JUST DO FITTIES

Better health with Fitties

성민기 윤여준

역할

팀원 소개

성민기

백엔드 40%

프론트엔드 60%

윤여준

백엔드 60%

프론트엔드 40%

Fitties

기획 배경

경쟁

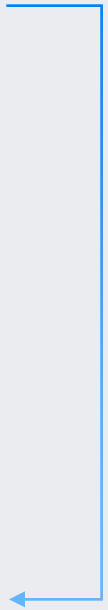
나와 유사한 사람

성장

눈에 보이는 차이점

동료

친구, 동료, 새로운사람



동기 부여

개발 환경

Clova

ChatGPT

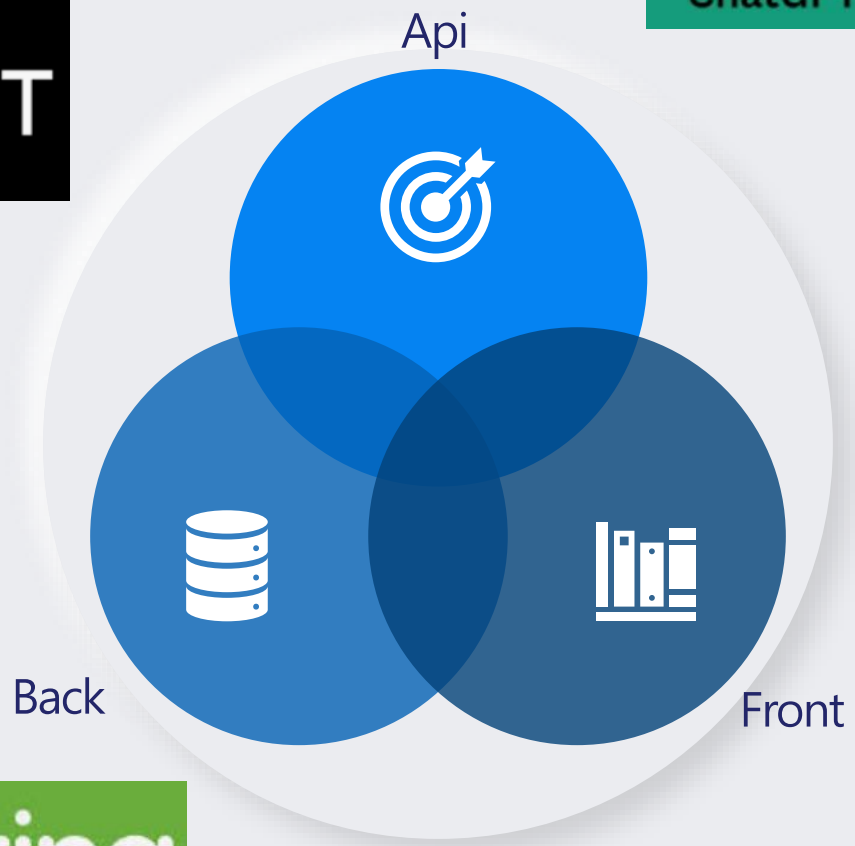
JWT

MyBatis

spring SECURITY

Vue.js

MySQL

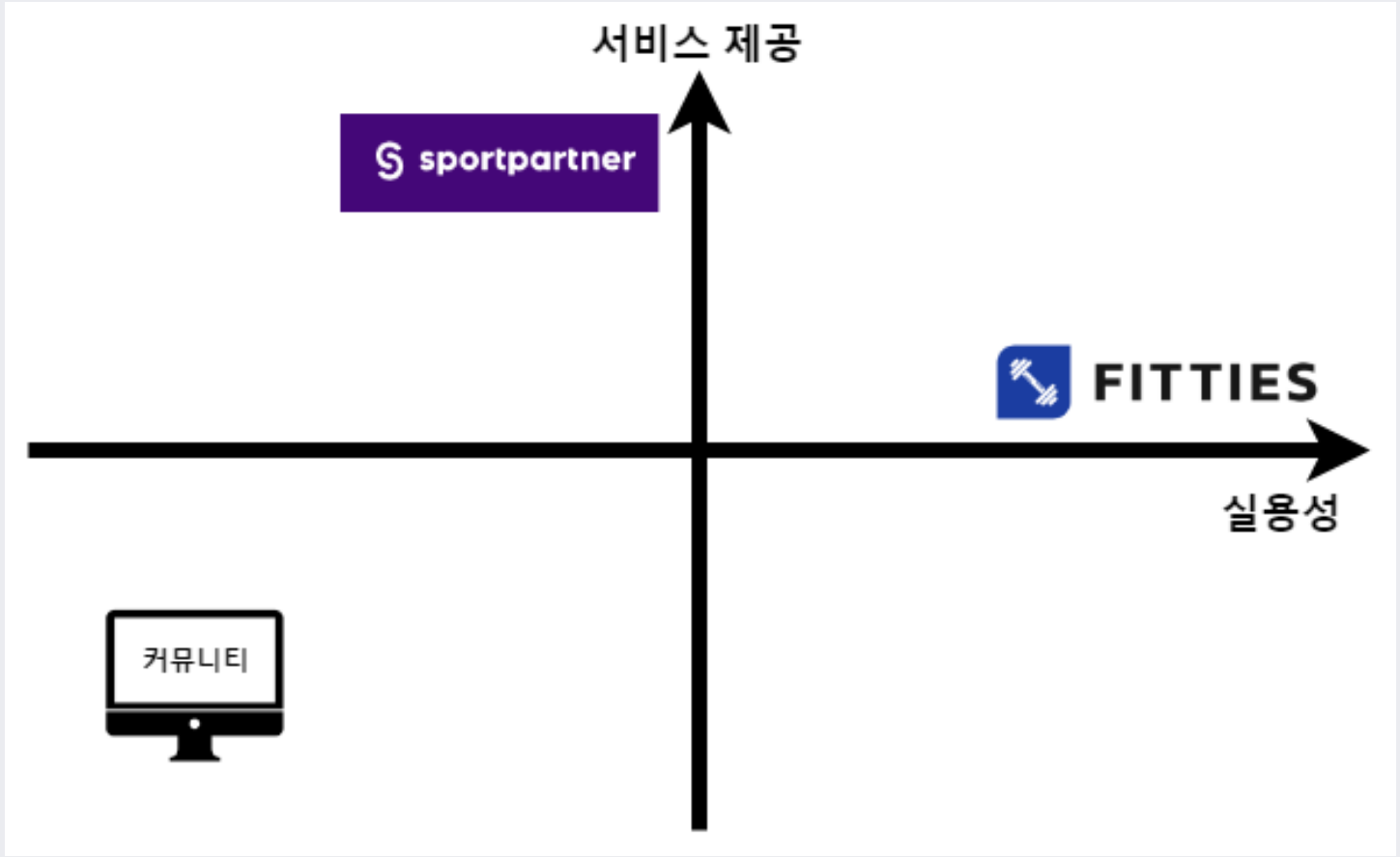


spring Boot

Bootstrap

시장분석현황

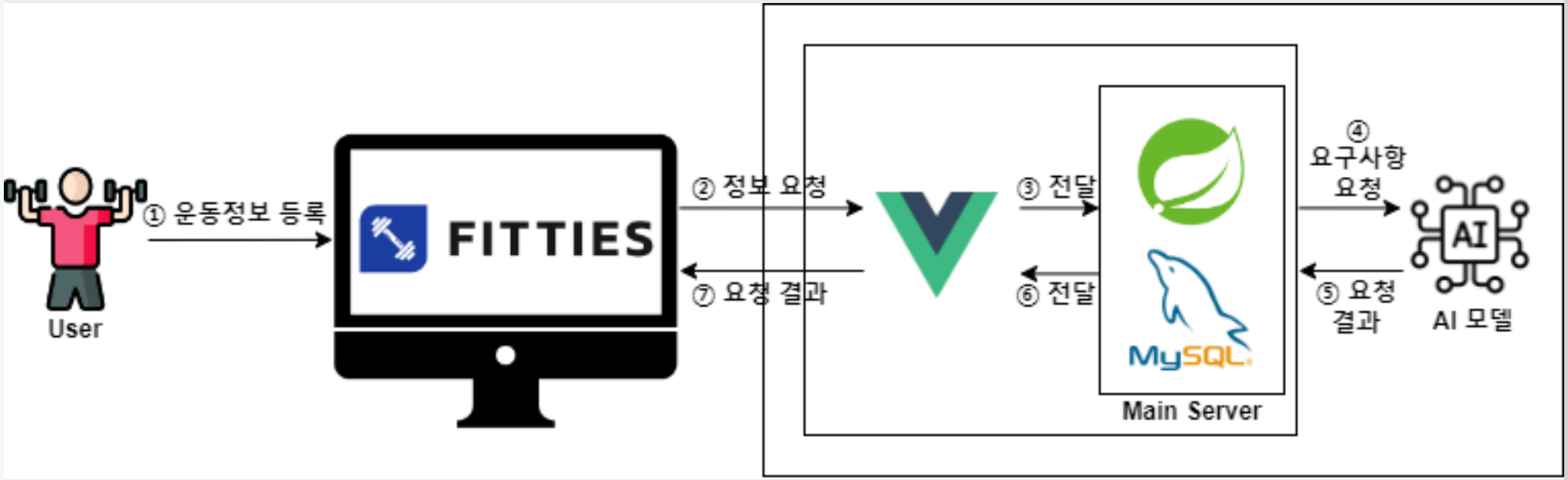
아이템 포지셔닝



아이템 포지셔닝

개발 결과

시스템 구조도



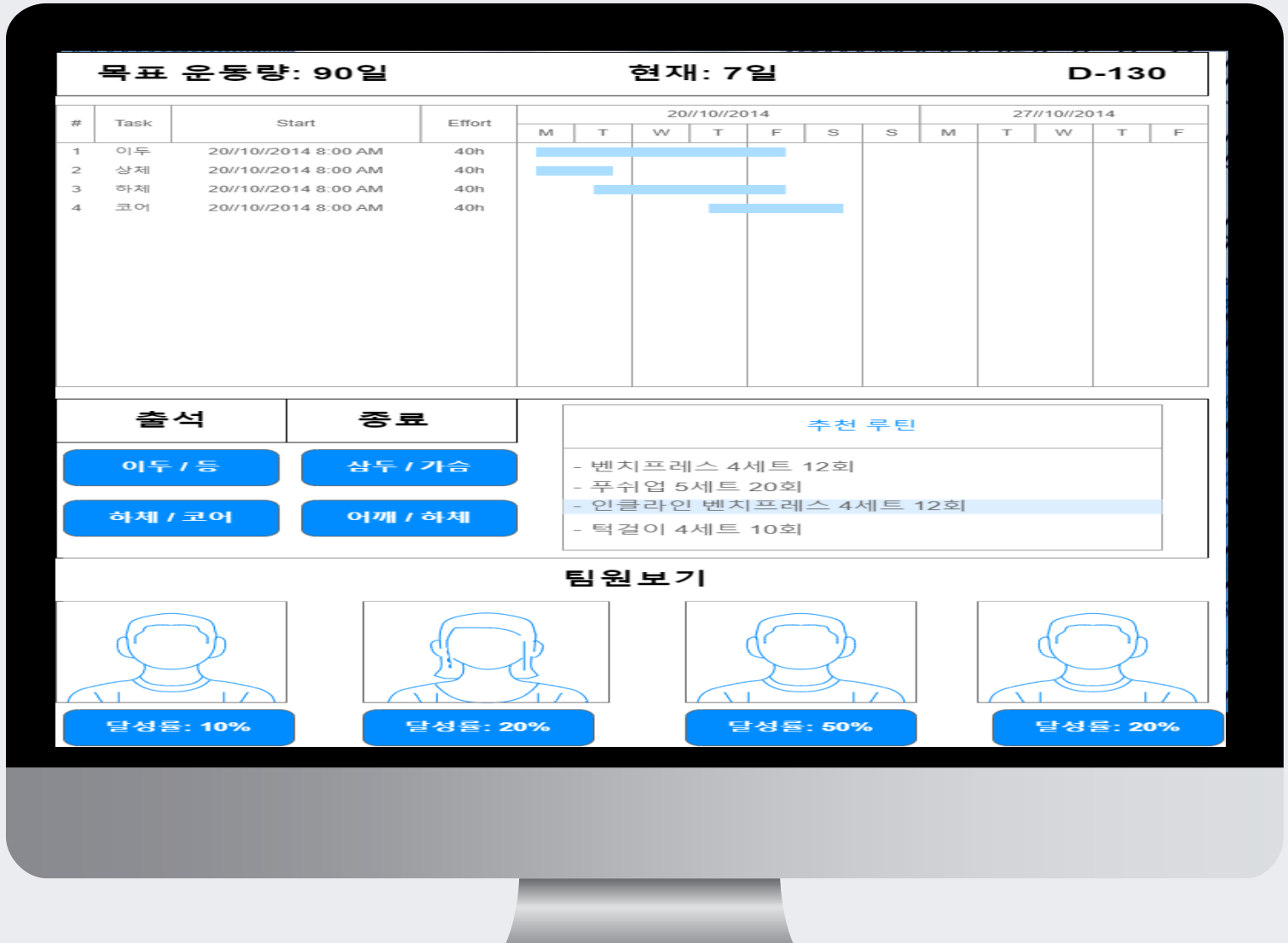
시스템 구성도 설계

Mock up

프로젝트 진행 계획

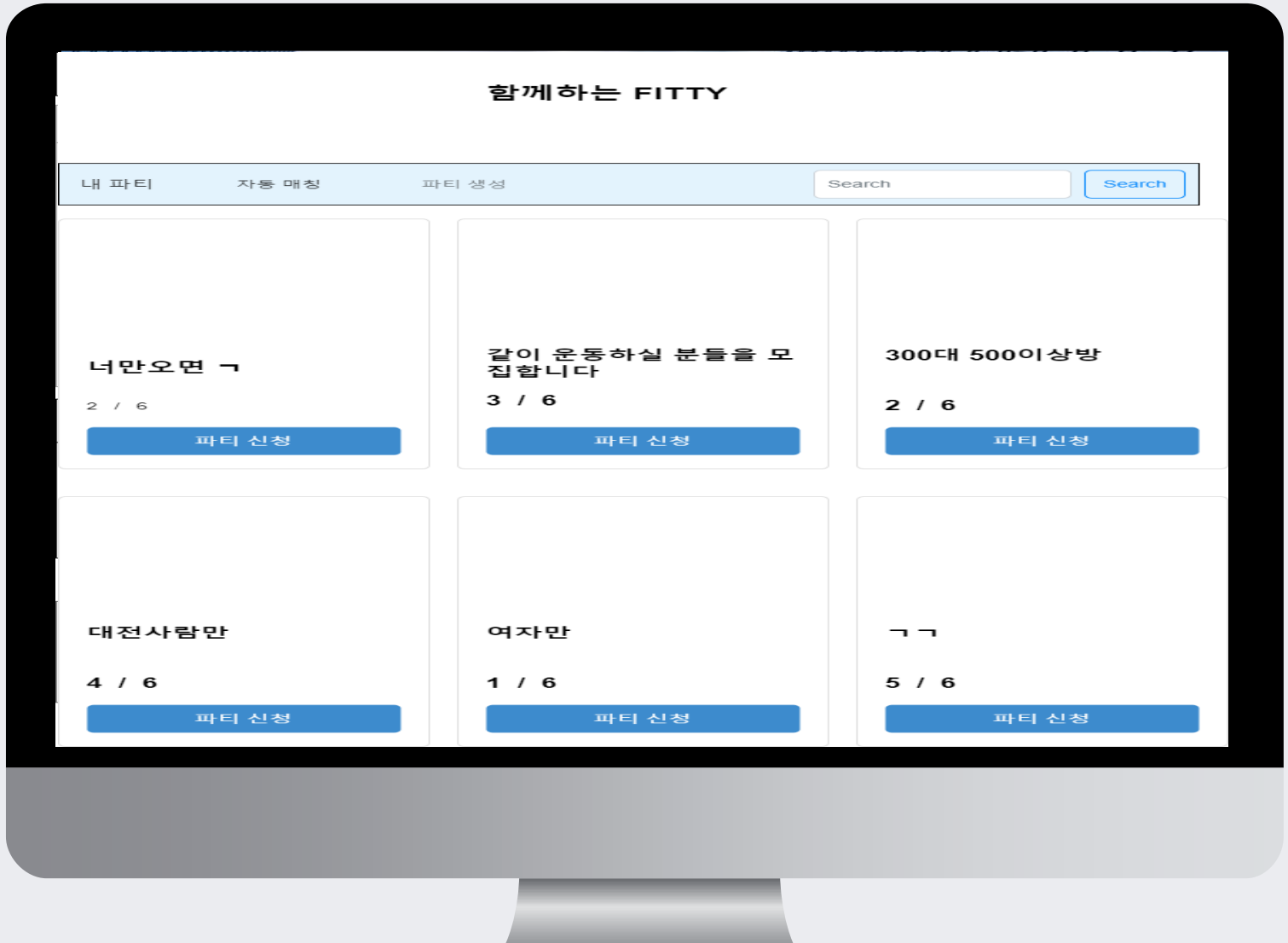
Mock up

프로젝트 진행 계획



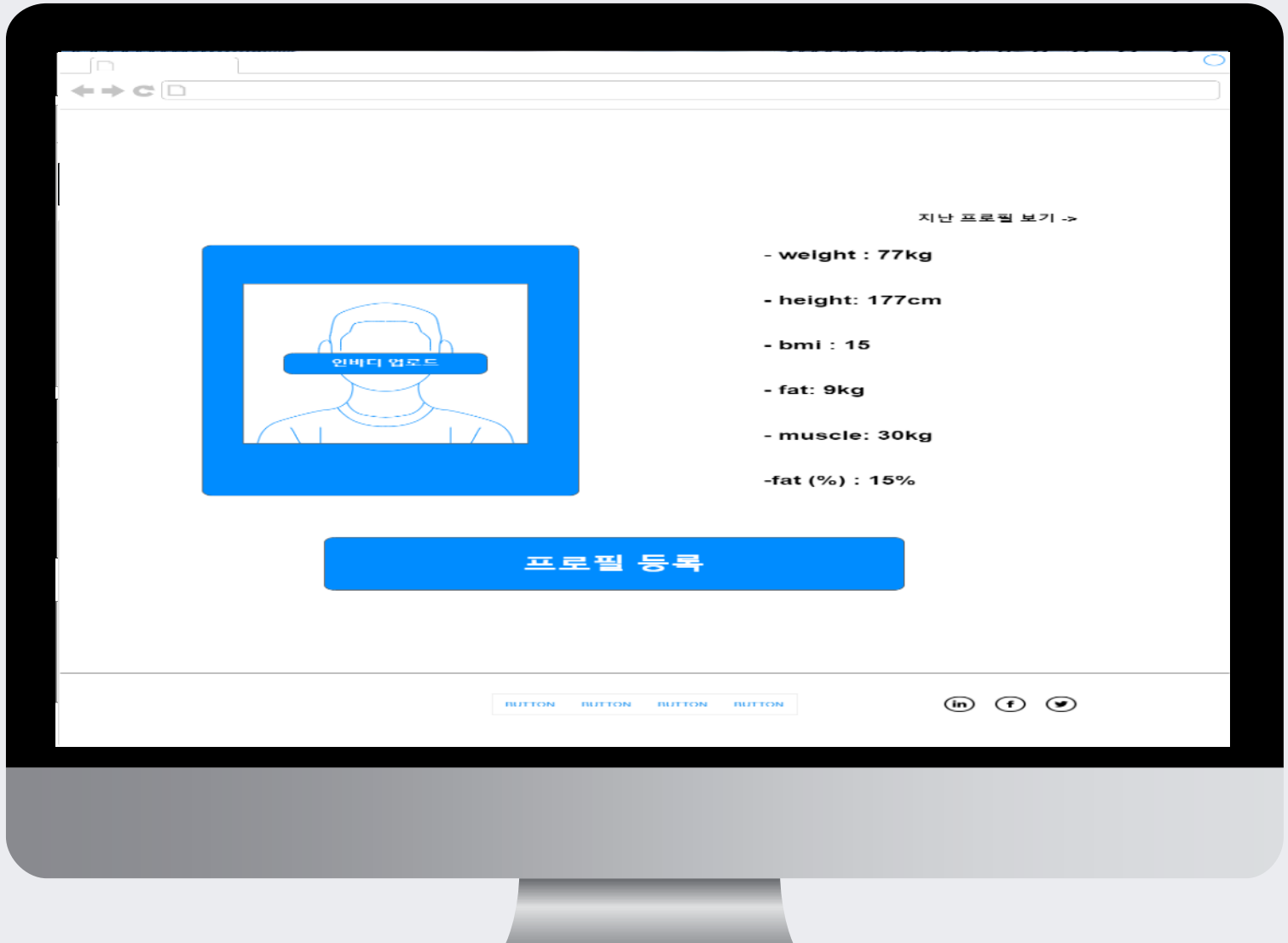
Mock up

프로젝트 진행 계획



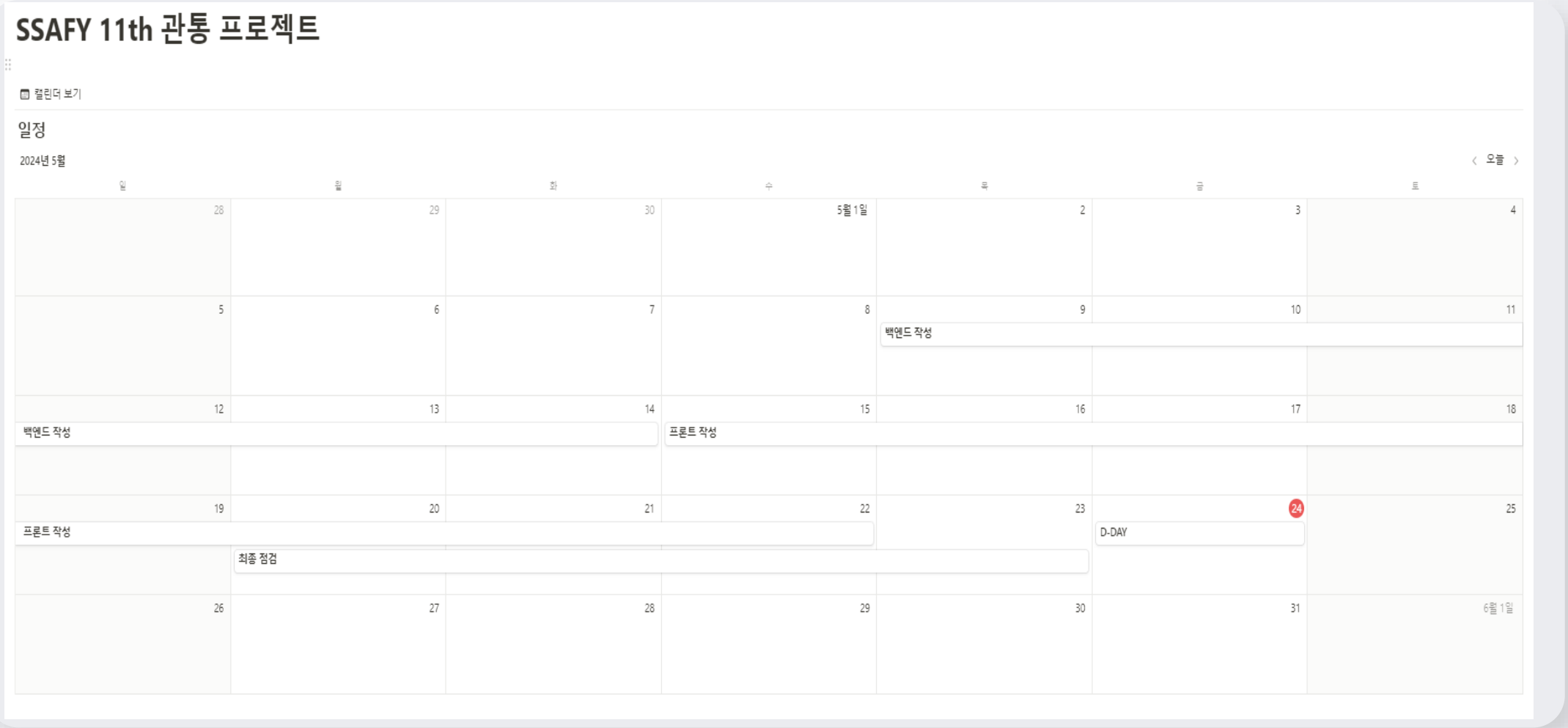
Mock up

프로젝트 진행 계획



전체 일정

프로젝트 진행 계획



간트 차트

프로젝트 진행 계획

SSAFY 11th 관통 Gantt Chart

프로젝트 이름	SSAFY 11th 관통프로젝트	프로젝트명	Fitties
프로젝트 팀 구성원	윤여준 설민기	날짜	2024-05-16 ~ 2024-05-24

WBS 번호	작업 제목	담당자	시작일	종료일	기간	작업 완료 비율	SSAFY 11th 관통 프로젝트 (2024-05) 16일	17일	18일	19일	20일	21일	22일	23일	24일
1	프로젝트 기획														
1.1	요구사항 정의서 작성		24년 5월 16일	24년 5월 16일	1	100%									
1.2	데이터 구조도(ERD)		24년 5월 16일	24년 5월 16일	1	100%									
1.3	UseCase 다이어그램		24년 5월 16일	24년 5월 16일	1	100%									
1.4	클래스 다이어그램		24년 5월 16일	24년 5월 17일	2	100%									
1.5	요구사항 명세서 작성		24년 5월 17일	24년 5월 17일	1	100%									
2	BackEnd														
2.1	로그인 기능		24년 5월 17일	24년 5월 18일	2	100%									
2.2	자유게시판 기능		24년 5월 17일	24년 5월 19일	3	100%									
2.3	캘린더 기능		24년 5월 18일	24년 5월 19일	2	100%									
2.2	파티게시판 기능		24년 5월 19일	24년 5월 20일	2	100%									
2.2	랭킹 기능		24년 5월 20일	24년 5월 21일	2	100%									
2.2	기능 테스트		24년 5월 21일	24년 5월 21일	1	100%									
3	FrontEnd														
3.1	로그인 페이지		24년 5월 18일	24년 5월 19일	2	100%									
3.2	자유게시판 페이지		24년 5월 19일	24년 5월 20일	2	100%									
3.3	캘린더 페이지		24년 5월 19일	24년 5월 20일	2	100%									
3.4	파티게시판 페이지		24년 5월 20일	24년 5월 21일	2	100%									
3.5	랭킹 페이지		24년 5월 20일	24년 5월 21일	2	100%									
3.6	메인 페이지		24년 5월 20일	24년 5월 21일	2	100%									
3.7	페이지 디자인		24년 5월 21일	24년 5월 23일	3	100%									
4	결과 준비														
4.1	코드 리뷰		24년 5월 18일	24년 5월 22일	5	100%									
4.2	발표 자료 준비		24년 5월 22일	24년 5월 23일	2	100%									
4.3	최종 점검		24년 5월 23일	24년 5월 23일	1	100%									
4.4	발표		24년 5월 24일	24년 5월 24일	1	100%									

회의록

프로젝트 진행 계획

회의록

Aa 이름	태그	생성 일시
240516 회의록	오전 회의	2024년 5월 16일 오후 2:36
240516 회의록	오후 회의	2024년 5월 16일 오후 3:53
240516 회의록	저녁 회의	2024년 5월 16일 오후 3:53
240517 회의록	오전 회의	지난주 금요일 오전 9:04
240518 회의록	오후 회의	지난주 토요일 오후 8:17
240519 회의록	새벽 회의	지난주 일요일 오전 1:10
240520 회의록	새벽 회의	월요일 오전 12:15
240521 회의록	오전 회의	화요일 오전 9:09
240522 회의록	오전 회의	수요일 오전 11:06
240523 회의록	오전 회의	어제 오전 11:52

240518 회의록

태그 오후 회의

생성 일시 지난주 토요일 오후 8:17

속성 추가

댓글 추가

Fitty 기능

- 파티 방장 권한
  - 수정
  - 삭제
  - 탈퇴
  - 강퇴
  - 가입신청
  - 가입수락 등 수행
- 매칭 기능 수행

Ranking 기능

- 만들기 시작

작업 리스트

프로젝트 진행 계획

활동 내용

Aa 이름	날짜	태그	다중 선택	상태
240516 작업사항 (FrontEnd)	2024년 5월 16일	FrontEnd	User	완료
240517 작업사항 (FrontEnd || BackEnd)	2024년 5월 17일	FrontEnd BackEnd	User Fitty	완료
240518 작업사항 (FrontEnd || BackEnd)	2024년 5월 18일	FrontEnd BackEnd	Ranking Fitty	완료
240519 작업사항	2024년 5월 19일	FrontEnd BackEnd	Fitty Ranking	완료
240520 작업사항	2024년 5월 20일	FrontEnd BackEnd	Fitty Ranking	완료
240521 작업사항	2024년 5월 21일	FrontEnd BackEnd	Ranking Calendar	완료
240522 작업사항	2024년 5월 22일	FrontEnd BackEnd	OCR	진행 중
240523 작업사항	2024년 5월 23일	FrontEnd	all	시작 전
Final Test Routine				시작 전

Session Storage 저장 문제 해결

- Login 정보를 백엔드에서 받아오는 과정에서 문제 발생해서 이 문제를 해결하기 위해서 Pinia persist?로 해결해볼 예정

Fitty 페이지 제작해보기

- 데이터베이스에 저장된 파티 정보를 가져와서 활용
  - 파티 생성 후 화면에 그 리스트 보이기
    - 파티의 인원이 6명이 넘김 || 비공개 상태 이면 화면에 파티리스트로 보이지 않기
      - (백엔드 fittyService 부분을 수정)
    - 이미 가입된 파티가 있다면 생성X
  - 파티 상세 현황
    - 가입된 파티 상태 보이는 것
    - 가입되어 있는 파티가 방장인 상태라면 자신이 가입되어있는 파티 멤버, 파티 가입 신청을 해놓은 게스트 현황을 보인다. (파티 관리 가능)

수정사항 리스트

프로젝트 진행 계획

수정사항 리스트

리스트

오류 발생 리스트

- RequestBody 문제 (240516)
- Session Storage 저장 문제 (240517)
- Fitty 기능 MySQL 값 연동 & Token 문제(240518)
- vue 생성주기 관련 nullPointException 문제 (240521)
- Test 주의사항 (240521)

RequestBody 문제 (240516)

태그 비어 있음

생성 일시 2024년 5월 16일 오후 3:04

속성 추가

댓글 추가

RequestBody 속성 정리

- Talend, Swagger로 백엔트 기능 테스트를 진행할 때는 괜찮았지만, Vue.js와의 연동 과정에서 User의 정보를 프론트에 가져오지 못하는 문제가 발생
  - 해결 방안
    Controller.java 파일에서 User 정보를 가져오는 @RequestBody의 import를 바꿔준다.

⚠ 변경 전:
import io.swagger.v3.oas.annotations.parameters.RequestBody;
변경 후:
import
org.springframework.web.bind.annotation.RequestBody;

To Do List

프로젝트 진행 계획

백엔드

분류	소분류	TODO	상태	작업일자	작업자	비고
User	User	로그인 기능	완료	2024년 5월 9일		
User	User	회원가입 및 탈퇴 기능	완료	2024년 5월 9일		
User	User	회원정보 수정 기능	완료	2024년 5월 9일		Token 검증
User	Rank	랭킹정보 등록, 수정, 삭제 기능	완료	2024년 5월 9일		
User	Rank	회원 랭킹정보 조회 기능	완료	2024년 5월 9일		Tokne 검증
User	Profile	회원 인바디정보 등록, 삭제, 조회 기능	완료	2024년 5월 9일		수기 입력방식, 텍스트를 읽어들이는 방법을 고려
User	Profile	모든 회원의 최신 인바디정보 조회 기능	완료	2024년 5월 12일		user_id 기준 Distinct
User	Point	회원의 포인트정보 등록, 업데이트, 삭제 기능	완료	2024년 5월 9일		
User	Point	모든 회원의 포인트 조회, 개별 열기 조회 기능	완료	2024년 5월 9일		
BoardFree	Comment	대댓글 작성, 수정, 삭제 기능	완료	2024년 5월 13일		
BoardFree	LikeCon	좋아요 기능	완료	2024년 5월 11일		
BoardFree	Comment	댓글 작성, 수정, 삭제 기능	완료	2024년 5월 13일		
BoardFree	Board	게시글 작성, 수정, 삭제, 조회, 검색 기능	완료	2024년 5월 13일		
Util	UploadFile	파일 업로드, 삭제, 조회 기능	완료	2024년 5월 10일		
Util	Ranking	랭킹보드 생성 및 삭제 기능	완료	2024년 5월 11일		UserRank정보 사용
Calendar	Schedule	목표 스케쥴 생성, 삭제 기능	완료	2024년 5월 10일		
Calendar	Routine	루틴 생성, 수정, 삭제, 조회 기능	완료	2024년 5월 10일		
Calendar	Routine	운동 루틴 추천 기능	완료	2024년 5월 10일		ChatGPT api 사용
Calendar	Detail	운동기록 생성, 조회, 수정, 삭제 기능	완료	2024년 5월 12일		
Calendar	Routine	운동 정보 조회 기능	완료			ChatGpt api 사용
Calendar	Fitty	같은 fitty원들의 정보 조회	진행 중			
Fitty	Fitty	파티 생성, 삭제, 조회, 매칭 기능	완료	2024년 5월 13일		매칭 알고리즘 (피어슨 상관계수 / 유클리드 거리 유사도)
Fitty	Board	파티모집 게시글 작성, 수정, 삭제, 조회, 검색 기능	완료	2024년 5월 12일		

To Do List

프로젝트 진행 계획

프론트엔드 기능

분류	소분류	TODO	상태	작업일자	작업자	비고
User	User	로그인 기능	완료	2024년 5월 9일		
User	User	회원가입 및 탈퇴 기능	완료	2024년 5월 9일		
User	User	회원정보 수정 기능	완료	2024년 5월 9일		Token 검증
User	Rank	랭킹정보 등록, 수정, 삭제 기능	완료	2024년 5월 9일		
User	Rank	회원 랭킹정보 조회 기능	완료	2024년 5월 9일		Tokne 검증
User	Profile	회원 인바디정보 등록, 삭제, 조회 기능	완료	2024년 5월 9일		수기 입력방식, 텍스트를 읽어들이는 방법을 고려
User	Profile	모든 회원의 최신 인바디정보 조회 기능	완료	2024년 5월 12일		user_id 기준 Distinct
User	Point	회원의 포인트정보 등록, 업데이트, 삭제 기능	완료	2024년 5월 9일		구현X
User	Point	모든 회원의 포인트 조회, 개별 유저 조회 기능	완료	2024년 5월 9일		구현X
Util	UploadFile	파일 업로드, 삭제, 조회 기능	완료	2024년 5월 10일		
Calendar	Schedule	목표 스케쥴 생성, 삭제 기능	완료	2024년 5월 10일		
Calendar	Routine	루틴 생성, 수정, 삭제, 조회 기능	완료	2024년 5월 10일		
Calendar	Routine	운동 루틴 추천 기능	완료	2024년 5월 10일		ChatGPT api 사용
BoardFree	Comment	대댓글 작성, 수정, 삭제 기능	완료	2024년 5월 13일		구현X
Util	Ranking	랭킹보드 생성 및 삭제 기능	완료	2024년 5월 11일		UserRank정보 사용
BoardFree	LikeCon	좋아요 기능	완료	2024년 5월 11일		구현X
Fitty	Fitty	파티 생성, 삭제, 조회, 매칭 기능	완료	2024년 5월 13일		매칭 알고리즘 (피어슨 상관계수 / 유클리드 거리 유사도)
Calendar	Detail	운동기록 생성, 조회, 수정, 삭제 기능	완료	2024년 5월 12일		
BoardFree	Comment	댓글 작성, 수정, 삭제 기능	완료	2024년 5월 13일		구현X
BoardFree	Board	게시글 작성, 수정, 삭제, 조회, 검색 기능	완료	2024년 5월 13일		구현X
Fitty	Board	파티모집 게시글 작성, 수정, 삭제, 조회, 검색 기능	완료	2024년 5월 12일		검색만하면됨
Fitty	Fitty	유효성 검사	완료			GPT 사용
AI	OCR	인바디정보 추출	진행 중			네이버클라우드 클로버 ocr
Calendar	Fitty	같은 파티원의 정보조회	시작 전			

핵심

주요 기능소개

Record

- 출석체크 및 정보 기록
- 스케줄관리 및 루틴 추천

Ranking

- 시즌별 티어제도
- 팀 내 경쟁 및 기록 포인트

Fitty

- 파티 맺기
- 팀원 정보 조회

Auto Matching

- Fitty 자동매칭
- 같은 스케줄 공유

Inbody

- 인바디 등록
- TimeLine

Routine

- 루틴 추천
- 운동 부위 추천

Counting sort

핵심 알고리즘

getMinOccurenceValue()

- 유저의 모든 운동기록을 수집
- 부위별 운동 횟수를 카운팅
- 가장 적게 등장한 운동부위를 추천

```
// 각 값의 등장 횟수를 세어서 가장 적게 등장한 값을 반환
const getMinOccurrenceValue = (arr) => {
  if (!arr || arr.length === 0) {
    console.warn("fitPartArray가 비어 있습니다.");
    return null;
  }

  const counts = {
    "이두+등": 0,
    "삼두+가슴": 0,
    "어깨+하체": 0,
    "하체+코어": 0,
  };

  arr.forEach((value) => {
    if (counts[value] !== undefined) {
      counts[value]++;
    }
  });

  let minCount = Infinity;
  let minValue = null;
  for (const [key, value] of Object.entries(counts)) {
    if (value < minCount) {
      minCount = value;
      minValue = key;
    }
  }

  return minValue;
};
```

Pearson Correlation Coefficient

핵심 알고리즘

pearsonCal()

- 파티에 가입하지 않은 모든 유저의 최신 인바디 정보를 수집
- Z – Score 정규화를 통해 평균이 0, 표준편차가 1인 데이터로 변형
- Min-Max 스케일링 정규화를 통해 데이터의 범위를 0과 1사이로 축소
- 사용자와 데이터리스트 간의 유사도를 비교하여 상관계수를 계산하고 리스트를 반환

```java
// Z-Score 정규화
public static double[] zScoreNormalization(double[] data) {
    double mean = calculateMean(data);
    double stdDev = calculateStdDev(data, mean);

    double[] normalizedData = new double[data.length];
    for (int i = 0; i < data.length; i++) {
        normalizedData[i] = (data[i] - mean) / stdDev;
    }

    return normalizedData;
}

// Min-Max 스케일링
public static double[] minMaxScaling(double[] data) {
    double min = findMin(data);
    double max = findMax(data);

    double[] normalizedData = new double[data.length];
    for (int i = 0; i < data.length; i++) {
        normalizedData[i] = (data[i] - min) / (max - min);
    }

    return normalizedData;
}

// 피어슨 상관계수 계산
public static double pearsonCal(double[] x, double[] y) {
    if (x.length != y.length) {
        throw new IllegalArgumentException("두 배열의 길이가 일치하지 않습니다.");
    }

    double meanX = calculateMean(x);
    double meanY = calculateMean(y);

    double numerator = 0.0, denominatorX = 0.0, denominatorY = 0.0;
    for (int i = 0; i < x.length; i++) {
        numerator += (x[i] - meanX) * (y[i] - meanY);
        denominatorX += Math.pow(x[i] - meanX, 2);
        denominatorY += Math.pow(y[i] - meanY, 2);
    }

    double denominator = Math.sqrt(denominatorX * denominatorY);

    if (denominator == 0) {
        return 0; // 분모가 0이면 상관관계가 없다고 가정
    }

    return numerator / denominator;
}
```

Euclidean Distance

핵심 알고리즘

euclideanDistance()

- 앞서 받은 데이터를 내림차 정렬하고 상위 10%의 데이터 추출
- 해당 데이터와 사용자 간의 유클리드 거리를 계산
- 유클리드 거리 값으로 오름차 정렬
- 상위 5명(최대) 추출 후 반환

```java
//두 데이터의 유클리드거리 반환
public static double euclideanDistance(double[] point1, double[] point2) {

    double sum = 0.0;
    for (int i = 0; i < point1.length; i++) {
        sum += Math.pow(point2[i] - point1[i], 2);
    }
    return Math.sqrt(sum);
}

//유사사용자 찾기
public List<UserProfile> calculateSimilar(List<UserProfile> list, UserProfile user) {

    // 사용자의 데이터 추출
    double[] userValues = { user.getMuscle(), user.getFat(), user.getBmi(), user.getFatPer(),user.getWeight()};
    List<double[]>pearson = new ArrayList<double[]>();
    List<double[]>euclide = new ArrayList<double[]>();
    //피어슨 상관계수를 구해서 pearson에 { 유저번호 , 상관계수값} 으로 add
    for (int i=0;i<list.size();i++) {
        UserProfile profile = list.get(i);
        if(!user.getGender().equals(profile.getGender())||user.getUserId().equals(profile.getUserId()))continue;
        double[] profileValues = { profile.getMuscle(), profile.getFat(), profile.getBmi(), profile.getFatPer(),profile.getWeight()};
    //    System.out.println(Arrays.toString(profileValues));
        double[] normalizedUser1Data = zScoreNormalization(userValues);
        double[] normalizedUser2Data = zScoreNormalization(profileValues);
        double correlation = pearsonCal(normalizedUser1Data, normalizedUser2Data);
        double [] tmp ={i, correlation};
        pearson.add(tmp);
    }
    //피어슨 상관계수 유사도로 정렬(내림차)
    Collections.sort(pearson, (o1, o2) -> Double.compare(o2[1], o1[1]));

    //상위 10% 가르기
    int tenPer = list.size()/10;

    //유클리드 거리 계산
    for(int i=0;i<tenPer;i++) {
        UserProfile profile = list.get((int)pearson.get(i)[0]);
        double[] profileValues = { profile.getMuscle(), profile.getFat(), profile.getBmi(), profile.getFatPer(),profile.getWeight()};
        double dif = euclideanDistance(profileValues, userValues);
        double [] tmp ={(int)pearson.get(i)[0], dif};
        euclide.add(tmp);
    }
    //유클리드 거리 유사도로 정렬(오름차)
    Collections.sort(euclide, (o1, o2) -> Double.compare(o1[1], o2[1]));

    //5명 반환 또는 사이즈만큼 반환
    List<UserProfile> result = new ArrayList<UserProfile>();
    for(int i=0;i<5 && i<euclide.size();i++) {
        int num = (int)euclide.get(i)[0];
        result.add(list.get(num));
    }
    return result;
}
```

Random Data Generator

테스트 환경

User Data

- 2000명
- Male / Female 각 1000명

UserRank Data

- 2000개
- 모든 값 랜덤적용(최소 최대 기준치 다름)

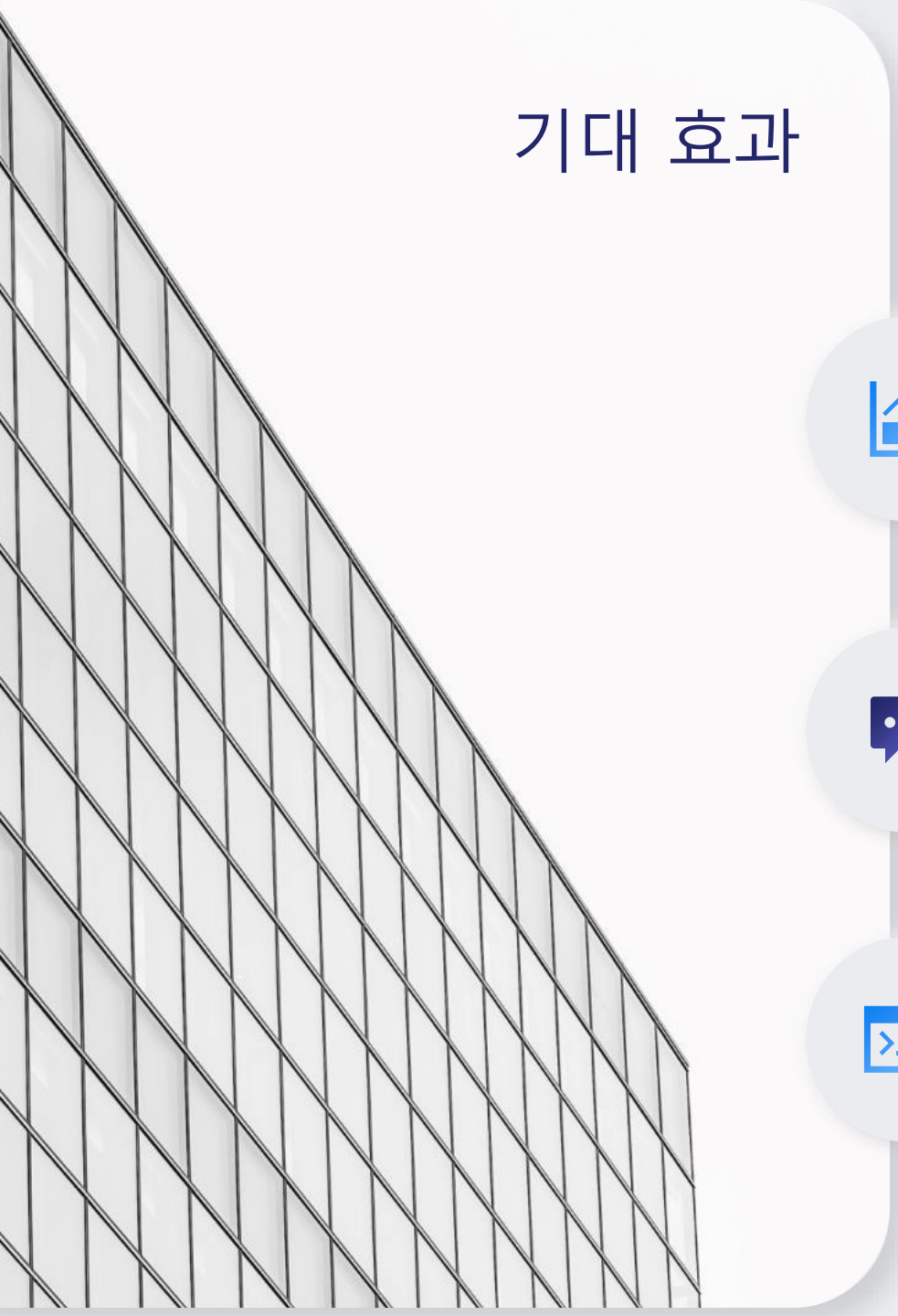
UserProfile Data

- 키 생성 – 키에 맞는 몸무게 생성 – 무게에 맞는 타 수치 생성

Fitty

- 1번~200번 유저 가입
- 무작위 1명 admin(20팀)
- 무작위 member/guest

```
// 근육량(muscle)과 체지방량(fat) 생성
public static double[] generateMuscleFat(String gender) {
    double muscle;
    double fat;
    double muscleStart;
    double fatStart;
    if (gender.equalsIgnoreCase("male")) {
        muscleStart = 20; // 남성 근육량 시작값 (kg)
        fatStart = 5; // 남성 체지방량 시작값 (kg)
    } else { // 여성인 경우
        muscleStart = 15; // 여성 근육량 시작값 (kg)
        fatStart = 5; // 여성 체지방량 시작값 (kg)
    }

    // 근육량과 체지방량에 대한 통합 백분율 생성 (0% ~ 100%)
    double totalPercentage = random.nextDouble();

    // 백분율에 따라 최종 값 생성 (근육량과 체지방량 각각의 최대값을 초과하지 않도록 조정)
    double totalWeight = 20;
    muscle = muscleStart + totalPercentage * totalWeight;
    fat = fatStart + totalPercentage * totalWeight;

    // 근육량에 대한 추가적인 랜덤한 퍼센트 적용 (-20% ~ 20%)
    double muscleVariation = random.nextDouble() * 0.4 - 0.2; // -0.2에서 0.2 사이의 값
    muscle *= (1 + muscleVariation);

    // 체지방량에 대한 추가적인 랜덤한 퍼센트 적용 (-20% ~ 20%)
    double fatVariation = random.nextDouble() * 0.4 - 0.2; // -0.2에서 0.2 사이의 값
    fat *= (1 + fatVariation);

    return new double[]{muscle, fat};
}

// 체지방율(fat_per) 생성
public static double generateFatPer(double weight, double fat) {
    double fatPer = fat / weight * 100; // 체지방율 계산
    return fatPer;
}

// 체질량지수(BMI) 생성
public static double generateBMI(double muscle, double fat, double height) {
    double weight = generateWeight(muscle, fat);
    double bmi = weight / (height * height);
    return bmi;
}

// 몸무게(weight) 생성
public static double generateWeight(double muscle, double fat) {
    // 근육량과 체지방량에 대한 총합
    double totalWeight = muscle + fat;

    // 근육량과 체지방량의 총합에 -30%부터 30% 사이의 랜덤한 값을 더함
    double variationPercentage = random.nextDouble() * 0.4 - 0.2; // -0.3에서 0.3 사이의 값
    double adjustedTotalWeight = totalWeight * (1 + variationPercentage);
    double result = adjustedTotalWeight+(totalWeight/2);
    if(result <45) {
        result = result/10 + 45;
    }
public static double generateHeight(double weight, String gender) {
    double minHeight;
    if (gender.equalsIgnoreCase("male")) {
        minHeight = 155; // 남성 최소 키 (cm)
    } else {
        minHeight = 150; // 여성 최소 키 (cm)
    }
```



기대 효과

부족한 시장구조 개선

운동 파트너를 찾고 함께하는 것에 대한 어려움을 개선
모르는 사람과도 손쉽게 파티를 맺어 운동 가능
손쉬운 매칭으로 운동 파티 문화 선도 가능

경쟁 구조로 동기 부여

개인이 스스로 부여하기 힘든 동기부여를 파티원 및 등록된
사용자들과의 경쟁을 통해 추가적인 목표 달성이 가능

개인 목표 달성

인바디 정보를 활용한 개인 목표 달성 수치를 시각적으로 손쉽게
확인하여 운동의 지속적인 동기부여를 제공하며, 인바디 사진 등록으로
변화를 더욱 직관적으로 추적 가능

BUSINESS REPORT

개발 후기

프로젝트 진행 후기

성민기

프로젝트를 통해 웹 서버 및 화면 구성, 데이터 로드 등에 대해 많이 배웠습니다. 하지만 배운 것을 적용하며 실습하는 과정에서 많은 부족함을 느껴 추가적인 공부가 필요하다고 느꼈습니다.

윤여준

기획했던 것들을 구현하면서 설계의 중요성을 뼈저리게 느꼈고, 앞으로는 구현하기 전에 튼튼하게 다져야겠다고 생각했습니다. 2학기 프로젝트에 진입한다면 시간과 능력의 한계로 기획 단계에서 드랍했던 아이디어들을 구현하고, 더 많은 API와 독창적인 알고리즘을 직접 구현해보고 싶습니다.