

PJT명	냠냠코치 Project	
단계	[Web(Back) PJT]	
진행일자	2025.09.26	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

- Servlet, JSP를 이해하고, Tomcat 환경에서 요청/응답 흐름을 제어하는 웹 서버를 구축할 수 있다.
- MVC 아키텍처를 이해하고 활용할 수 있다.
- Front Controller → Controller → Service → Repository의 계층을 설계하고, 책임 분리와 재사용성을 고려하여 코드를 작성할 수 있다.
- JSP EL·JSTL을 활용하여 코드를 작성할 수 있다.
- 세션·필터·에러를 처리하도록 구현할 수 있다.
- Front-End 관통 프로젝트를 이용하여 JSP에 활용할 수 있다.

2. 준비사항

1) 사용 데이터

- 음식 DB (식품의약품안전처 제공)

(<https://various.foodsafetykorea.go.kr/nutrient/general/down/historyList.do>)

식품코드	식품명	식품대분류명	영양성분활용기준량(%)[kcal]	수분(g)	단백질(g)	지방(g)	이불(g)	탄수화물(g)	당류(g)	비이성유(g)	질습(g)	질(mg)	인(mg)	칼륨(mg)	나트륨(mg)	Ajug	마테리얼(g)	카로틴(g)	
D101-004160000-0001	국밥_돼지머리	밥류	100g	137	6.7	5.16	0.63	15.94	0.16	0.7	9	0.24	24	47	126	5	0	61	
D101-004310000-0001	국밥_순대국밥	밥류	100g	75	3.17	2.28	0.48	10.38	0.17	1.3	9	0.67	31	34	126	5	0	55	
D101-004500000-0001	국밥_콩나물	밥류	100g	52	86.8	1.45	0.24	0.56	10.93	0	1.2	6	0.18	16	42	172	2	0	22
D101-006000000-0001	기방밥	밥류	100g	166	59.1	2.44	0.57	0.15	36.77	0	1.5	4	0.33	28	25	1	0	0	0
D101-007000000-0001	김밥	밥류	100g	140	69.5	4.84	4.55	1.11	19.98	0	1.4	36	0.3	55	118	307	113	3	1318
D101-007070000-0001	김밥_김치	밥류	100g	130	71.2	4.3	4.03	1.29	19.17	0.29	1.8	44	0.31	64	126	349	41	8	392
D101-007090000-0001	김밥_날치말	밥류	100g	177	59.6	6.1	4.26	1.13	28.66	1.43	2.1	20	0.62	82	116	299	61	15	559
D101-007120000-0001	김밥_토치즈	밥류	100g	202	55.8	5.77	5.81	0.95	31.64	1.33	2.1	15	0.49	74	134	241	22	0	259
D101-007280000-0001	김밥_소고기	밥류	100g	179	61.2	6.46	5.56	1	25.78	0.03	1.5	30	0.87	64	96	267	35	9	318
D101-007450000-0001	김밥_장치	밥류	100g	174	64.2	7	7.22	1.32	20.26	0.71	1.8	51	0.72	73	123	335	66	10	680
D101-007480000-0001	김밥_제소	밥류	100g	158	63.8	4.6	3.65	1.28	26.65	0.14	3.4	38	0.44	69	164	309	160	5	1851
D101-007490000-0001	김밥_치즈	밥류	100g	177	63.3	6.24	7.03	1.33	22.1	0.21	2.6	6	0.21	15	41	169	58	13	532
D101-007520000-0001	김밥_포기주	밥류	100g	169	61.9	4.84	4.41	1.32	27.52	0.09	3.3	19	0.49	70	148	327	104	10	1117
D101-010080000-0001	덮밥_낙지	밥류	100g	150	65.9	5.88	3.34	0.8	24.12	2.16	2.3	15	0.36	47	95	212	9	0	105
D101-010110000-0001	덮밥_닭고기	밥류	100g	125	70.8	11.4	2.18	0.76	14.82	1.07	1.1	21	0.53	108	137	153	63	50	167
D101-010150000-0001	덮밥_돼지고기/례	밥류	100g	202	62	9.43	10.77	0.95	16.86	1.98	1.4	11	0.49	101	196	174	31	3	343
D101-010240000-0001	덮밥_불고기	밥류	100g	182	60.2	6.6	5.31	0.97	26.96	0.07	2.8	6	0.82	73	138	253	1	0	7
D101-010390000-0001	덮밥_오징어	밥류	100g	135	68.1	7.18	2.01	0.81	21.94	0.66	2.6	8	0.22	78	135	162	46	1	539
D101-016000000-0001	보리밥	밥류	100g	161	59.9	2.9	0.24	0.24	36.77	2.57	2.4	7	0.2	34	28	4	0	0	0
D101-017000000-0001	볶음밥	밥류	100g	183	56.9	5.56	2.76	0.84	33.97	0	1.7	13	0.52	61	99	212	40	10	356
D101-017030000-0001	볶음밥_개란	밥류	100g	225	56.6	6.62	11.28	1.25	24.23	0.17	1.8	18	0.44	94	159	318	47	16	378
D101-017070000-0001	볶음밥_김치	밥류	100g	167	68.1	5.32	9.18	1.62	15.74	13.89	2.8	30	0.7	72	168	470	25	13	141
D101-017260000-0001	볶음밥_새우	밥류	100g	172	64.1	6.31	6.14	0.61	22.85	0.71	1.6	12	0.35	62	65	146	38	13	304
D101-017280000-0001	볶음밥_소고기	밥류	100g	176	63.2	7.59	6.48	0.92	21.76	0.08	2.1	18	0.62	86	167	215	31	18	151
D101-017450000-0001	볶음밥_장치	밥류	100g	180	57.4	4.62	3.27	1.83	32.9	0.12	1.6	15	0.11	46	66	631	7	0	90
D101-017480000-0001	볶음밥_제소	밥류	100g	182	59.7	4.72	5.39	1.44	28.74	0.29	2.3	17	0.42	61	156	408	32	15	210
D101-017510000-0001	볶음밥_표고버섯	밥류	100g	184	58.9	3.36	4.81	1.12	31.79	0	2.6	5	0.3	39	167	299	0	0	0
D101-018000000-0001	비빔밥	밥류	100g	142	68.9	6.86	4.32	1.11	18.84	0.05	2.0	31	1.04	79	180	232	73	9	771
D101-018130000-0001	비빔밥_돌솥	밥류	100g	147	65.7	5.39	2.95	1.25	24.68	1.24	2.4	25	0.59	72	114	337	56	8	582
D101-018430000-0001	비빔밥_육회	밥류	100g	147	65.7	6.39	3.36	1.6	22.92	4.45	2.6	28	0.89	95	178	446	36	0	427
D101-019460000-0001	김밥_침치미요	밥류	100g	199	59.6	5.93	8.21	0.81	25.45	0	2.8	10	0.38	58	76	230	23	1	269
D101-021000000-0001	수수밥	밥류	100g	161	60.8	3.16	0.92	0.15	35.02	0.08	1.1	4	0.17	25	26	1	0	0	0
D101-023000000-0001	알밥	밥류	100g	128	72.8	4.73	5.11	1.59	15.75	2.3	3.5	43	0.44	61	128	430	39	0	467
D101-026000000-0001	영양동чин밥	밥류	100g	184	57.4	3.43	2.87	0.21	36.06	0.63	2.1	9	0.48	32	36	6	10	0	124

● 2023 국민건강 통계 (질병관리청 제공)

(https://knhanes.kdca.go.kr/knhanes/archive/wsiStatsClct.do;jsessionid=jaYoRyHMvc31xWBxtA9zAGVSy237N_DwOYKdJKv2.knhanes_20)

00. 2023국민건강통계 표목록.xlsx	12. 비만.xlsx
01. 흡연.xlsx	13. 고혈압.xlsx
02. 음주.xlsx	14. 당뇨병.xlsx
03. 신체활동.xlsx	15. 이상지질증증.xlsx
04. 정신건강.xlsx	16. 간염.xlsx
05. 손상 및 안전의식.xlsx	17. 빈혈.xlsx
06. 건강검진 및 예방접종.xlsx	18. 만성콩팥병.xlsx
07. 의료이용.xlsx	19. 구강건강.xlsx
08. 삶의 질.xlsx	20. 이비인후질환.xlsx
09-1. 식품 섭취.xlsx	21. 굴절이상.xlsx
09-2. 영양소 섭취.xlsx	22. 악력.xlsx
10. 영양 부족 및 과잉.xlsx	23. 기타.xlsx
11. 식생 활행태.xlsx	24. 국내외지표자료(OECD,WHO,HP2030).xlsx
12. 비만.xlsx	

표번호	표14-3
지표명	당뇨병 유병률(공복혈당 또는 당화혈색소 기준)
지표정의	1) 당뇨병 유병자가 아니면서 공복혈당이 100-125mg/dL 또는 당화혈색소가 5.7% 이상 6.4% 이하인 분율 2) 공복혈당이 126mg/dL 이상이거나 의사진단을 받았거나 혈당강하제복용 또는 인슐린 주사를 사용하고 있거나, 당화혈색소 6.5% 이상인 분율
분자	1) 당뇨병 유병자가 아니면서 공복혈당이 100-125mg/dL 또는 당화혈색소가 5.7% 이상 6.4% 이하인 사람 수 2) 공복혈당이 126mg/dL 이상이거나 의사진단을 받았거나 혈당강하제복용 또는 인슐린 주사를 사용하고 있거나, 당화혈색소 6.5% 이상인 사람 수
분모	만19세이상 분석대상자 수
비고	3) 소득수준 : 월가구균등화소득(월가구소득 / √가구원수)을 성별·연령별(5세단위) 오분위로 분류 4) 2005년 추계인구로 연령표준화 ※ 변동계수(coefficient of variation) : * 25-50% ** 50% 이상

구 분	전체				남자			
	n	정상	당뇨병전단계 ¹⁾	당뇨병 ²⁾	n	정상	당뇨병전단계	당뇨병
19세이상	5,504	57.8 (0.9)	29.1 (0.7)	13.2 (0.6)	2,396	52.8 (1.1)	31.7 (1.0)	15.5 (0.8)
30세이상	4,922	51.6 (0.9)	33.0 (0.7)	15.3 (0.7)	2,123	46.0 (1.2)	35.9 (1.1)	18.1 (1.0)
65세이상	1,679	32.7 (1.2)	40.1 (1.3)	27.2 (1.2)	746	30.8 (2.0)	38.8 (2.1)	30.3 (1.8)
연령(세)								
19-29	582	89.9 (1.5)	8.5 (1.4)	1.7 (0.6)*	273	86.3 (2.4)	10.9 (2.2)	2.7 (1.0)*
30-39	627	77.6 (2.0)	19.9 (1.9)	2.5 (0.6)*	288	70.7 (2.8)	26.1 (2.6)	3.2 (1.0)*
40-49	938	64.4 (1.6)	27.9 (1.5)	7.7 (1.0)	404	54.5 (2.4)	34.2 (2.3)	11.3 (1.8)
50-59	1,032	45.5 (1.8)	37.6 (1.7)	16.9 (1.3)	405	35.3 (2.7)	42.8 (2.7)	21.9 (2.3)
60-69	1,253	37.1 (1.5)	38.7 (1.4)	24.2 (1.3)	550	34.4 (2.3)	36.3 (2.3)	29.3 (2.0)
70+	1,072	31.8 (1.5)	41.1 (1.5)	27.1 (1.5)	476	30.4 (2.4)	41.1 (2.4)	28.4 (2.3)

2) 개발언어 및 툴

- Java, STS, HTML, CSS, JavaScript, VScode, Tomcat

3) 필수 라이브러리 / 오픈소스

- BootStrap Framework

3. 작업 순서

- 1) 팀원과 같이 요구사항(기본/추가/심화)을 검토하고, 역할을 분배한다.
- 2) MVC 계층(Controller → Service → Repository)을 배치한 아키텍처·클래스 디자인 초안을 작성한다.
- 3) 생성형 AI를 활용하여 디자인의 개선 포인트를 정리한다.
- 4) 설계한 프로젝트를 구현 개발한다.
- 5) 기능을 테스트하고 디버깅 한다.

4. 요구사항

식단 분석을 활용한 개인 맞춤형 건강 코칭 서비스(이하 "냠냠코치"라 한다.)를 구현하려고 한다. 냠냠코치에서는 식품의약품안전처에서 제공한 음식 별 영양성분 데이터를 활용하여 작성한 식단을 사용자가 원하는 목표(운동, 질병 등)에 따라 분석하여 정보를 제공한다. 또한 건강 목표를 위한 챌린지, 식단 리뷰 및 다양한 정보를 교류할 수 있는 커뮤니티 등 다양한 기능을 지원한다. 팀원과 상의 후 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

이번 관통 프로젝트는 '냠냠코치' 프로젝트의 Back-End 부분을 구현하고, Front-End 프로젝트의 화면을 JSP를 이용하여 구현한다.

아래의 참고용 요구사항 예시를 활용하여 보다 개선된 프로젝트의 요구사항을 정리하고 분석하여 요구사항 기능 명세서를 작성해보자. (아래 필수 기능 포함)

● 요구사항 예시(참고용)

- 시장 조사를 통하여 '냠냠코치' 프로젝트의 요구사항을 완성해보자
- 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, 필수 기능은 구현해야 한다.
(ex. 컬럼의 내용을 추가할 수도 있음, 요청 URL 등)

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	식단	식단 작성	섭취한 음식을 DB에서 선택하여 식단기록을 작성하는 기능	필수
F02	식단	식단 조회	식단 ID를 활용하여 식단기록 내용을 상세하게 조회하는 기능	필수
F03	식단	식단 수정	작성한 식단기록의 내용을 수정하는 기능	필수
F04	식단	식단 삭제	작성한 식단기록을 삭제하는 기능	필수
F05	식단	식단 분석	입력한 식단 기록을 통해 (공식 or AI) 영양 정보를 분석하여 표현하는 기능	필수
F06	회원	회원 작성	회원 가입을 통해 User의 정보를 등록하는 기능 (개인 프로필 – 키, 몸무게, 질환 등을 관리)	필수
F07	회원	회원 조회	회원 정보를 조회하는 기능	필수
F08	회원	회원 수정	회원 정보를 수정하는 기능	필수
F09	회원	회원 삭제	회원 정보를 삭제하는 기능 (삭제 대신 '비활성화'로 상태를 바꿀 수도 있음)	필수
F10	회원	로그인/로그아웃	로그인/로그아웃 기능	필수
F11	회원	팔로우/팔로잉	팔로우 추가, 취소, 목록 조회할 수 있는 기능	추가
F12	챌린지	챌린지 정보관리	챌린지 정보 CRD (기간, 설명, 이미지 등 관리)	추가
F13	챌린지	챌린지	사용자의 챌린지 참여, 달성을 등 정보	추가

		사용자참여	관리	
F14	커뮤니티	게시판	식단 리뷰, 전문가, 자유 게시판 글 작성, 조회, 수정, 삭제 기능	추가
F15	커뮤니티	댓글	게시글에 댓글 작성, 조회, 수정, 삭제 기능	추가
F16	AI	AI 식단 분석	생성형 AI를 활용한 식단 분석 기능	심화
F17	AI	AI 운동 코칭	생성형 AI를 활용한 운동 코칭 기능	심화
...

비 기능적 요구사항

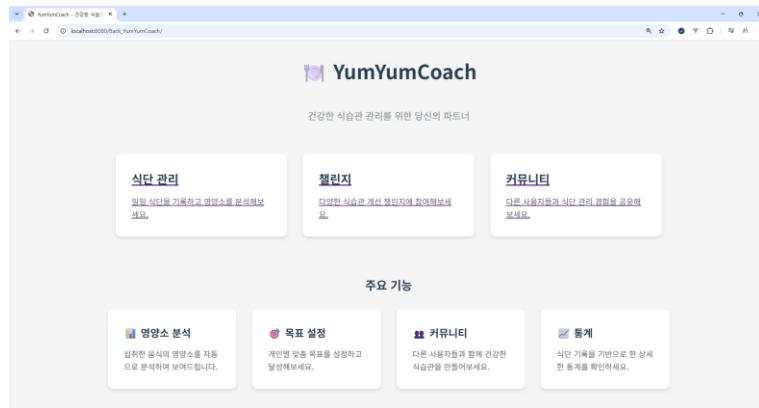
NF01	UX	사용자 편의성	서비스에 대한 사전지식이 없어도 사용하기 편해야 함.	-
NF02	UI	가용성	언제나 (어떤 디바이스로든) 서비스 가능해야 함.	-
NF03	효율성	응답성	조회에 대한 결과를 빠르게 응답해야 함.	-
NF04	효율성	정확성	신뢰성 있는 데이터를 사용하여 분석 결과 등의 내용이 정확해야 함.	-
...

1) 기본(필수) 기능

'냠냠코치' 프로젝트의 식단기록 관리 및 회원 정보 관리 기능을 구현하여 보자. 사용자는 식품의약품안전처에서 제공한 음식 DB(영양분 정보)를 활용하여 섭취한 식단기록을 관리할 수 있고, 주어진 통계자료를 활용하여 목표(건강, 질병, 다이어트 등)에 맞게 식단을 분석하여 점수를 산출할 수 있다.

● 식단기록 관리

- 식단 기록 관리 기능(작성, 조회, 수정, 삭제)을 위한 MVC 아키텍처 설계 및 코드 작성 (Controller/Service/Dao/DTO/JSP)



(화면 예시)

```
@WebServlet("/diet/*")
public class DietController extends HttpServlet {
    private DietService dietService = new DietService();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String pathInfo = request.getPathInfo();

        // 1) 목록 조회
        if (pathInfo == null || pathInfo.equals("/") || pathInfo.equals("/list")) {
            List<Diet> diets = dietService.getAllDiets();
            request.setAttribute("diets", diets);
            request.getRequestDispatcher("/WEB-INF/views/diet/list.jsp")
                .forward(request, response);
        }
    }
}
```

(Controller 예시)

```
@WebServlet("/diet/*")
public class DietController extends HttpServlet {
    private DietService dietService = new DietService();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String pathInfo = request.getPathInfo();

        // 1) 목록 조회
        if (pathInfo == null || pathInfo.equals("/") || pathInfo.equals("/list")) {
            List<Diet> diets = dietService.getAllDiets();
            request.setAttribute("diets", diets);
            request.getRequestDispatcher("/WEB-INF/views/diet/list.jsp")
                .forward(request, response);
        }
    }
}
```

(Service 예시)

```
public interface DietLogDao {
    DietLog getDietLogById(int dietLogId);

    List<DietLog> getDietLogsByUserId(int userId);

    void insertDietLog(DietLog dietLog);

    void updateDietLog(DietLog dietLog);

    void deleteDietLog(int dietLogId);
}
```

(Dao 예시)

```

public class DietLog {
    private Integer dietLogId; // 식단 로그 ID
    private Integer userId; // 사용자 ID
    private String logDate; // 식단 기록 일시
    private Double totalCalorie; // 총 칼로리 (옵션)
    private String memo; // 메모/코멘트
    private String imageUrl; // 식단 이미지 URL
    private Date createdAt; // 생성 일시
    private Date updatedAt; // 수정 일시
    private List<DietLogItem> items; // 식단 상세 항목 리스트

    // Getter 및 Setter
    public Integer getDietLogId() {
        return dietLogId;
    }
}

```

(DTO 예시)

- 식단 분석을 통한 점수 산출 기능 작성
- 회원 정보 관리
 - 회원 정보 관리 기능(작성, 조회, 수정, 삭제)을 위한 MVC 아키텍처 설계 및 코드 작성 (Controller/Service/Dao/DTO/JSP)
 - 사용자 프로필 (키, 몸무게, 질환 등) 관리 기능 구현
 - 로그인/로그아웃 기능 구현

2) 추가 기능

위 필수 기능을 모두 구현했다면 추가 기능을 구현해보자

사용자는 건강 목표를 위한 게이미피케이션 요소를 활용한 챌린지 기능을 사용할 수 있고, 회원들 간의 커뮤니티를 위한 식단 리뷰, 전문가, 자유 게시판 기능을 활용할 수 있다.

- 챌린지 관리
 - 건강목표 챌린지 관리 기능(작성, 조회, 삭제)을 위한 MVC 아키텍처 설계 및 코드 작성 (Controller/Service/Dao/DTO/JSP)
 - 사용자 참여/달성을 관리 기능 구현
- 커뮤니티 관리
 - 식단기록 리뷰 관리 기능(작성, 조회, 수정, 삭제)을 위한 MVC 아키텍처 설계 및 코드 작성 (Controller/Service/Dao/DTO/JSP)
 - 전문가, 자유게시판 관리 기능을 위한 MVC 아키텍처 설계 및 코드 작성 (Controller/Service/Dao/DTO/JSP)

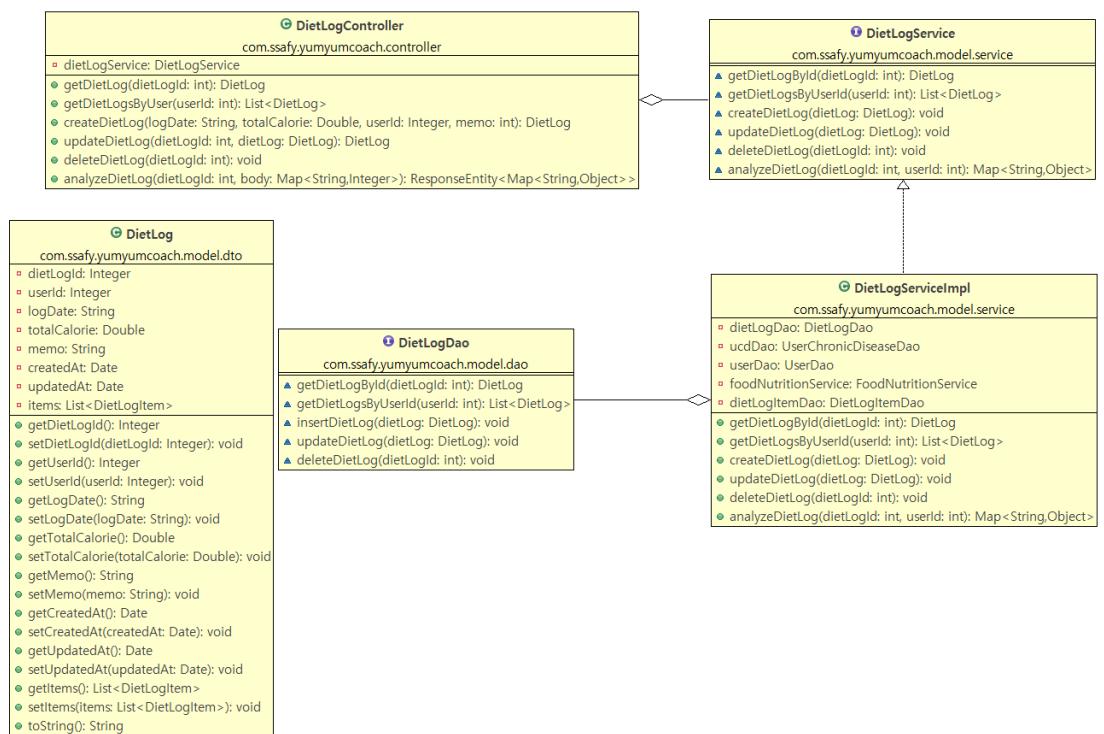
- 회원간 팔로우/팔로잉 관계 기능

3) 심화 기능

- 그 외 프로젝트의 다양한 기능을 위해서 타 사이트를 벤치마킹 하거나, 각 팀별 새로운 아이디어를 추가하여 프로젝트를 확장해보자
- 생성형 AI 활용 방안: OpenAI 등 생성형 AI를 활용할 수 있는 방법에 대해서 팀원과 이야기를 해보고, 추가할 수 있는 기능을 정리하여 실제 프로젝트에 적용 및 구현해보자.
- 적용할 심화 기능 항목을 선정하면, 요구사항 명세서(표)에 반영해보자

5. 참고자료

● 식단 기록 클래스 다이어그램 (예시)



6. 결과

프로젝트 최종적으로 제출해야 할 항목은 아래와 같고, GIT의 README.md파일이나 워드, 파워포인트 문서를 활용하여 작성하고, GitLab에 업로드 한다.

● 산출물과 제출

- README.md
- 구현 소스코드 및 실행 화면 캡처자료
- 냠냠코치_Back_지역_반_성명1_성명2.zip으로 제출

7. 채점 기준

난이도	구현 기능	점수	비고
기본	식단기록 관리 기능	30	화면, 등록, 조회, 수정, 삭제 기능
	회원정보 관리 기능	30	화면, 등록, 조회, 수정, 삭제 기능
추가	챌린지 기능	10	화면, 등록, 조회, 수정, 삭제 기능
	커뮤니티 기능	10	화면, 등록, 조회, 수정, 삭제 기능
심화	생성형 AI 적용 기능/아이디어	10	
	팀별 아이디어	10	타 사이트 벤치마킹 또는 새로운 아이디어 구현