

대학교 상담 관리 시스템(CMS)

TEAM NAME : IPv4

TEAM LEADER : 김선익

TEAM MEMBER : 구나영, 오민영, 임민섭

MENTOR : 한준성 멘토

Get Started



목 차

01. 프로젝트 개요
02. 프로젝트 팀 구성 및 역할
03. 프로젝트 수행 절차 및 방법
04. 화면 설계
05. 프로젝트 수행 경과
06. 트러블 슈팅
07. 자체 평가 의견



01. 프로젝트 개요



프로젝트 주제 및 선정 배경, 기획의도

관리자, 상담사, 내담자와 같이 여러 클라이언트가 존재하기 때문에 다양한 사용자 경험을 제공할 수 있으며, 기술적으로도 다양한 접근 방식을 실험할 수 있다고 생각하여 선정하게 됨. 세 클라이언트 그룹 간의 유기적인 상호작용(협력, 소통, 정보공유)을 통해 각자의 역할을 최적화하여 높은 사용자 경험을 얻도록 하고자 함.

프로젝트 내용

관리자, 상담사, 메인 페이지로 구성하기로 하였으며, 관리자 페이지에서는 상담 일정 관리, 상담사 관리 및 메인 페이지의 전반적인 관리를 수행할 수 있도록 구성하기로 함.

상담사 페이지는 상담사가 자신의 일정을 확인할 수 있는 캘린더, 상담 신청 내역, 상담 일지 작성 및 공지사항 게시판 등 상담 관련 기능으로 구성하기로 함. 관리자와 상담사 간의 소통이 가능한 채팅 기능도 제공하기로 함.

메인 페이지는 학생에게 상담에 대한 정보를 제공하고, 상담을 신청 할 수 있는 상담 신청 페이지와 자가 진단 페이지, 상담 후기, 공지사항, FAQ 등의 게시판, 마이페이지를 운영하기로 함.

또한, 사용자가 보다 편리하게 페이지를 이용할 수 있도록 챗봇 기능도 제공하기로 함.



활용 장비 및 재료

프론트 엔드는 Bootstrap, CSS, JS, ES6, HTML을 사용하여 개발하였음. 개발 환경은 STS4이고, 개발 언어는 JAVA 17이며, 빌드 도구로는 MAVEN을 사용함. 웹 프레임워크는 Spring Boot와 Spring MVC를 활용하였고, ORM 프레임워크로 Mybatis를 사용함. 보안은 Spring Security로 강화하였으며, 템플릿 엔진은 Thymeleaf를 사용함. 클라우드 및 배포는 Docker와 AWS EC2를 통해 진행하고, 웹 서버는 Nginx를 이용하여 구성함.

활용방안 및 기대효과

관리자와 상담사 간의 채팅을 통하여 상담사의 스케줄을 효율적으로 조정할 수 있으며, 관리자 페이지에서 상담 일정 관리 메뉴를 통해 상담 관리의 효율성을 높였음.

상담 통계와 상담 후기 게시판을 활용해 상담 프로그램의 효과에 대해 평가하고 개선할 수 있음.

또한, 챗봇 및 다양한 게시판을 통해 학생들에게 상담 정보를 쉽게 제공하여 상담에 대한 접근성을 향상시키는 기대 효과가 있음.



02. 프로젝트 팀 구성 및 역할



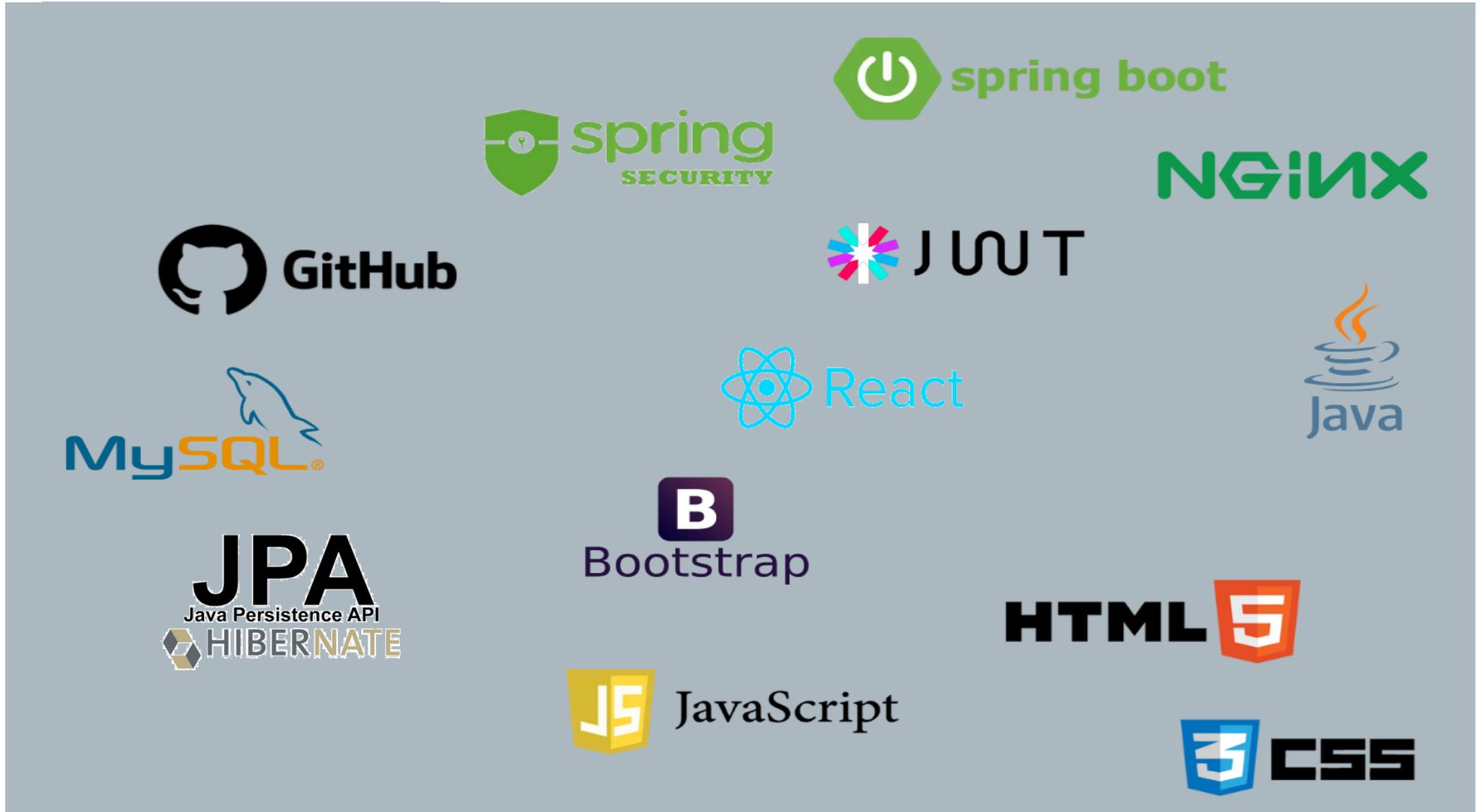
훈련생	역할	담당 업무
김선익		공지사항, 핵심역량
구나영		로그인, 마이페이지, 메인페이지
오민영		비교과, 마일리지
임민섭		상담

03. 프로젝트 수행 절차 및 방법

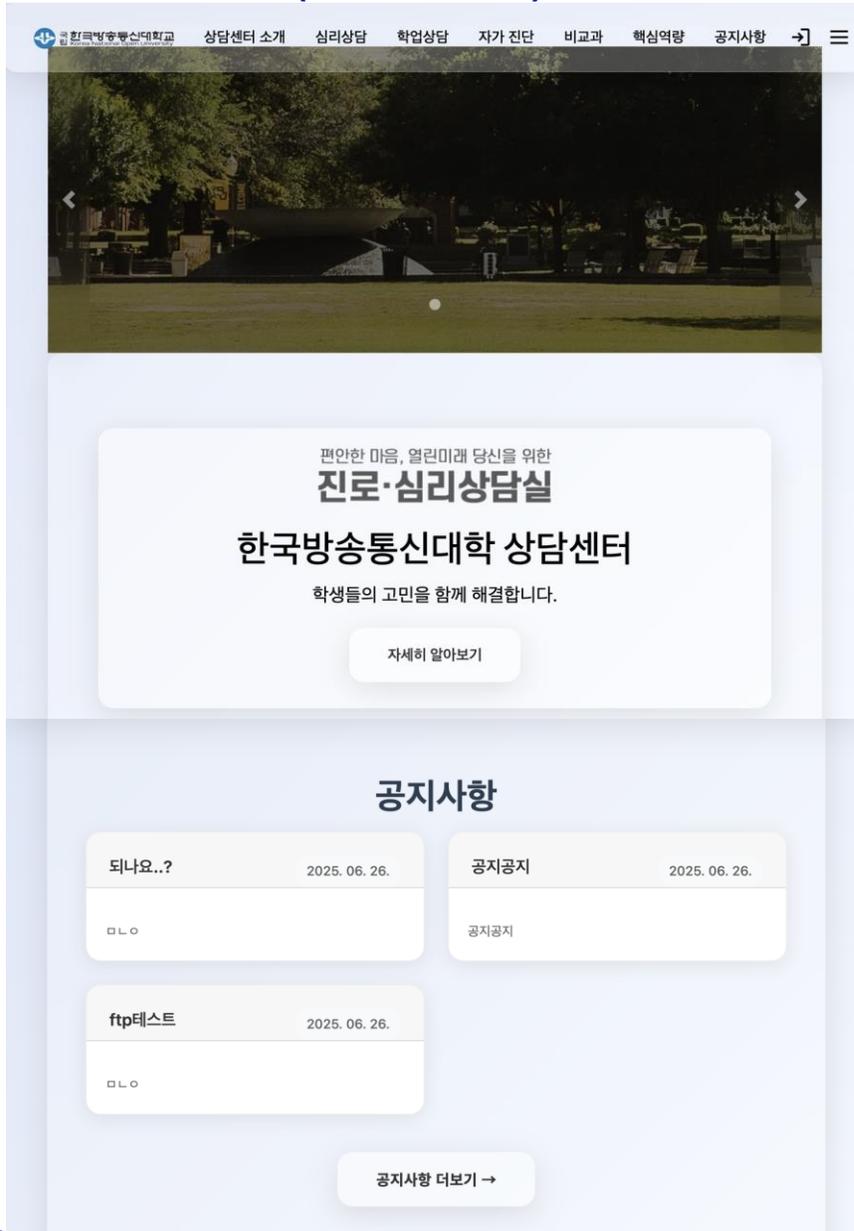
CMS
PROJECT

	6/2~6/6	6/9~6/13	6/16~6/20	6/23~6/30
사전 기획 및 요구 분석	사전 기획 및 요구 분석			
설계		설계		
구현		구현		
시험 및 피드백			테스트	
배포				배포

04. 프로젝트 수행 경과 (개발환경)



04. 화면 설계 (메인 페이지)



① 상단 네비게이션 바

고정 메뉴 구성:

상담센터 소개, 심리상담, 학업상담, 자가진단, 비교과, 핵심역량, 공지사항

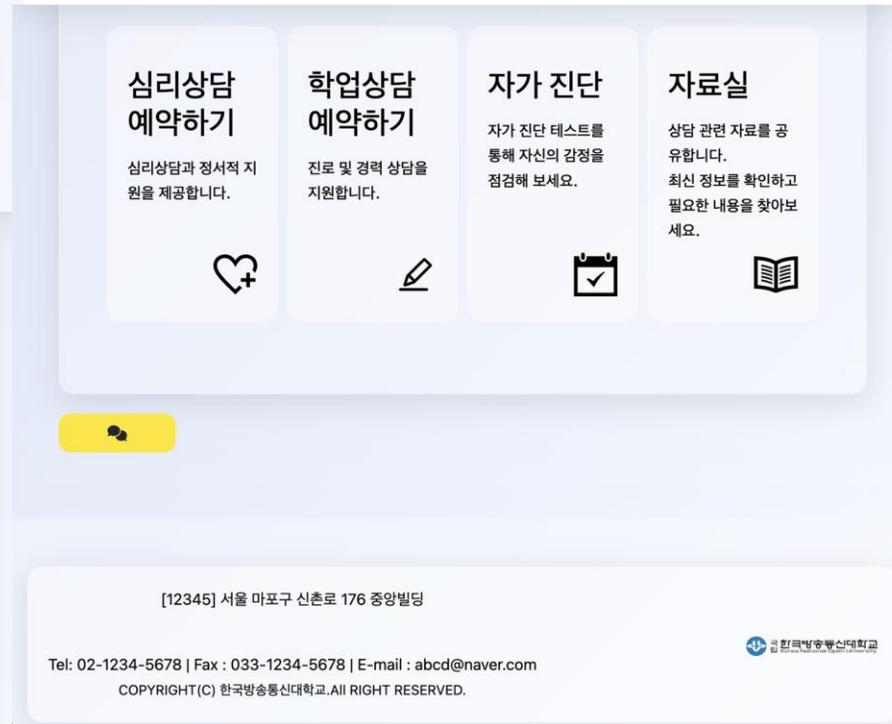
사용자가 원하는 기능으로 빠르게 이동 가능

② 대표 이미지 + 상담센터 소개 카드

주요 메시지:

"진로·심리상담실 - 한국방송통신대학 상담센터"

[자세히 알아보기] → 상담센터 소개 페이지로 이동



③ 공지사항 요약 블록

작성자 및 등록일과 함께 최신 공지 4건 노출
하단 [공지사항 더보기] 버튼으로 전체 보기

④ 주요 기능 카드 메뉴 (우측)

4가지 핵심 기능을 아이콘 카드로 구성
심리상담 예약하기
학업상담 예약하기
자가진단
자료실

⑤ 실시간 상담

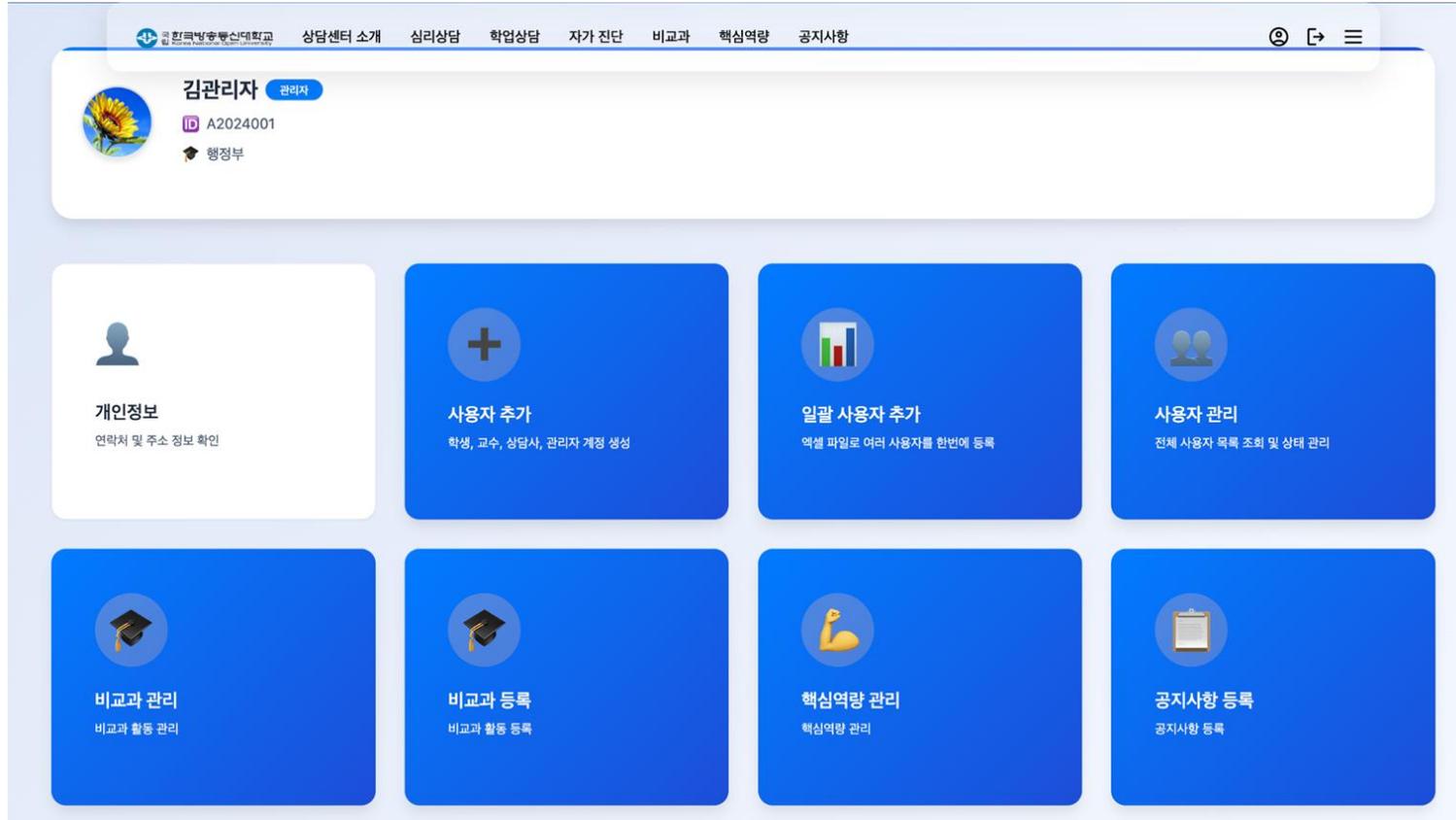
하단 우측 노란색 카카오톡 아이콘 클릭 시 실시간 상담사 채팅 팝업 연결

⑥ 하단 Footer 영역

학교 주소, 연락처, 이메일 등 기본 정보 제공
저작권 문구 및 학교 로고 배치

04. 화면 설계 (관리자 페이지)

CMS 관리자 페이지의 메인 대시보드로, 관리자가 사용자 및 비교과 활동, 핵심역량, 공지사항 등을 통합적으로 관리할 수 있도록 설계



개인정보 : 연락처, 주소 등 관리자 개인 정보 확인 가능

사용자 추가 : 학생, 교수, 상담사, 관리자 계정 신규 생성

일괄 사용자 추가 : 엑셀 파일 업로드로 다수 사용자 일괄 등록

사용자 관리 : 전체 사용자 목록 확인 및 계정 상태 관리

비교과 관리 : 비교과 활동 전체 리스트 관리 기능

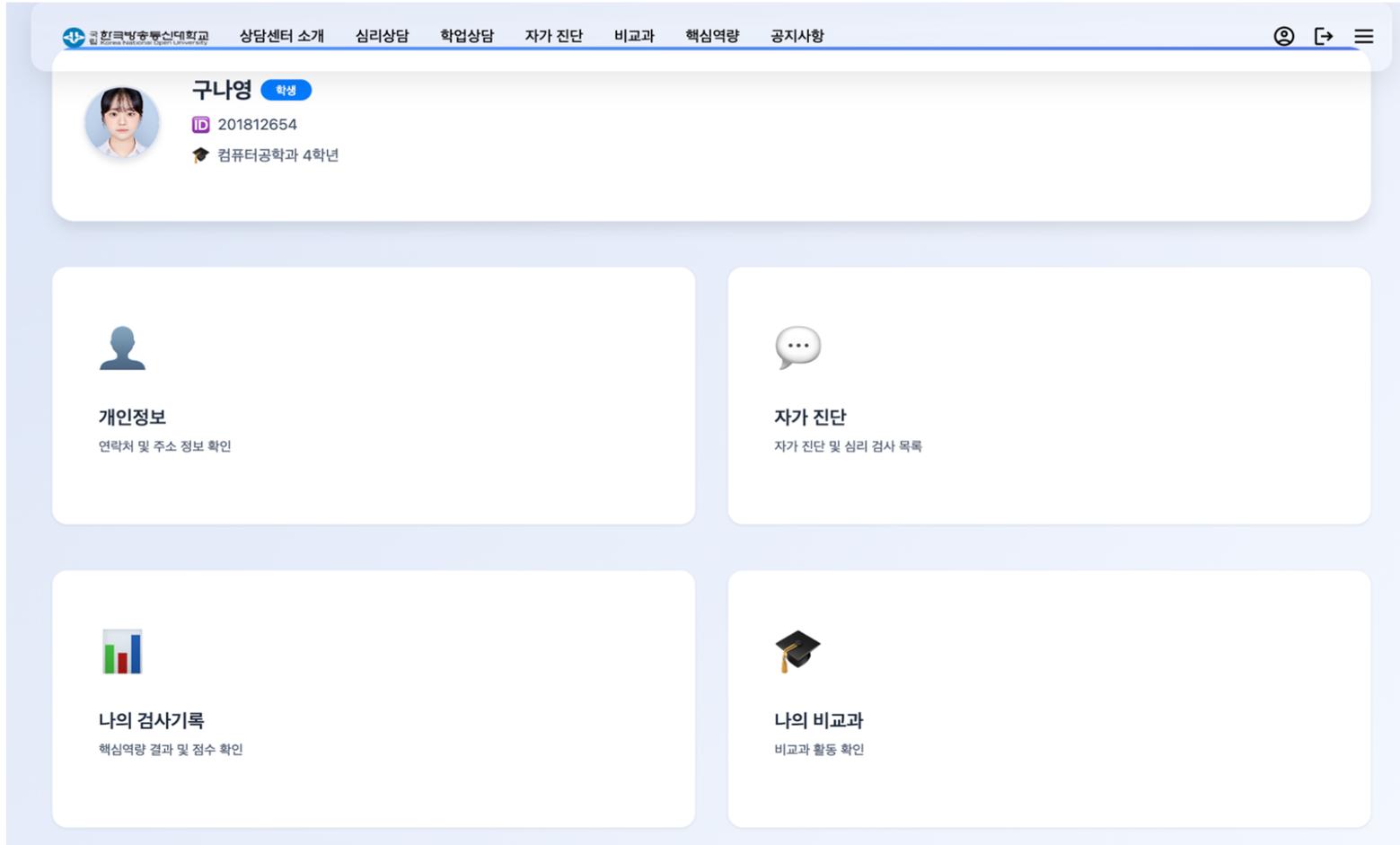
비교과 등록 : 신규 비교과 활동을 개별 등록

핵심역량 관리 : 학생 핵심역량 항목 설정 및 수정

공지사항 등록 : 관리자 권한으로 공지사항 게시

04. 화면 설계 (마이 페이지)

로그인한 학생 본인의 개인 정보, 자가 진단 결과, 핵심역량 검사 이력, 비교과 활동 등을 한눈에 확인할 수 있는 개인 맞춤형 대시보드 화면



개인정보 : 연락처, 주소 등 내 정보 확인 가능

자가 진단 : 자가진단 및 심리검사 목록 확인 가능

나의 검사기록 : 핵심역량 진단 결과 및 점수 확인 가능

나의 비교과 : 비교과 프로그램 참여 내역 확인

04. 화면 설계 (로그인 페이지)

학교 구성원(학생, 교직원)과 교외 사용자 모두를 위한 통합 로그인 창을 제공

각 사용자 유형에 맞는 로그인 방식 분기 및 소셜 로그인 기능 제공

① 학교 구성원 로그인 (좌측)

사용 대상: 학생, 교수, 교직원 등 내부 사용자

② 교외 사용자 로그인 (우측)

사용 대상: 외부 사용자
 입력 방식: 소셜 로그인으로만 로그인 가능
 소셜 계정 버튼: Kakao, Naver, Google

OAuth 2.0 소셜 로그인

(Google, Kakao, Naver)

팝업 방식 로그인 처리

(Popup window + PostMessage 통신)

HttpOnly 쿠키를 통한 토큰 보안 저장

백엔드에서 인증 로직 구현 (Spring Security + JWT)

04. 화면 설계 (상담사 페이지)

상담원 현황 (총 10 명)

1

모두보기

모두보기

배정전

내 상담

2

이름을 입력하세요.

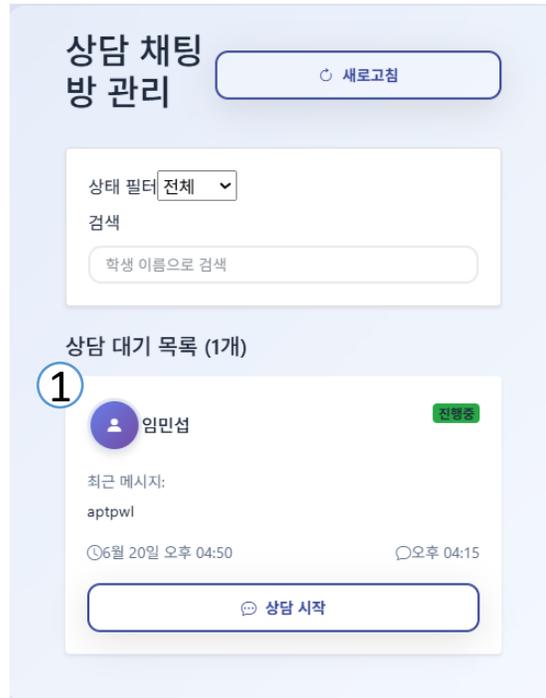
3

상담원 이름	학번	이메일	연락처	배정
상담원	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담대기	2024001234	minsub00@university.ac.kr	010-1234-5678	배정
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료
상담중	2024001234	minsub00@university.ac.kr	010-1234-5678	배정완료

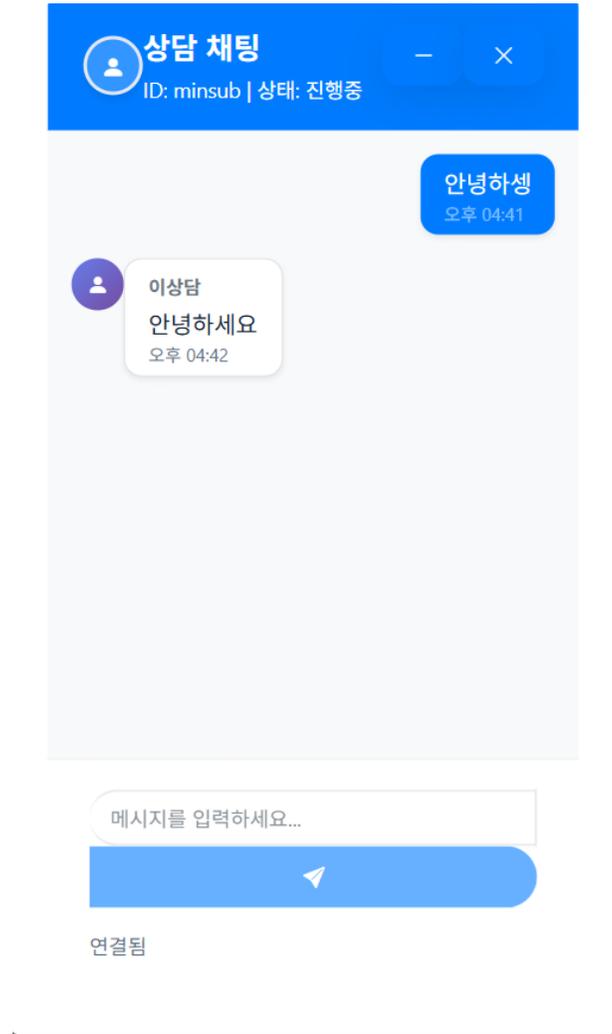
이전 1 다음

1. 셀렉트 박스 선택에 따른 정렬
2. 이름, 학번, 이메일 검색
3. 배정 버튼 클릭 시 배정 완료 처리

04. 화면 설계 (상담사 페이지)



②



1. 학생이 버튼 클릭 시 방 생성
2. 실시간 채팅 가능

04. 화면 설계 (공지사항 - 리스트)

공지사항

번호	제목	작성자	조회수	작성일
11	asd	ccc00	0	2025-07-01 04:30:09
10	┌──────────────────┐	ccc00	2	2025-07-01 03:39:37
9	oooooooooooooooooooo	ccc00	0	2025-07-01 03:39:32
8	oooooooooooooooooooo	ccc00	0	2025-07-01 03:39:27
7	□□□□□□□□	ccc00	0	2025-07-01 03:39:22
6	□□□□	ccc00	0	2025-07-01 03:39:17
5	□L□□	ccc00	0	2025-07-01 03:39:12
4	□L□□□□L□□□	ccc00	0	2025-07-01 03:39:08
3	되나요..?	ccc00	30	2025-06-26 17:27:50
2	공지공지	ccc00	28	2025-06-26 17:07:09

이전 1 2 다음

글쓰기

공지사항의 리스트로 10개의 공지사항 기준으로 페이지네이션을 설정하였으며 로그인한 유저의 정보에 따라 글쓰기 기능이 활성화 됩니다.

04. 화면 설계 (공지사항 - 등록)

공지사항 등록

제목

작성자

ccc00

내용

← → Paragraph ▼ **B** *I*    ▼   ▼    

첨부파일 선택된 파일 없음

등록

공지사항 등록 화면으로 CKEditor를 넣었으며 학생을 제외한 유저만 등록 가능합니다.

04. 화면 설계 (공지사항 - 등록)

공지사항 상세

제목	되나요..?
작성자	ccc00
조회수	28
작성일	2025-06-26 17:27:50
첨부파일	snake-head.png

□ L O

목록 수정 삭제

공지사항 상세 페이지로 학생을 제외한 유저만 수정, 삭제 기능이 활성화 되며 사용 가능합니다. 파일을 저장할 수 있습니다.

04. 화면 설계 (핵심역량 - 등록 1)

핵심역량 설문 관리

설문 리스트

제목	설명	학과	등록일	Actions
컴공 설문 2	aa	컴퓨터공학과	2025. 6. 22.	수정 삭제
경영학과 설문1	경영학과 학생들은 꼭 해주세요!!	경영학과	2025. 6. 22.	수정 삭제
컴공설문1	sss	컴퓨터공학과	2025. 6. 18.	수정 삭제
창의융합역량	새로운 아이디어를 창출하고 다양한 분야를 융합하여 문제를 해결하는 능력		2025. 6. 16.	수정 삭제
창의적사고	기존의 틀을 벗어나 독창적인 아이디어를 생성하는 능력		2025. 6. 16.	수정 삭제
융합적사고	서로 다른 분야의 지식을 연결하여 새로운 가치를 창출하는 능력		2025. 6. 16.	수정 삭제

신규 설문 등록

설문 제목

설문 설명

학과 선택

핵심역량 설문관리 페이지로 기존의 설문을 수정, 삭제할 수 있으며 신규 설문을 등록할 수 있습니다. 신규 설문 등록 시 설문의 제목, 설명, 학과를 선택하고 다음을 클릭 시 다음 화면과 같이 하단에 출력됩니다.

04. 화면 설계 (핵심역량 - 등록 2)

신규 설문 등록

의사소통

삭제

문항 추가

문제해결

삭제

문항 추가

자기관리

삭제

문항 추가

대인관계

삭제

문항 추가

글로벌역량

삭제

문항 추가

직업윤리 및 책임역량

삭제

문항 추가

이전

설문 등록하기

신규 설문 등록 화면으로
핵심역량 6가지에 따라 각각 추가가 가능합니다.
추후 학생이 설문 완료 시 각 각의 핵심 역량을 점수로
나누어 100점 만점으로 점수를 출력 합니다.

04. 화면 설계 (핵심역량 - 분석 1)

핵심역량 이력 및 분석 (상담사용)

학과 선택

전체

학생 선택

선택하세요

최근 설문 이력

[2025. 6. 26.] 최영희 (20240004)

문제해결: 100.00

의사소통: 80.00

글로벌역량: 100.00

대인관계: 100.00

직업윤리 및 책임역량: 100.00

[2025. 6. 25.] 구나영 (201812654)

문제해결: 100.00

의사소통: 100.00

자기관리: 100.00

글로벌역량: 100.00

대인관계: 100.00

직업윤리 및 책임역량: 100.00

[2025. 6. 30.] 임민섭 (2024001234)

의사소통: 100.00

문제해결: 40.00

자기관리: 20.00

글로벌역량: 100.00

대인관계: 60.00

직업윤리 및 책임역량: 80.00

핵심역량 이력 및 분석 화면으로 상담사가 모든 학생의 핵심역량 이력을 확인할 수 있습니다. 학생을 선택 시 다음과 같은 화면이 출력됩니다.

04. 화면 설계 (핵심역량 - 분석 2)

핵심역량 분석 결과

최근 응시일자: 2025. 6. 25.

문제해결: 100.00

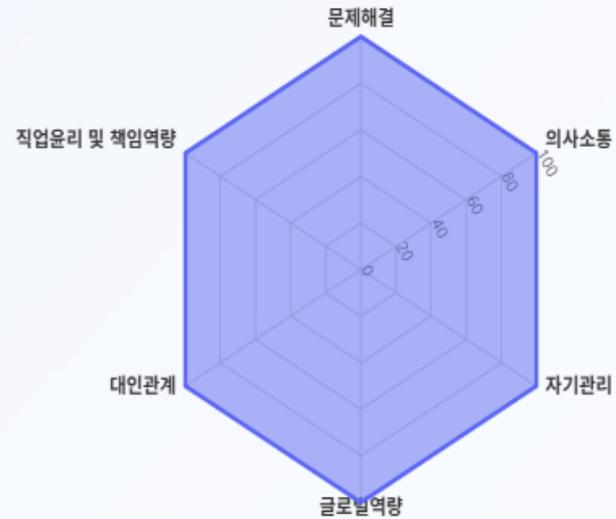
의사소통: 100.00

자기관리: 100.00

글로벌역량: 100.00

대인관계: 100.00

직업윤리 및 책임역량: 100.00



학생 선택 시 나오는 화면으로
최근 응시 일자, 각 역량 별 점수를 Rechart.js를
사용하여 출력시킵니다.

04. 화면 설계 (핵심역량 설문 - 리스트)

핵심역량

전체

컴공 설문 2
aa
설문하기

경영학과 설문1
경영학과 학생들은 꼭 해주세요!!
설문하기

컴공설문1
sss
설문하기

창의융합역량
새로운 아이디어를 창출하고 다양한 분야를 융합하여 문제를 해결하는 능력
설문하기

창의적사고
기존의 틀을 벗어나 독창적인 아이디어를 생성하는 능력
설문하기

융합적사고
서로 다른 분야의 지식을 연결하여 새로운 가치를 창출하는 능력
설문하기

핵심역량 설문 리스트 화면입니다.
Select에 전체, 자신의 학과가 출력되며 검색이 가능하고 설문하기 버튼을 클릭하여 설문 화면으로 넘어갑니다.

04. 화면 설계 (핵심역량 설문 - 설문)

핵심역량 설문

핵심역량 ID: 16deba4e-d1a0-4841-b31a-7c430b3c5201

문항 수: 6문항

사용자 학번: C20240012

문항 1. ○

1점 2점 3점 4점 5점

문항 2. L

1점 2점 3점 4점 5점

문항 3. R

1점 2점 3점 4점 5점

문항 4. ○ □

1점 2점 3점 4점 5점

문항 5. □

1점 2점 3점 4점 5점

문항 6. □ L

1점 2점 3점 4점 5점

설문 제출하기

핵심역량 설문 화면입니다.

전체 문항 수, 현재 로그인한 학생의 학번이 출력되며
문항 별 1~5점을 계산해 백분율로 결과를
저장시킵니다.

04. 화면 설계 (핵심역량 - 내 결과)



내 핵심역량 결과 화면입니다.

종합적으로 방사형 차트를 이용해 자신의 점수, 학과 점수, 전체 점수 평균을 보여주고 부족한 부분과 강점 부분을 나타내고 개선이 필요한 부분의 비교과를 추천해줍니다.

04. 화면 설계 (비교과 프로그램 - 리스트)

1 비교과 프로그램 정보

비교과 프로그램 리스트 출력
모집정보와 모집 상태, DDAY 등을 목록에서 확인 가능

비교과 프로그램 내 비교과 활동

전체 상태 | 프로그램명으로 검색

총 2개의 프로그램

AI 알고리즘 해커톤 캠프
D-8

모집중 300

컴퓨터공학과

2025.07.08. ~ 2025.07.10. 30명

POLICY WORKSHOP

모집마감

종료 200

창업 아이디어 챌린지

경영학과

2025.06.24. ~ 2025.06.27. 20명

2 학생 비교과 활동 내역

로그인 한 학생의 비교과 프로그램
신청내역 및 이수 완료내역, 비교과로 획득한 마일리지
를 모달로 확인 가능

내 비교과 활동 [X]

200 총 마일리지	1 이수완료	1 총 신청
신청 내역	이수완료	마일리지 내역
창업 아이디어 챌린지 신청일: 2025. 6. 22.	이수완료	
신청 내역	이수완료	마일리지 내역
창업 아이디어 챌린지 신청일: 2025. 6. 22.	이수완료	
신청 내역	이수완료	마일리지 내역
창업 아이디어 챌린지 2025. 6. 30. 가산		+200점

04. 화면 설계 (비교과 프로그램 - 상세)

1

신청하기 클릭 시 신청 modal 노출

이미 신청이 완료되었다면 신청완료, 종료된 프로그램의 경우 신청불가로 버튼 텍스트 변경

AI 알고리즘 해커톤 캠프



문의처
 ✉ 이메일 computersc@knou.kr
 ☎ 전화번호 02-4568-5968
 📄 운영부서 컴퓨터공학과
 📍 장소 IT융합관 해커톤랩

모집조건
 ✎ 대상 재학생
 🎓 학과 전체
 ☑ 학년 전체
 📅 교육기간 2025. 07. 08. ~ 2025. 07. 10.

📍 모집인원 300명

신청현황 0명 / 30명

신청하기

핵심역량

전공 실문 2
 경영학과 실문1
 경영실문1
 창의융합역량
 창의적사고
 융합핵사고

프로그램 소개

인공지능(AI) 알고리즘을 활용한 실전 문제 해결 해커톤 캠프입니다. 딥러닝, 머신러닝 기초 지식을 바탕으로 팀 프로젝트를 구성하여 실제 문제 상황에 적용 가능한 AI 모델을 기획하고 구현합니다. 이 프로그램은 AI에 관심 있는 모든 학과 학생이 참여 가능하며, 전공 수준에 맞춰 난이도를 조정하여 운영됩니다.

프로그램 일정

· 프로그램 일정 1일차 AI 산업 트렌드 및 기술 이해 문제정의 및 팀 빌딩 데이터 이해 및 전처리 실습 2일차 머신러닝/딥러닝 모델 설계 워크숍 팀별 알고리즘 구현 및 테스트 중간 피드백 & 리빙딩 3일차 발표자로 준비 팀별 프로젝트 발표 및 평가 전문가 멘토 피드백 기대 효과 AI 기술 및 머신러닝 기초에 대한 실습 중심의 이해 문제 해결 중심의 프로젝트 수행 경험 팀 협업과 실천 발표 능력 향상 실무 중심의 데이터 기반 사고력 향상

첨부파일

AI 알고리즘 해커톤 캠프 신청서.docx 19.7KB

목록으로 돌아가기

프로그램 신청서

신청 프로그램

AI 알고리즘 해커톤 캠프
 기간: 2025. 7. 8. ~ 2025. 7. 10.
 모집인원: 30명 (현재 0명 신청)

신청자 정보

학번: 2024001234
 이름: 임민섭

학과: 컴퓨터공학과
 학년: 3학년

이메일: minsub00@university.ac.kr

연락처 * : 010-1234-5678
*연락처는 수정 가능합니다

신청서 및 추가정보

신청서 파일 업로드

AI 알고리즘 해커톤 캠프 신청서.docx

* PDF, DOC, DOCX, HWP 파일만 업로드 가능

선택된 파일
 AI 알고리즘 해커톤 캠프 신청서.docx

비고사항

추가로 전달하고 싶은 내용이나 특이사항이 있다면 작성해주세요.

2

학생 신청서 첨부 기능

04. 화면 설계 (비교과 관리자페이지 - 프로그램 리스트)

비교과 프로그램 관리

프로그램 관리 | 신청자 관리 | 통계

프로그램명 검색 모든 상태 모든 부서

총 2개 프로그램

프로그램 목록 **새 프로그램 등록**

프로그램명	운영부서	모집기간	정원	신청자	상태	액션	관리
SI 알고리즘 해커톤 캠프	컴퓨터공학과	2025. 7. 8. ~2025. 7. 10.	30명	0명	모집중	모집중	수정 삭제
창업 아이디어 챌린지	경영학과	2025. 6. 24. ~2025. 6. 27.	20명	0명	종료	종료	수정 삭제

1

비교과 프로그램 등록

2

비교과 프로그램 신청자 관리

클릭시 해당 프로그램의 신청자 관리로 이동

04. 화면 설계 (비교과 관리자페이지 - 프로그램등록)

비교과 프로그램 등록

기본 정보

프로그램명 *

프로그램 소개 *

인공지능(AI) 알고리즘을 활용한 실전 문제 해결 해커톤 캠프입니다. 딥러닝, 머신러닝 기초 지식을 바탕으로 팀 프로젝트를 구성하여 실제 문제 상황에 적용 가능한 AI 모델을 기획하고 구현합니다. 이 프로그램은 AI에 관심 있는 모든 학과 학생이 참여 가능하며, 전공 수준에 맞춰 난이도를 조정하여 운영됩니다.

핵심역량 매핑

이 프로그램과 관련된 핵심역량을 모두 선택해주세요.

프로그램 정보

모집인원(정원) * **지금 마일리지 ***

학적

전학년	전체	전체
-----	----	----

시작일시 * **종료일시 ***

운영 정보

운영부서 * **교육 장소**

문의 이메일 **문의 연락처**

세부 정보 및 파일 첨부

세부 일정 *

문제정의 및 팀 빌딩
데이터 이해 및 전처리 실습
2일차

첨부파일 (선택)

(각 50MB 이하, 다중 선택 가능)

AI 알고리즘 해커톤 캠프 신청서.docx (19.72 KB)

썸네일 이미지 (선택)

(10MB 이하 이미지 파일 (목록 페이지에 표시됩니다))

1 비교과 프로그램에 해당하는 핵심역량 매핑

핵심역량 다중 선택 가능

2 FTP를 활용한 다중 파일첨부 기능 및 썸네일 등록

비교과 프로그램 소개 및 신청서, 프로그램 썸네일을 위해 FTP를 이용한 다중 파일 첨부 시스템 썸네일 미첨부시 기본 이미지 등록

3 비교과 프로그램별 마일리지 설정

04. 화면 설계 (비교과 관리자페이지 - 신청자 관리)

비교과 프로그램 관리

프로그램 관리 신청자 관리 통계

AI 알고리즘 해커톤 캠프
컴퓨터공학과 | 정원: 30명 | 신청: 1명

모든 상태 ▾

1 비교과 프로그램 신청상태 변경
다중 및 개별 상태 변경 가능
프로그램 이수완료 시설정한 마일리지 지급

일괄 승인 일괄 거부 일괄 이수완료

<input checked="" type="checkbox"/>	신청번호	학번	이름	신청일시	신청구분	신청서	상태	액션
<input checked="" type="checkbox"/>	APLY20250002	2024001234	임민섭	2025. 6. 30. 오후 9:02:40	일반신청	↓	신청완료	승인 거부

2 학생 신청서 다운로드
학생이 첨부한 신청서 다운로드 기능

04. 프로젝트 수행 경과 (ERD)

ERD URL :

05. 프로젝트 수행 경과

테스트 아이디	아이디	비밀번호
학생		
상담사		
최고관리자		

영상 부분 해당 영상은
시연회 때 부착

06. 트러블 슈팅 - 김선익

문제 식별

프로젝트 초기에는 핵심역량 데이터를 일관성 있게 관리하지 못해, 학생별로 설문 응답 이력과 역량 점수를 체계적으로 조회하거나 비교하기 어려웠습니다. 이로 인해 학생 개개인의 강점과 약점을 파악하고 맞춤형 지도를 제공하는 데 필요한 기본 자료가 부족한 상태였습니다.

문제 해결 접근 방법

비즈니스 로직 분리: CoreCptEvalController와 CcaScoreService를 도입하여 역량별 점수 산출 및 요약 로직을 서비스 계층에 구현했습니다.

사용자 인터페이스 개선: CCA_AnalysisPage.jsx에서 학생 선택과 분석 결과 시각화를 지원해, 사용자(상담 교사 또는 관리자)가 클릭 한 번으로 관련 데이터를 확인할 수 있도록 구성했습니다.

해결 과정

데이터 저장 및 가공

학생이 설문을 제출하면 응답 기록과 역량별 원점수가 데이터베이스에 저장됩니다.

CcaScoreService에서 저장된 점수를 불러와 각 역량별 평균점수와 추천 메시지를 계산합니다.

컨트롤러 연동

CoreCptEvalController가 HTTP 요청을 받아 서비스 계층 호출 후, 클라이언트에 JSON 형태로 결과를 반환합니다.

화면 구현

React 컴포넌트인 CCA_AnalysisPage.jsx에서 드롭다운으로 학생을 선택하면, API 응답을 받아 레이더 차트로 시각화합니다.

레이더 차트는 recharts 라이브러리를 활용해 각 역량별 점수를 직관적으로 비교할 수 있도록 구성했습니다.

결과 및 교훈

핵심역량을 세부 항목으로 분류·관리함으로써, 학생별 강점과 약점을 한눈에 파악할 수 있는 기반이 마련되었습니다. 상담 시 구체적인 데이터에 기반한 피드백 제공이 가능해졌습니다.

CORS 설정, API 베이스 URL 등 환경 의존 설정은 외부 설정 파일로 분리해야 변경 관리가 용이합니다.

역량 분석 로직은 신규 역량 항목 추가나 가중치 조정 등 요구사항 변화에 대비해 모듈화 및 인터페이스 확장이 가능하도록 설계해야 합니다.

앞으로는 분석 결과를 대시보드 형태로 통합해, 실시간 모니터링 및 전체 현황 조회 기능을 강화할 계획입니다.

06. 트러블 슈팅 - 구나영

문제 식별

소셜 로그인 기능을 OAuth2 기반으로 구현하고, 사용자 경험 향상을 위해 팝업 방식을 함께 적용하였다. 그러나 개발 과정에서 연속적인 문제가 발생하였다. 첫 번째는 로그인 시 500 에러가 발생한 것이었고, 로그에는 "Name for argument of type [java.lang.String] not specified"라는 메시지가 출력되었다. 이는 @PathVariable이 파라미터 이름을 인식하지 못하는 문제였다. 두 번째는 CORS 정책에 의해 프론트엔드 요청이 403 에러로 차단되는 이슈였다. 마지막으로 가장 예기치 못했던 문제는 팝업 창에서 로그인 완료 후 popup.closed로 상태를 확인하려 할 때, 브라우저에서 Cross-Origin-Opener-Policy에 의해 window.closed 호출이 차단되는 것이었다.

문제 해결 접근 방법

단순히 각각의 에러를 해결하는 데 그치지 않고, 보안성과 사용자 경험을 함께 고려한 구조를 목표로 설정하였다. 문제가 발생한 원인을 코드와 설정 파일, 브라우저 정책 등 다양한 관점에서 분석했고, 궁극적으로 유지보수와 확장성까지 고려한 방식으로 개선하기 위해 구조적인 접근을 시도하였다. 또한 문제가 발생한 기술 요소들(Spring 컴파일러 옵션, CORS 설정, 브라우저 보안 정책 등)을 정확히 이해하고, 각 기술이 시스템 전체에 어떤 영향을 미치는지를 고려하여 대응 방향을 잡았다.

해결 과정

먼저 @PathVariable 관련 500 에러는 Java 컴파일 시 매개변수 이름을 유지하지 않아 발생한 문제였다. 이는 build.gradle 파일에 -parameters 컴파일러 플래그를 추가하고, 파라미터에 명시적으로 이름을 지정하는 방식으로 해결하였다. 두 번째 CORS 관련 문제는 Spring Security에서 외부 요청을 차단하고 있었기 때문이었다. 이를 해결하기 위해 CorsConfigurationSource를 구성하고, 허용할 도메인과 HTTP 메서드를 명확히 설정하여 서버가 CORS 요청을 정상적으로 허용하도록 수정하였다. 마지막으로, 팝업 방식에서 발생한 Cross-Origin-Opener-Policy 관련 문제는 최신 브라우저 보안 정책으로 인해 popup.closed를 사용할 수 없게 된 것이 원인이었다. 이에 로그인 완료 후 팝업 창에서 메인 창으로 postMessage를 통해 완료 메시지를 전달하고, 메인 페이지에서는 이를 수신해 로그인 상태를 처리하도록 하였다. 또한 만약 메시지를 수신하지 못했을 경우를 대비해 서버에 로그인 완료 상태를 확인하는 API를 호출하는 폴백 로직도 함께 구현하였다.

결과 및 교훈

최종적으로 HttpOnly 쿠키에 토큰을 저장함으로써 XSS 공격을 차단하고, 팝업 방식을 통해 페이지 이동 없이 부드러운 로그인 사용자 경험을 제공할 수 있었다. 또한 하나의 OAuth 인증 흐름으로 구글, 카카오, 네이버 등의 플랫폼을 통합 처리할 수 있는 확장성 있는 구조도 완성하였다. 이 과정을 통해 단순히 발생한 오류를 수정하는 것을 넘어, 보안과 UX를 모두 만족시키는 구조적 해결이 중요하다는 것을 체감했다. 특히 문제를 다양한 관점에서 분석하고 원인을 정확히 파악하는 능력, 그리고 새로운 보안 정책이나 브라우저 동작 방식에 유연하게 대응하는 역량의 필요성을 깨달을 수 있었다. 예기치 못한 문제가 오히려 시스템의 완성도를 높이는 계기가 되었으며, 이후 비슷한 상황에서도 빠르게 대응할 수 있는 기반이 마련되었다.

06. 트러블 슈팅 - 오민영

문제 식별

비교과 프로그램 시스템에서 프로그램 ID와 신청 ID를 생성하는 idgenerator를 구현했는데, 동시에 여러 사용자가 신청하거나 프로그램을 등록할 때 가끔 중복된 ID가 생성되는 문제가 발생했습니다. 특히 신청 마감일에 학생들이 몰리면서 동시 신청이 증가했을 때 같은 신청 ID가 생성되어 데이터베이스 PRIMARY KEY 오류가 발생했습니다.

문제 해결 접근 방법

처음에는 Java의 synchronized 키워드로 메서드를 동기화해보려고 했지만, 근본적인 문제는 데이터베이스에서 최신 ID를 조회하는 시점과 새로운 ID로 저장하는 시점 사이에 다른 트랜잭션이 끼어들 수 있다는 것이었습니다. 그래서 데이터베이스 레벨에서 원자적으로 처리할 수 있는 방법을 찾아보기로 했습니다. Redis 같은 NoSQL을 사용하면 해결할 수 있다는 것을 알고는 있었지만, 현재 프로젝트에서는 별도 인프라 추가가 어려운 상황이었습니다.

해결 과정

이에 기존 코드에서 최소한의 변경으로 해결하기로 했습니다. 처음에는 데이터베이스에 별도 시퀀스 테이블을 만들어서 해결하는 방법도 고려해봤지만, 테이블 추가와 기존 로직 변경이 너무 큰 작업이라고 판단했습니다. 그래서 Java의 synchronized 키워드를 사용해서 ID 생성 메서드 전체를 동기화 처리하는 방법을 선택했습니다. 이 방법은 한 번에 하나의 스레드만 ID 생성 로직을 실행할 수 있도록 제한해서 동시성 문제를 방지하는 방식입니다. 물론 성능 측면에서는 약간의 병목이 생길 수 있지만, ID 생성 자체가 빈번하게 발생하는 작업이 아니고 처리 시간도 매우 짧기 때문에 실제 사용자 경험에는 영향이 없을 것으로 예상했습니다.

결과 및 교훈

synchronized 키워드를 추가한 후 동시 신청이 몰리는 상황에서도 중복 ID 생성 문제가 해결되었습니다. 메서드 레벨에서 동기화 처리를 하기 때문에 한 번에 하나의 스레드만 ID를 생성할 수 있어 중복이 발생하지 않았습니다. 다만 synchronized는 단일 서버 환경에서만 유효하다는 한계가 있어서, 향후 서버를 여러 대로 확장하게 되면 결국 같은 문제가 다시 발생할 수 있다는 것을 인식하게 되었습니다. 이 과정에서 Redis와 같은 인메모리 데이터베이스나 NoSQL의 카운터 기능들을 공부해보니, 실제로는 이런 분산 환경에서의 ID 생성이나 순차 번호 관리에 더 적합한 해결책이라는 것을 알게 되었습니다. Redis는 애초에 이런 용도로 설계된 기능들을 제공하고, 멀티 서버 환경에서도 문제없이 동작할 수 있기 때문입니다. 결국 현재 프로젝트에서는 synchronized로 임시 해결했지만, 다음에는 Redis 같은 기술들을 더 깊이 공부해서 처음부터 확장성을 고려한 설계를 적용하고 싶다는 생각이 들었습니다.

06. 트러블 슈팅 – 임민섭

문제 식별

cms에서 WebSocket을 활용한 1:1 실시간 채팅 기능 구현 중,
상담사 여러 명이 동일한 채팅방에 동시에 입장하는 문제가 발생하였습니다.
하나의 사용자가 상담방을 생성하면 상담사 중 1명만 입장할 수 있도록 하려고 했으나,
입장 제한 없이 누구나 동일 방에 접속되어 다중 입장으로 인한 메시지 혼선이 발생하였습니다.

문제 해결 접근 방법

처음에는 프론트엔드 단의 버튼 제어나 화면 상태 문제로 생각했으나,
백엔드에서 ChatRoom의 assignedCounselorId가 조건 없이 덮어쓰지는 구조였음을 발견하였습니다.
또한 실시간 처리이기 때문에 동시 요청 처리 시점에 따라 이미 배정된 방에 다른 상담사가 입장하는 상황이 발생하였고,
따라서 세션 상태 판단 + 입장 제어 + 중복 요청 방지를 통합적으로 고려해야 함을 깨달았습니다.

해결 과정

비즈니스 제약 명확화 : assignedcounselorId가 null인 경우에만 입장 허용, 이미 배정된 경우에는 403 에러로 응답

DB에서도 동시성 제어 보완 : roomId Unique 제약 추가, @Transaction 사용

프론트엔드 처리 추가 : 입장 실패 시 UI 출력 "다른 상담사가 먼저 입장하였습니다", 중복 요청 방지 > 입장 버튼 클릭 시 disable 처리

결과 및 교훈

실시간 채팅 시스템에서의 비즈니스 제약 조건은 서버가 책임지고 강제해야 한다는 원칙을 확실히 확인하였으며,
단순히 화면 제어로 해결하려 하지 말고, DB 제약 + 백엔드 로직 + UI 처리를 함께 설계해야 한다는 것을 알았습니다.
트러블을 통해 assignedCounselorId가 핵심 컨트롤 포인트라는 인식을 명확히 하고,
이후 모든 채팅 로직 설계 시 이를 기준으로 동작 흐름을 명확히 정리할 수 있게되었습니다.

07. 자체 평가 의견

프로젝트 결과물에 대한 완성도 평가 (10점 만점)

[프로젝트의 기획 의도 부합성 및 평가] - 8점

이번 상담 시스템 CMS 프로젝트는 학교 상담센터의 진로·심리상담, 학업상담, 비교과 활동, 핵심역량 평가 등 다양한 기능을 통합적으로 관리할 수 있는 시스템을 구축하는 것이 목표였다. 사용자인 학생과 관리자의 관점 모두를 반영하여, 각각에 맞는 대시보드와 기능을 제공하고자 하였다.

전체적인 결과물을 기준으로 볼 때, 초기 기획에서 제시한 권한 기반 화면 분기, 사용자별 기능 제공, 소셜 로그인 기능 등 핵심 목표는 대부분 구현되었으며, 실제 사용 환경을 가정한 UX 설계와 보안 구조도 비교적 충실하게 반영되었다. 특히 관리자와 학생의 접근 권한을 구분하고, OAuth2 기반의 인증 방식을 통해 보안성과 확장성을 동시에 확보한 점은 기획 의도에 부합하는 성과라고 평가할 수 있다.

다만 일부 UI/UX 세부 기능과 사용자 편의 요소(알림 시스템, 접근 로그 등)는 구현되지 않아, 완성도 측면에서 아쉬운 점도 존재한다. 프로젝트의 기획 의도 자체에는 높은 점수를 줄 수 있으나, 최종 결과물이 기획 전부를 만족시키지는 못했기 때문에 부합성 평가 점수는 0점 처리한다.

[프로젝트 결과에 따른 추후 개선점]

프론트엔드 구조 개선

React의 컴포넌트 구조화, 상태 관리 방식 개선 등 코드의 재사용성과 유지보수성을 높이기 위한 리팩토링 필요

사용자 피드백 기반의 기능 추가

실제 상담센터 사용자(학생, 관리자)의 피드백을 받아, UI 간편화, 마이페이지 기능 개선, 알림 기능 도입 등 사용성 향상을 위한 기능 보완 필요

[프로젝트 수행에 따른 느낀 점]

이번 프로젝트는 단순한 기능 구현을 넘어, 사용자 경험과 보안 구조를 동시에 고려한 설계와 개발이 얼마나 중요한지를 깊이 체감할 수 있는 경험이었다. Spring Security, JWT, OAuth2 등의 인증 기술을 실제 서비스 환경에 맞게 적용하면서 이론과 실무의 차이를 실감했고, 에러 발생 시 그 원인을 구조적으로 분석해 해결하는 과정을 통해 개발자로서의 문제 해결 역량도 키울 수 있었다.

개인적으로는 백엔드 역할에 집중하며 인증과 권한 처리, 사용자 맞춤 정보 제공 등을 안정적으로 구현했지만, 프론트엔드 프레임워크인 React에 대한 이해가 부족해 충분히 능동적으로 활용하지 못한 점은 아쉬움으로 남는다. 앞으로는 백엔드뿐 아니라 프론트엔드 구조에도 더 깊이 있는 이해를 갖춰 전체 시스템을 조율할 수 있는 개발자로 성장하고자 한다.

07. 자체 평가 의견

프로젝트 결과물에 대한 완성도 평가 (10점 만점)

[프로젝트의 기획 의도 부합성 및 평가] - 5점

핵심역량 관리 기능은 CoreCptEvalController와 CcaScoreService를 통해 설문 제출과 점수 계산 로직을 갖추고 있습니다. 점수 요약과 추천 프로그램을 함께 제공하는 API가 구현되어 있으며 공지사항 기능은 파일 첨부와 다운로드를 지원하는 createWithFiles 메서드를 포함하여 CRUD 전반을 처리합니다. 전반적으로 기능 구현은 되어 있으나 코드 구조와 설정 관리가 미완성이라 기획 의도에 완전히 부합했다고 보기는 어려워 10점 만점에 5점으로 평가합니다.

[프로젝트 결과에 따른 추후 개선점]

SecurityConfig에서 허용 Origin을 외부 설정으로 빼내고, 프론트엔드 코드에 하드코딩된 http://localhost:8082도 환경 변수로 관리하면 다양한 배포 환경에서 활용하기 쉬울 것 같음

[프로젝트 수행에 따른 느낀 점]

핵심역량 데이터를 세분화하여 저장하고, 각 항목별 평균과 추천 프로그램을 제공하는 구조를 만들면서 역량 분석의 중요성을 체감했습니다.

공지사항 기능에서는 파일 업로드·다운로드 처리와 권한 관리를 동시에 다뤄야 해서 복잡도가 높았지만, 사용자가 원하는 정보를 한 번에 조회할 수 있도록 구성할 수 있었습니다.

이후에는 초기 설계 단계부터 환경 설정과 코드 정리를 더 치밀하게 준비해야 유지보수와 배포가 수월하다는 교훈을 얻었습니다.

07. 자체 평가 의견

김선익

개인 성과 : 핵심역량을 여섯 가지로 세분화해 기록 및 관리하면서 분석 및 피드백 능력이 향상되었습니다.
데이터를 기반으로 학생의 강점과 약점을 빠르게 파악해 상담사가 적절한 조언이나 지원을 제공할 수 있었습니다.

개인적 아쉬운 점 : 초기 계획 단계에서 역량별 목표와 측정 지표를 명확히 세우지 못해, 중간에 방향을 수정하느라 시간이 조금 소요되었습니다.

구나영

개인 성과 : Spring Security 기반의 인증·인가 체계를 설계하고, JWT를 활용한 토큰 기반 인증 방식과 OAuth2 소셜 로그인을 성공적으로 구현했다. 특히 보안성과 사용자 경험을 모두 고려하여 HttpOnly 쿠키를 통한 토큰 관리, 팝업 방식의 소셜 로그인, CORS 및 브라우저 정책 대응까지 실무 수준의 인증 구조를 완성할 수 있었다. 또한 사용자 권한에 따라 접근을 제어하는 로직과 마이페이지의 사용자 맞춤 정보를 안정적으로 처리하며 백엔드의 흐름을 전체적으로 이해하고 구현하는 역량을 높일 수 있었다.

개인적 아쉬운 점 : 프론트엔드에서는 React를 기반으로 작업했지만, 당시 리액트에 대한 학습 시간이 부족해 프레임워크의 장점을 충분히 활용하지 못한 점이 아쉬움으로 남는다. 컴포넌트 재사용, 상태관리, 비동기 처리 등에서 더 효율적인 코드 작성이 가능했음에도 이를 깊이 있게 적용하지 못해 아쉬웠고, 이후 개인적으로 React 관련 내용을 다시 학습하며 개선 중이다.

임민섭

개인 성과 : WebSocket을 이용한 실시간 채팅 시스템과 상담사 관리페이지를 구현하며, 백엔드 조건 체크 및 프론트 UI 제어의 중요성을 깨달음

개인적 아쉬운 점 : React의 기능을 반도 활용하지 못한 것이 아쉬움, React에 대해 제대로 이해하고 시작했더라면 더 나은 결과가 있었을 것 같아 React에 대해 공부해보고 싶은 계기가 됨



Thank You For Listening!