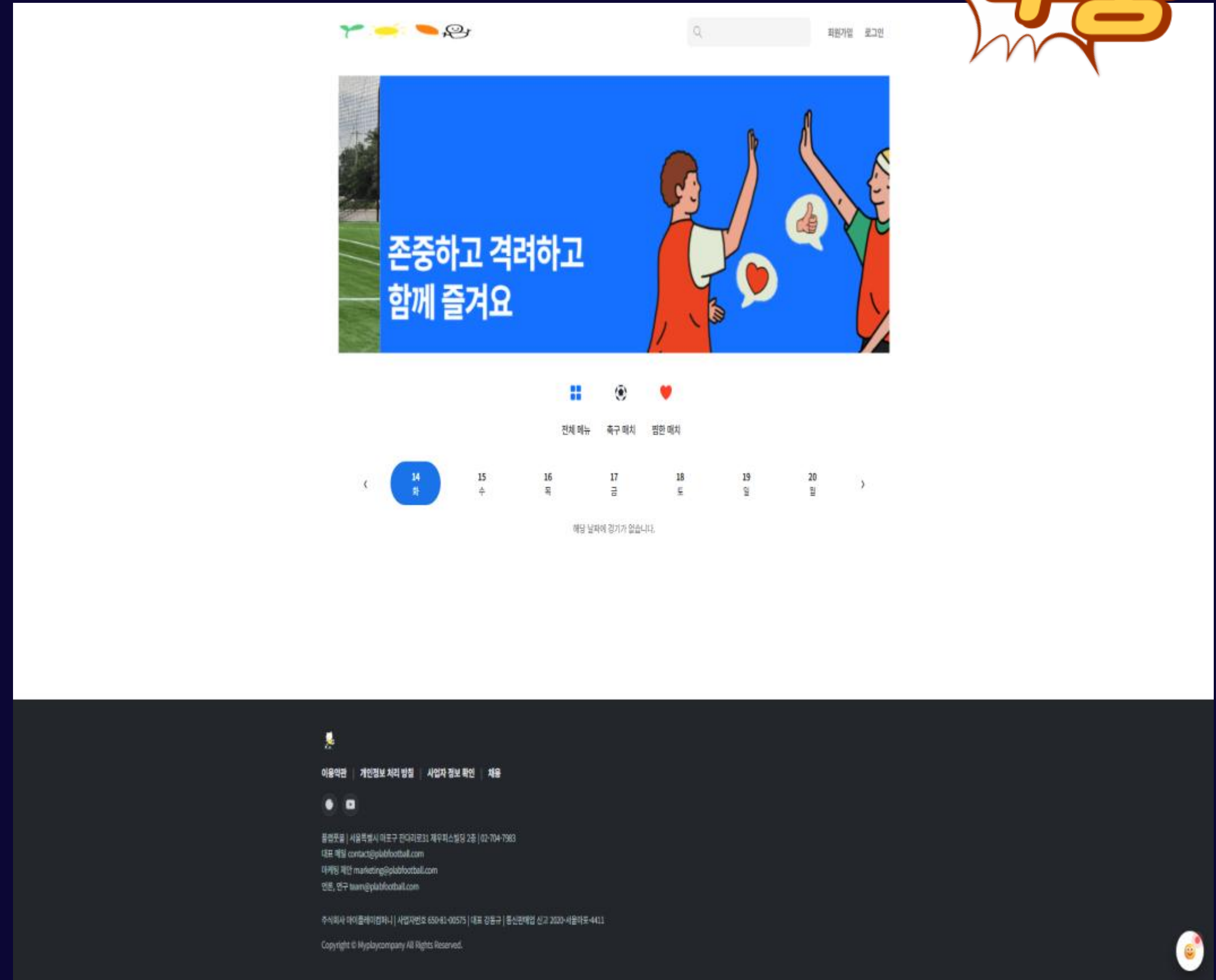
## 2. 기술스택(이운서) 메인페이지구현



두칫 두칫

# 메인페이지!

회원가입,로그인,축구매치등을 다른  
페이지로 이동할수있게  
만들었습니당~!





## 2. 기술스택(이운서) 매니저 페이지 구현

경기일 선택

연도-월-일

시간

09시  
10시  
11시  
12시  
14시  
16시

저장

이용약관 | 개인정보 처리 방침 | 사업자 정보 확인 | 채용

문의처: 서울특별시 강남구 테헤란로211 제우테크빌 2층 | 02-704-7963  
대표메일: contact@plabfootball.com  
마케팅 담당: marketing@plabfootball.com  
연락처: team@plabfootball.com

주식회사 마이플래이션(주) | 사업자번호: 830-81-00375 | 대표: 김동규 | 통신판매업 신고: 2020-서울강남-4421  
Copyright © Myplaycompany All Rights Reserved.

경기일 선택

경기시간 선택

2025-09-30 14:00 ~ 16:00

경기금액

저장

이용약관 | 개인정보 처리 방침 | 사업자 정보 확인 | 채용

문의처: 서울특별시 강남구 테헤란로211 제우테크빌 2층 | 02-704-7963  
대표메일: contact@plabfootball.com  
마케팅 담당: marketing@plabfootball.com  
연락처: team@plabfootball.com

주식회사 마이플래이션(주) | 사업자번호: 830-81-00375 | 대표: 김동규 | 통신판매업 신고: 2020-서울강남-4421  
Copyright © Myplaycompany All Rights Reserved.

경기추가

구단명

지역

연령: 5y5x5

금액: 9000원

등록

이용약관 | 개인정보 처리 방침 | 사업자 정보 확인 | 채용

문의처: 서울특별시 강남구 테헤란로211 제우테크빌 2층 | 02-704-7963  
대표메일: contact@plabfootball.com  
마케팅 담당: marketing@plabfootball.com  
연락처: team@plabfootball.com

주식회사 마이플래이션(주) | 사업자번호: 830-81-00375 | 대표: 김동규 | 통신판매업 신고: 2020-서울강남-4421  
Copyright © Myplaycompany All Rights Reserved.

# 구장추가, 시간추가, 경기추가

관리자가 손쉽게 추가하여 DB에 추가될 수 있도록 구현했습니다~!

## 2. 기술스택(박종훈)/stadium

```
@app.route('/', methods=['GET']) &lt;get>
def get_matches():
    date_str = request.args.get("date")

    if date_str:
        start_day = datetime.strptime(date_str, "%Y-%m-%d")
    else:
        start_day = datetime.today()

    start_day = start_day.replace(hour=0, minute=0, second=0, microsecond=0)
    end_day = start_day + timedelta(days=1)

    stadium_matches = (
        StadiumMatch.query
        .join(MatchTime)
        .filter(MatchTime.start_time >= start_day, MatchTime.start_time < end_day)
        .order_by(MatchTime.start_time)
        .all()
    )

    result = []
    for sm in stadium_matches:
        reserved = len(sm.user_matches)
        capacity = sm.stadium.headcount

        # 기본 상태
        status = "신청가능"
        if reserved >= capacity / 3 * 2 and reserved < capacity:
            status = "마감임박"
        elif reserved == capacity:
            status = "신청마감"

        # 로그인 유저가 이미 신청한 경우 -> 신청완료
        if g.user:
            if any(un.user_id == g.user.user_id for un in sm.user_matches):
                status = "신청완료"
```

오산 AJ 풋살파크

전체 메뉴 축구 펍서 빙판 매치

< 14 화 15 수 16 목 17 금 18 토 19 일 20 월 >

16:00  
신청가능 제주월드컵경기장

16:00  
신청가능 울산문수경기장

### Step 1 경기 목록 조회 (특정 날짜 기준)

#### 핵심 기능

사용자가 특정 날짜(또는 오늘)의 경기 리스트를 확인

날짜 파라미터 받기

date\_str = request.args.get("date")

→ 사용자가 선택한 날짜가 있으면 해당 날짜, 없으면 오늘 날짜로 설정

URL: /stadium/

기능: 사용자가 특정 날짜(또는 오늘)의 경기 리스트를 확인

#### 기술 포인트

ORM join/filter/order\_by로 SQL문 없이 DB조회

상태 로직 분기로 실시간 경기 상태 표시

날짜별 필터링을 datetime 객체로 정확히 처리

## 2. 기술스택(박종훈)/stadium

```
@bp.route(rule="/<int:match_id>", methods=["GET"]) ㄹ girint
def match_detail(match_id):
    match = StadiumMatch.query.get_or_404(match_id)

    reserved = len(match.user_matches)
    capacity = match.stadium.headcount

    # 기본 상태
    status = "신청가능"
    if reserved >= capacity / 3 * 2 and reserved < capacity:
        status = "미감입박"
    elif reserved == capacity:
        status = "신청마감"

    # ✅ 로그인 유저가 이미 신청한 경우 → 신청완료
    if g.user:
        if any(um.user_id == g.user.user_id for um in match.user_matches):
            status = "신청완료"

    return render_template(
        template_name_or_list="stadium/stadium_detail.html",
        match=match,
        stadium=match.stadium,
        reserved=reserved,
        capacity=capacity,
        status=status
    )
```



### 매치 정보

🔥 레벨: 모든 레벨      🏠 성별: 남녀 모두  
⌚ 경기 시간: 2시간      🏟️ 방식: 5vs5, 3쿼터  
정원: 18명 (현재 0명 신청)

📅 2025-10-14 16:00 ~ 18:00

### 제주월드컵경기장

📍 위치: 제주도

💰 참가비: 6000원

정원: 18명

현재 예약: 0명

상태: 신청가능

신청하기

목록으로

### Step2 경기 상세 페이지

#### 핵심 기능

#### 경기 ID 기반 데이터 조회

match = StadiumMatch.query.get\_or\_404(match\_id)

→ 없는 경기면 자동 404 반환

#### 참여자 수, 정원, 상태 계산

reserved = len(match.user\_matches)

capacity = match.stadium.headcount

→ 신청된 인원수 기반으로 상태 표시

render\_template("stadium/stadium\_detail.html", match=match, ...)

#### 기술 포인트

get\_or\_404로 에러 핸들링 내장

관계형 모델 연동

→ match.stadium, match.user\_matches 등 ORM 관계 매핑 활용

UI 상태 동적 표시

→ 신청자 수에 따라 버튼/문구 자동 변경

## 2. 기술스택(박종훈)/stadium

```
@bp.route(rule="/apply/<int:match_id>", methods=["POST"]) & gprint
def apply_match(match_id):
    match = StadiumMatch.query.get_or_404(match_id)

    # 로그인 확인
    if not g.user:
        flash(message="로그인이 필요합니다.", category="warning")
        return redirect(url_for("auth.login"))

    # 정원 마감 확인
    if len(match.user_matches) >= match.stadium.headcount:
        flash(message="정원이 마감되었습니다!", category="danger")
        return redirect(url_for(endpoint="stadium.match_detail", match_id=match.id))

    # 중복 신청 확인
    existing = UserMatch.query.filter_by(user_id=g.user.user_id, stadium_match_id=match.id).first()
    if existing:
        flash(message="이미 신청한 경기입니다!", category="info")
        return redirect(url_for(endpoint="stadium.match_detail", match_id=match.id))

    # 신청 저장
    booking = UserMatch(user_id=g.user.user_id, stadium_match_id=match.id)
    db.session.add(booking)
    db.session.commit()

    flash(message="신청이 완료되었습니다!", category="success")
    return redirect(url_for(endpoint="stadium.match_detail", match_id=match.id))
```

### Step 3 경기 신청 페이지

핵심 기능

중복 신청 방지

```
existing = UserMatch.query.filter_by(user_id=g.user.user_id,
stadium_match_id=match.id).first()
```

신청 데이터 저장

```
booking = UserMatch(user_id=g.user.user_id, stadium_match_id=match.id)
```

```
db.session.add(booking)
```

```
db.session.commit()
```

기술 포인트

Flask g 객체로 로그인 상태 전역 관리

SQLAlchemy session을 통한 트랜잭션 처리

Flash 메시지로 사용자 피드백

중복신청 및 정원체크 로직으로 데이터 무결성 보장

10월 14일 목요일

16:00 ~ 18:00

제주시 제주월드컵경기장

내 구분

0개 보유

캐시/포인트

6000원 사용

잔액 19000원

결제

이용금액

6000

캐시 사용

-6000원

추가 결제 금액

0원

원금영수증 div 만들기 그 여러개중 선택하는 액션으로 만들기

이것만큼 찍!

부상과 보험

환불안내

신청 전 상세 안내와 본 결제 페이지의 내용을 확인하였으면, 만 16세 이상 이용자 및 개인정보 제3자 제공에 동의합니다.

신청하기

## 2. 기술스택(박종훈)/stadium

```
@bp.route("/mylist/<int:user_id>") & gprint
def my_match_list(user_id: int):
    date_str = request.args.get("date")
    query = UserMatch.query.filter_by(user_id=user_id).join(StadiumMatch).join(MatchTime)

    if date_str:
        try:
            selected_date = datetime.strptime(date_str, format="%Y-%m-%d").date()
            query = query.filter(db.func.date(MatchTime.start_time) == selected_date)
        except ValueError:
            selected_date = datetime.today().date()
    else:
        selected_date = datetime.today().date()

    my_matches = query.order_by(MatchTime.start_time).all()

    return render_template(
        template_name_or_list="matches/match_list.html",
        my_matches=my_matches,
        selected_date=selected_date
    )
```



### 내 신청 내역

10월 06일 14:00

서울시 마포구 · 서울월드컵경기장

정원: 18명 / 현재 1명 신청

신청내역 보기

날짜를 선택하면 해당 날짜의 내 신청 경기를 보여줍니다.

### Step 1 내 신청 내역

핵심 동작

날짜 파라미터 처리

date\_str = request.args.get("date")

URL에서 jQuery형태 (?date=2025-10-13) 로 날짜 전달 가능

없을 시, default값은 오늘 날짜

DB 조회 (ORM)

query = UserMatch.query.filter\_by(user\_id=user\_id)

.join(StadiumMatch).join(MatchTime)

UserMatch → StadiumMatch → MatchTime 관계를 join

유저가 참여 중인 경기만 추출

날짜 필터링

db.func.date(MatchTime.start\_time) == selected\_date

기술 포인트

SQLAlchemy의 join + func.date 활용

날짜 선택 UI(캘린더)와 연동

데이터는 전부 ORM 객체 기반으로 전달



## 2. 기술스택(박종훈)/stadium

```
@bp.route("/detail/<int:usermatch_id>") @girint
def my_match_detail(usermatch_id):
    user_match = UserMatch.query.get_or_404(usermatch_id)
    stadium_match = user_match.stadium_match
    return render_template(
        template_name_or_list="matches/match_detail.html",
        stadium_match=stadium_match,
        usermatch_id=usermatch_id)
```

**All**  
**social futsal**

Date  
2025년 10월 06일

stadium  
서울시 마포구 서울월드컵경기장

type  
장내모두 6vs6

time  
4:00~16:00

결제정보

이용금액  
0000

입시  
0000

신청 취소

입장 후 30분 이내에는 무료 취소 가능해요.

### Step 2 상세페이지

핵심 기능

신청 내역 단일 조회

user\_match = UserMatch.query.get\_or\_404(usermatch\_id)

→ 없는 데이터면 자동 404

관계 매핑 활용

stadium\_match = user\_match.stadium\_match

→ UserMatch 모델의 FK로 연결된 StadiumMatch 정보 가져오기

템플릿 렌더링

render\_template("matches/match\_detail.html", stadium\_match=stadium\_match)

술 포인트

ORM 관계 참조 (user\_match.stadium\_match)

템플릿에서 연관 데이터 표시

404 자동 처리로 예외 최소화

## 2. 기술스택(박종훈)/stadium

### Step 3 신청 취소

핵심 기능

폼 데이터 수신

match\_id = request.form.get("match\_id")

→ HTML form에서 전달된 match\_id 파라미터 확인

데이터 검증

user\_match = UserMatch.query.filter\_by(usermatch\_id=match\_id).first()

잘못된 요청, 없는 ID 처리

유효하지 않은 경우 flash로 오류 메시지

DB 삭제 및 커밋

db.session.delete(user\_match)

db.session.commit()

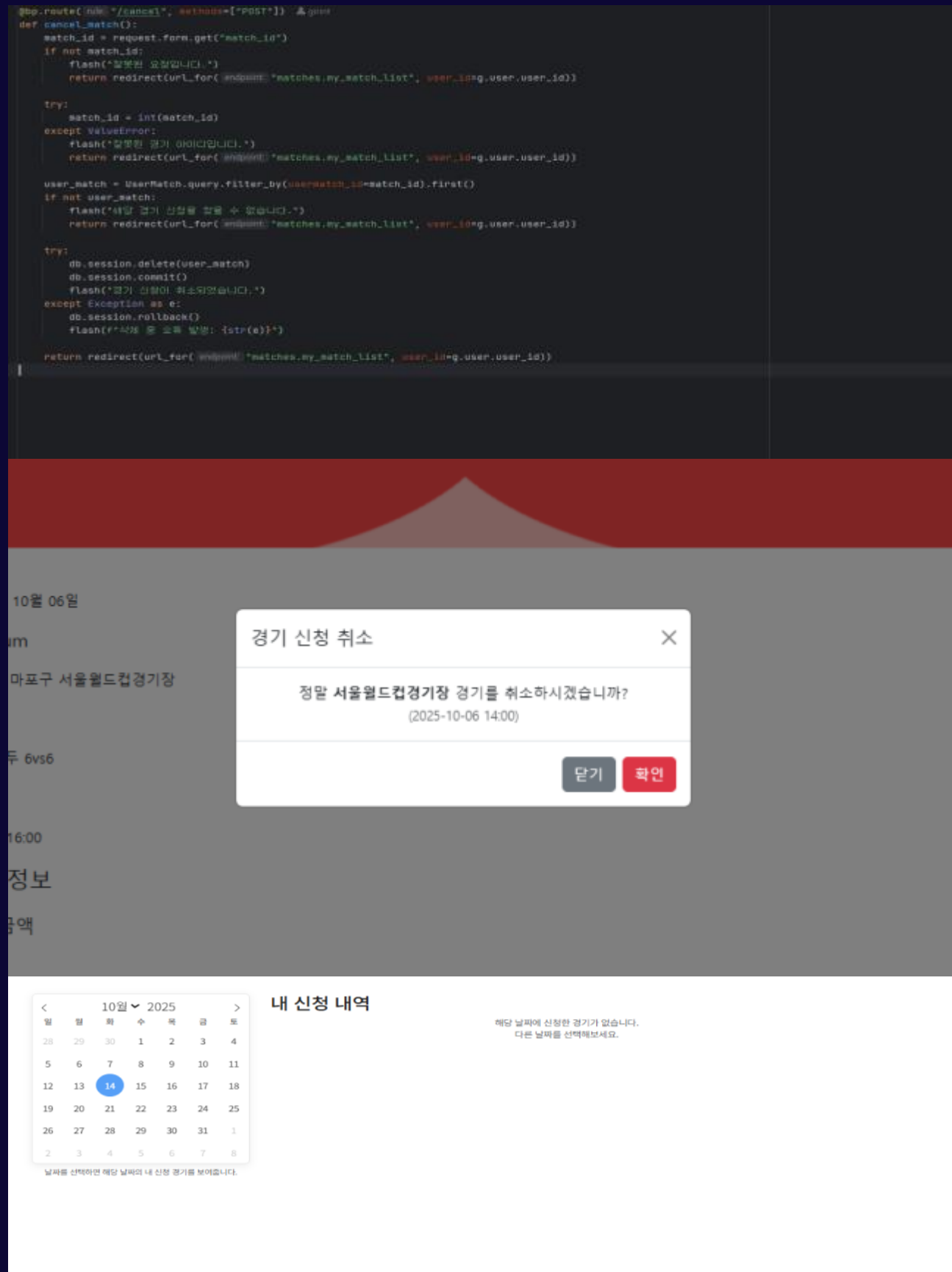
→ 실제 데이터 삭제

기술 포인트

POST 요청 보호 (데이터 변경 시)

트랜잭션 처리 + 예외 복구

flash 메시지로 UI 피드백 제공



## 2. 기술스택(박혜영) auth 페이지 제작

### 회원가입

닉네임

비밀번호

비밀번호 확인

이메일

휴대폰 번호

생년월일  
년도  월  일

성별

☐ 전체 동의  
☐ 만 16세 이상입니다 (필수)  
☐ 개인정보 수집 및 이용 동의 (필수)  
☐ 이용약관 동의 (필수)  
☐ 개인정보 마케팅 동의 (선택)  
☐ 이벤트 및 맞춤 정보 문자 수신 (선택)

```
<script>
// 전체 동의 체크박스 ID들 (jQuery가 id를 몰드명으로 현대화)
const ids = {
  all: 'agree_all',
  children: ['agree_age', 'agree_personal', 'agree_terms', 'agree_marketing', 'agree_event']
};

const $all = document.getElementById(ids.all);
const $children = ids.children.map(id => document.getElementById(id)).filter(Boolean);

if ($all && $children.length) {
  // 마스터 → 자식
  $all.addEventListener('change', () => {
    $children.forEach(cb => {
      cb.checked = $all.checked;
    });
  });
  // 자식 → 마스터
  $children.forEach(cb => {
    cb.addEventListener('change', () => {
      $all.checked = $children.every(x => x.checked);
    });
  });
}
```

성별

☒ 전체 동의  
☒ 만 16세 이상입니다 (필수)  
☒ 개인정보 수집 및 이용 동의 (필수)  
☒ 이용약관 동의 (필수)  
☒ 개인정보 마케팅 동의 (선택)  
☒ 이벤트 및 맞춤 정보 문자 수신 (선택)

전체동의 클릭시  
나머지  
체크박스들 함께  
체크

### 로그인

사용자 이름

비밀번호

[회원가입](#) [로그인](#)

[이용약관](#) | [개인정보 처리 방침](#) | [사업자 정보 확인](#) | [채용](#)

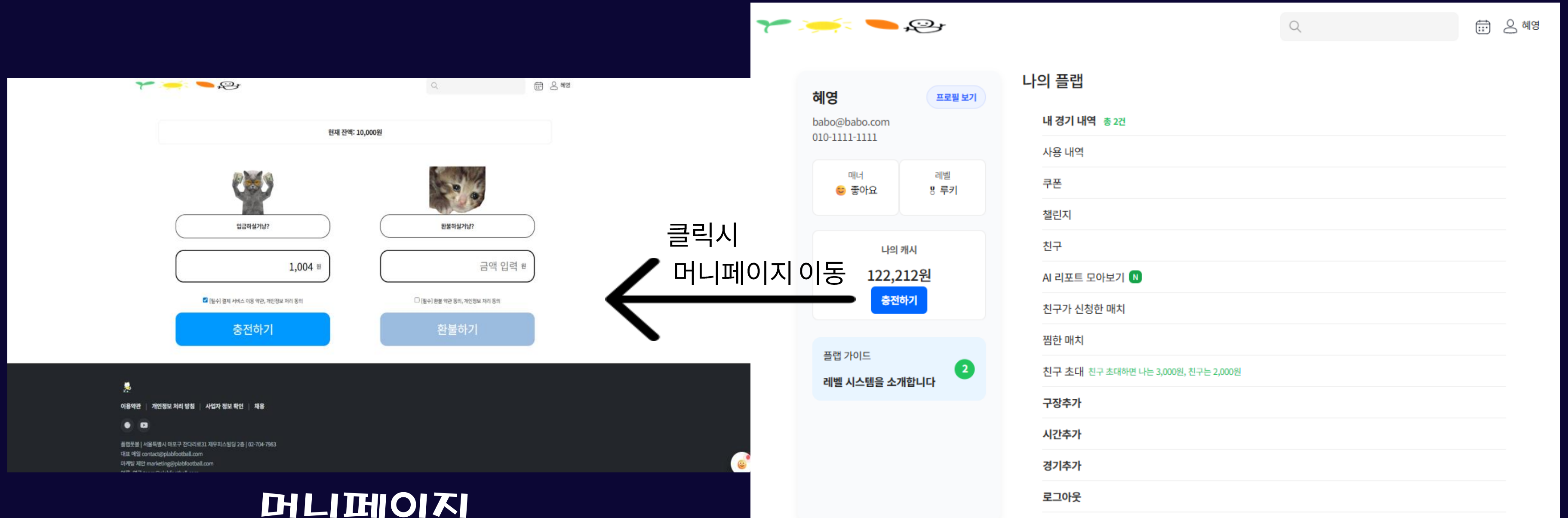
플랜트볼 | 서울특별시 마포구 한대리31 제우퍼스트빌딩 2층 | 02-704-7983  
대표 이메일 [contact@plabfootball.com](mailto:contact@plabfootball.com)  
마케팅 채널 [marketing@plabfootball.com](mailto:marketing@plabfootball.com)  
인력, 연구 [team@plabfootball.com](mailto:team@plabfootball.com)

주식회사 마이플레이컴퍼니 | 사업자번호 650-81-00575 | 대표 강동규 | 통신판매업 신고 2020-서울마포-4411  
Copyright © Myplaycompany All Rights Reserved.

회원가입 완료하면 자동으로  
로그인 페이지 연결



## 2. 기술스택(박혜영) 머니페이지 마이페이지 구현



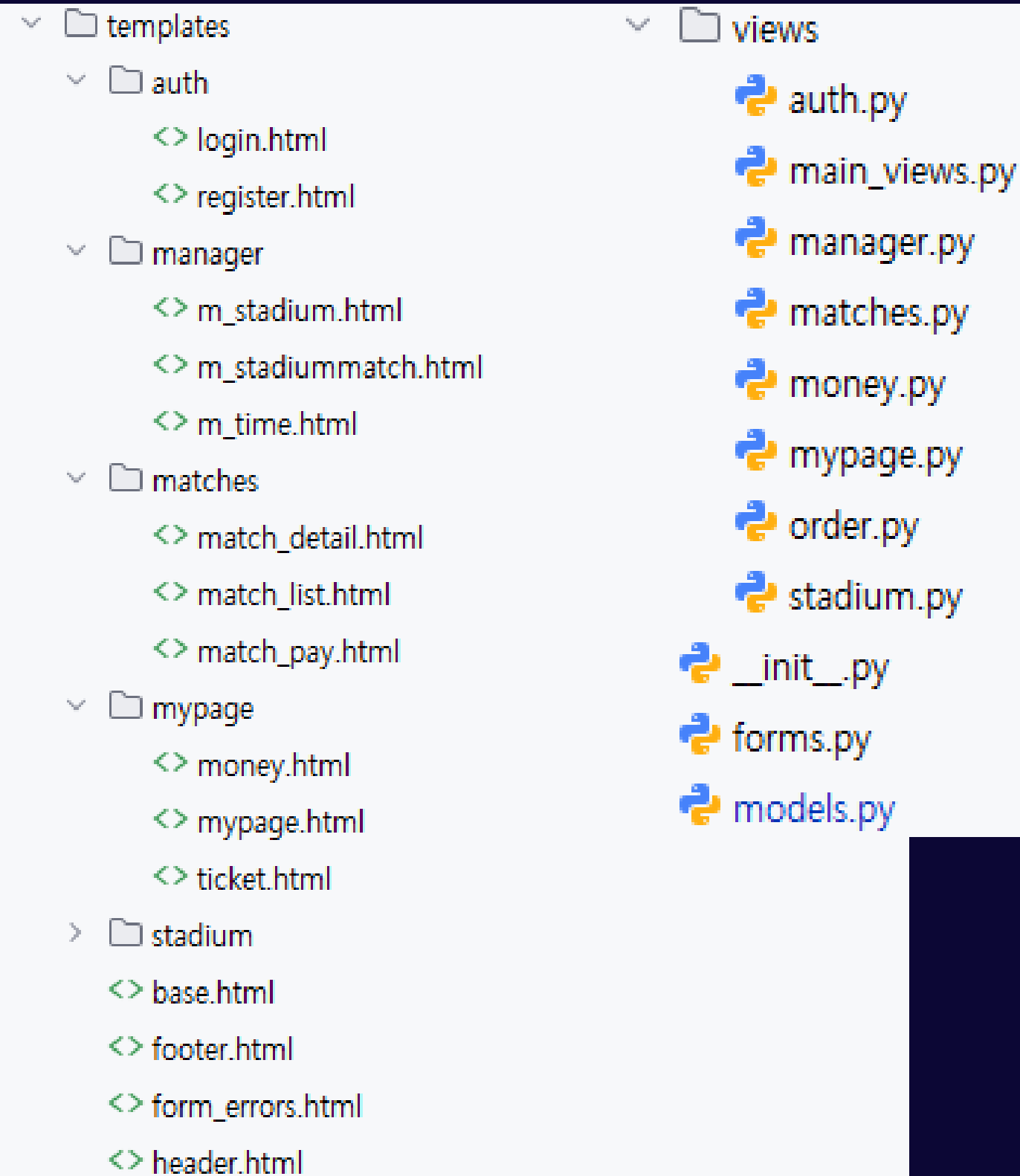
### 머니페이지

돈을 충전하고 환불할 수 있는 시스템  
구현

### 마이페이지 구현

백엔드와 연결하여 나의 경기신청 내역과 캐시  
비동기 정보 가져오고 보내기

## 2. 기술스택(백기름) 템플릿과 백엔드 구조 설계



- 핵심 기능별로 파일과 폴더를 체계적으로 분리
- 서비스 흐름의 일관성 확보
- 역할별 모듈화로 유지보수 확장성 강화
- 조직화를 주도하여 효율적 협업 제공

## 2. 기술스택(백기름) 데이터베이스 설계 및 관리

```
from setting import db

# User 모델
class User(db.Model):
    __tablename__ = "user"
    user_id = db.Column(db.Integer, primary_key=True)
    password = db.Column(db.String(200), nullable=False)
    email = db.Column(db.String(200), nullable=False)
    money = db.Column(db.Integer, nullable=True, default=0)
    nickname = db.Column(db.String(10), nullable=False)
    photo_number = db.Column(db.String(15), nullable=False)

    match_list = db.relationship("UserMatch", back_populates="user", cascade="all, delete-orphan")

# Stadium 모델
class Stadium(db.Model):
    __tablename__ = "stadium"
    stadium_id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
    location = db.Column(db.String(200), nullable=False)
    photo = db.Column(db.String(300))
    headcount = db.Column(db.Integer, nullable=False)
    entry_fee = db.Column(db.Integer, nullable=False)

    stadium_match = db.relationship("StadiumMatch", back_populates="stadium")

# MatchTime 모델
class MatchTime(db.Model):
    __tablename__ = "matchtime"
    matchtime_id = db.Column(db.Integer, primary_key=True)
    start_time = db.Column(db.DateTime, nullable=False)
    end_time = db.Column(db.DateTime, nullable=False)

    stadium_match = db.relationship("StadiumMatch", back_populates="matchtime")

.
.
.
.

# Manager 모델
class Manager(db.Model):
    __tablename__ = "manager"
    manager_id = db.Column(db.Integer, primary_key=True)
    password = db.Column(db.String(20), nullable=False)
    email = db.Column(db.String(200), nullable=False)
    nickname = db.Column(db.String(10), nullable=False)
    number = db.Column(db.String(15))

    stadium_match = db.relationship("StadiumMatch", back_populates="manager")
```



- 데이터 모델링 및 관계 설정
- 효율적인 데이터 CRUD 및 트랜잭션 처리
- 안정적인 서버-DB 연동 및 데이터 무결성 유지



마무리

1

2

3

4

5

6



**thank  
you**

