

# 2025-1학기 학습플러스 코칭 동아리 보고서

## 동아리 정보

동아리 명	너의 수강은	활동 차시	5주차
이번 차시 활동 요약 (학습목표, 달성도 등)	4주차까지 제작한 웹사이트 프로토타입을 피드백을 반영해 개선함. 남은 학점을 기준으로 필요한 과목을 추천하는 알고리즘 개발 중임.(달성도: 80%). 2. 시간표 내 강의실 위치를 고려해 이동 시간이 최소화되도록 조합을 추천하는 거 리 최적화 알고리즘 개발중임.(달성도: 80%)		
모임 일시 및 진행장소	2025년 5월 29일(목) 19:00~21:00		
	대면 미팅(부산대학교 10공학관 5층)		

## 구성원 명단 및 참석여부

연번	구분	이름	참석 여부(○/×)	불참 시 작성
1	리더	황연경	○	불참사유:
2	팀원	안나연	○	불참사유:
3	팀원	이지환	○	불참사유:
4	팀원	정재원	○	불참사유:
5	팀원	임승우	○	불참사유:
6	팀원			불참사유:
7	팀원			불참사유:
8	팀원			불참사유:

본인은 위와 같이 학습동아리 활동 일지를 제출하며,  
위의 사항이 사실과 틀림없으며 잘못된 기재나 표기로 인해  
발생하는 불이익에 대해 책임질 것을 약속합니다.

2025년 5월 29일

리더

황연경

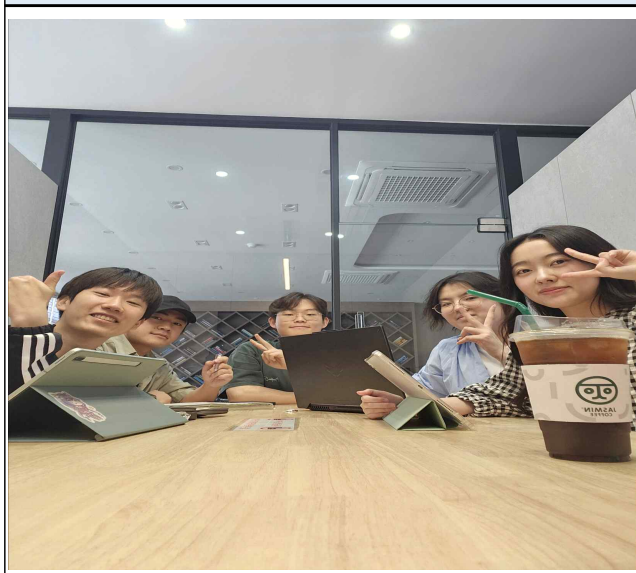
황연경

학습 주제 및 내용	
학습 목표	<p>1. 남은 학점을 기준으로 필요한 과목을 추천하는 알고리즘 개발.</p> <p>2. 시간표 내 강의실 위치를 고려해 이동 시간이 최소화되도록 조합을 추천하는 거리 최적화 알고리즘 개발.</p>
학습 내용	<p>1. 피드백을 반영한 프로토타입 개선</p> <p>4주차에 진행한 Google Form 피드백 분석 결과, 졸업 요건 자동 정리 기능의 필요성이 가장 높게 나타났음. 이에 따라 사용자 학번, 주전공, 복수전공 여부에 따라 이수 과목을 자동 분류하고, 졸업 요건 충족 여부를 시각적으로 보여주는 기능 개발에 집중함.</p> <p>시간표, 과목, 사용자, 졸업요건 등 핵심 테이블의 관계를 반영한 ERD를 설계를 마무리하고 하고, 이 구조를 기반으로 데이터베이스를 구축함.</p> <p>2. 남은 학점을 기준으로 필요한 과목을 추천하는 알고리즘 개발</p> <p>졸업 요건 분석을 기반으로 남은 학점을 계산하고, 이수 가능한 과목 중 우선순위를 설정하여 추천하는 알고리즘을 파이썬으로 개발 중임.</p> <p>※ 구현 방식</p> <p>① 사용자의 이수 과목 정보 → 졸업요건 매핑 → 남은 학점 도출</p> <p>② 과목 데이터베이스(과목코드, 학점, 개설학기, 선수과목)와 비교</p> <p>③ 필수 과목, 선수과목 만족, 수강 가능 학기 여부 등을 기준으로 우선순위 부여</p> <p>→ 사용자는 졸업에 필요한 과목을 직관적으로 파악하고, 계획적으로 수강 신청 가능.</p> <p>3. 시간표 내 강의실 위치를 고려해 이동 시간이 최소화되도록 거리 최적화 알고리즘 개발</p> <p>강의실 간의 실제 거리를 고려해 연속된 강의 간 이동 동선을 최소화하는 거리 기반 최적화 알고리즘을 파이썬으로 개발 중임.</p> <p>※ 구현 방식</p> <p>① 각 강의의 강의실 위치(건물/호실)를 좌표화하여 거리 행렬 생성</p> <p>② 시간표 상 인접 시간대 강의 간 이동 거리를 계산</p> <p>③ 시간표 후보 중 총 이동 거리 최소화되는 조합을 추천</p> <p>실제 캠퍼스 환경을 반영하여 학생의 이동 피로도 감소와 시간표 만족도 향상을 기대함.</p> <p>4. 주차별 산출물 및 다음 단계 준비</p>

			<p>5주차 산출물로는 ERD 설계 최종본 및 관련 테이블 스키마 정의서, 과목 추천 알고리즘 로직 초안 및 테스트용 샘플 데이터셋, 강의실 거리 기반 이동 최적화 알고리즘의 기본 구조가 있음.</p> <p>6주차에는 알고리즘 구축을 마무리하고, 캠퍼스맵과 연동하여 이동 시간을 자동 계산하고 시각화하는 기능을 추가적으로 구현할 예정임.</p>
자체 평가			<p>5주차에는 사용자 피드백을 바탕으로 요구 기능을 우선순위화하고, 그중 졸업 요건 자동 정리 기능에 집중함. ERD 설계 및 테이블 구조 정리가 마무리되면서 전체 시스템 구조에 대한 이해도 높아졌음. 각 기능 간 데이터 흐름을 명확히 정리할 수 있었음.</p> <p>과목 추천 알고리즘과 거리 최적화 알고리즘의 기초 구조를 설계하면서, 이론적으로만 접근했던 알고리즘 로직을 실제 서비스 흐름에 맞게 구체화하는 과정에서 많은 고민이 있었음. 특히 조건 분기, 데이터 필터링 우선순위 정립 과정에서 설계 미흡한 부분이 드러났음. 테스트를 통해 보완이 필요하다고 판단됨.</p> <p>전체적으로는 사용자의 실제 요구를 반영해 실질적인 기능 개선 방향을 잡았다는 점에서 의의가 있었음. 다만 설계에만 시간을 많이 쓰면서 일부 구현이 지연된 부분은 아쉬움으로 남음. 다음 주차에는 프로토타입 작동 여부를 중심으로 실제 데이터 테스트를 통한 검증과 기능 고도화가 필요함.</p>
학습 참고자료 출처			<p>1. Lin, Yao Tsung, Shian-Shyong Tseng, and Chi-Feng Tsai. "Design and implementation of new object-oriented rule base management system." Expert Systems with Applications 25.3 (2003): 369-385.</p> <p>2. Chi, Yu-Liang. "Ontology-based curriculum content sequencing system with semantic rules." Expert Systems with Applications 36.4 (2009): 7838-7847.</p>
활동 성찰내용			
연번	이름	성찰 내용	
1	황연경	<p>사용자 피드백을 바탕으로 기능의 우선순위를 재정립하면서 단순한 기능 구현보다 실제 사용자 요구를 반영하는 설계가 중요하다는 점을 체감함. 초기에는 전체 기능을 빠르게 구현하는 데 집중했으나, 사용자가 무엇을 가장 필요로 하는지 파악한 후에는 방향성이 명확해졌음.</p> <p>ERD 설계 단계에서 데이터 간 관계 정의가 미흡했던 점이 구현 과정에서 오류로 이어졌고, 결국 구조를 다시 정리해야 했음. 이후 기능 간 의존성을 고려한 설계의 중요성을 깨달음. 데이터 흐름과 테이블 구조를 먼저 명확히 정리하고 시작해야 개발 과정의 반복을 줄일 수 있다고 판단함.</p> <p>과목 추천 및 거리 최적화 알고리즘 설계 시, 조건이 다양하고 경우의 수가 많아 우선순위를 어떻게 설정할지에 대한 기준 마련이 어려웠음. 사용자 입장에서 실질적인 도움이 되는 우선순위 로직을 구성하려면 기능 단위가 아닌 사용 시나리오 중심 사고가 필요하다는 점을 알게 됨.</p>	
2	안나연	<p>사용자 피드백을 바탕으로 졸업 요건 자동 정리 기능에 집중하여 실질적인 서</p>	

		<p>비스 개선 방향을 설정할 수 있었음. 서비스하기 위해서 실제 사용자의 요구사항이 중요하다는 생각을 했었는데 설문 조사 결과를 좀 더 반영할 수 있어서 좋았음.</p> <p>과목 추천 알고리즘과 거리 최적화 알고리즘을 실제 서비스 환경에 맞게 구체화하는 과정에서 조건 분기와 우선순위 설정의 복잡성을 체감했음. 자료를 정리해보며 학생들이 지금 접하는 UI와 사이트 구조가 직관적이지 않다는 점을 다시 체감함. 기존 사이트도 이런 생각을 안 한 건 아니겠다는 생각이 들어 이론과 실제 구현 사이의 간극을 깨달음. 설계 단계에 많은 시간을 투자하면서 일부 구현이 지연된 점은 아쉬웠지만, 탄탄한 기반을 구축했다는 점에서 의미가 있었음.</p>
3	이지환	거리 최적화 알고리즘을 현실에 대입하는 과정에서 추가적으로 고려해야 할 여러 제약들을 파악하게 됨. 여러 조건들을 넣기도 하고 빼기도 하며 우리 상황에 맞는 알고리즘을 설계할 수 있었음. 하지만 아직 완벽하지 않고 개선할 여지가 많다고 판단하여 더욱 정진 할 예정임.
4	임승우	디자인과 개발의 협업 과정에서 사용자 중심의 시각이 중요하며, 문서에 담긴 피드백과 실험 결과를 바탕으로 설계를 개선하는 경험을 쌓았다. 앞으로는 문서에 기록된 사용자 요구와 경험을 더 깊이 반영하여, 프론트엔드 개발과 UX 설계를 조화롭게 연결하는 데 집중하겠다.
5	정재원	이번 주차 활동을 통해, 사용자의 페인 포인트를 명확하게 정의한 후 개발하는 것이 얼마나 중요한지 다시 한번 깨달았습니다. 프로그램의 완성도나 퀄리티를 높이는 것도 물론 중요하지만, 무엇보다 사용자의 요구 사항에 맞춰 개발하는 것이 진정으로 좋은 프로그램을 만드는 길이라는 생각을 하게 되었습니다.

### 활동사진(첨부)



```

과목추천알고리즘 ☆
파일 수정 보기 삽입 런타임 도구 도움말

검색 명령어 + 코드 + 텍스트

# 사용자 총 이수 학점 계산
def calculate_completed_credits(user_info, course_db):
    return sum(course_db[c]["credit"] for c in user_info["completed_courses"] if c in course_db)

# 과목 추천 로직
def recommend_courses(user_info, course_db, graduation_requirements):
    recommendations = []

    for course, info in course_db.items():
        if course in user_info["completed_courses"]:
            continue # 이미 수강한 과목 제외

        # 선수과목 체크
        prereq = info["prerequisite"]
        if prereq and prereq not in user_info["completed_courses"]:
            continue # 선수과목 미이수 시 추천 제외

        # 수강 가능 학기 체크
        if info["semester"] != user_info["current_semester"]:
            continue # 현재 학기에 개설되지 않으면 제외

        # 우선순위: 필수 과목 여부
        priority = 1 if course in graduation_requirements["required_courses"] else 2

        recommendations.append((priority, course))

    # 우선순위 오름차순 정렬
    recommendations.sort()

    return [course for _, course in recommendations]

# 실행
completed_credits = calculate_completed_credits(user_info, course_db)
remaining_credits = graduation_requirements["total_required_credits"] - completed_credits

```